



---

# Orbix 3.0.2

## Release Notes

### March 2000

---

#### Contents

<b>Introduction</b>	<b>3</b>
<b>Orbix 3.0.1 and Later Versions</b>	<b>3</b>
<i>Development Environments</i>	3
Solaris 7	3
Solaris 2.5.1	3
HP-UX 10.20	5
HP-UX 11.00	5
IRIX 6.5	6
NT 4.0	7
Win-95 and Win-98	7
<i>New Features in Orbix 3.0.1 and Later Versions</i>	7
Server Context Handler API Changes	7
<i>Bugs Fixed in Orbix 3.0.1 and Later Versions</i>	8
<i>Known Problems, Workarounds, and Tips</i>	9
Const* Char Warning	9
SPAR Compiler Stream Bug	9
To Remove Inlining	9
Porting Existing Code	9
Use of Underscore in IORs	9
IIOP 1.1 and SSL	9
Performance of Anys	9
Length of Configuration Strings	10
<b>Orbix Code Generation ToolKit 3.0.1</b>	<b>11</b>
<i>New Features</i>	11
Java Genies	11
<i>Third Party Contributions</i>	12
<i>Incidents Cleared in This Release</i>	12
<i>Documentation Errata</i>	13
<i>Known Problems, Workarounds and Tips</i>	13
Configuration Issues	13
Setting the JDK Location	13
C++ Wizard Genies	14
Java Genies	14
<b>OrbixCOMet Desktop 3.0.1</b>	<b>15</b>
<i>Development Environments</i>	15



---

# Orbix 3.0.2

## Release Notes

### March 2000

---

<i>New Features</i>	16
SSL Support	16
<i>Incidents Cleared in this Release</i>	21
<b>OrbixNames 3.0.1 and Later Versions</b>	<b>22</b>
<i>Development Environments</i>	22
<i>Compatibility with Other IONA Products</i>	22
<i>New Features in OrbixNames 3.0.1 and Later Versions</i>	22
SSL Support	22
-j Switch	22
Documentation	22
Configuration	22
<i>Incidents Cleared in OrbixNames 3.0.1 and Later Versions</i>	24
<i>Known Problems, Workarounds and Tips</i>	25
OrbixNames Utilities	25
Interoperability Problem	25
Java Runtime Environment Problem	25
Changing the OrbixNames Port Number	25
<b>Orbix Wonderwall 3.0.1</b>	<b>26</b>
<i>Licensing</i>	26
<i>Development Environments</i>	26
<i>Compatibility</i>	26
<i>New Features in Orbix Wonderwall 3.0</i>	26
NT-Specific Changes	26
SSL-Tagged Component Support in iortool	27
SSL Details Added to Configuration GUI	27
<i>Incidents Cleared in Orbix Wonderwall 3.0</i>	27
<i>Known Problems, Workarounds and Tips</i>	27
Fragmented Replies and HTTP Tunneling	27
Timing Out of Servers with Transformers	27
Host Names and Orbix 2.3c	27
Contacting an Unregistered Server	28
<b>Further Information</b>	<b>29</b>

## Introduction

This document contains information about Orbix 3.0.1 and later versions, 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 3.0.2
- Orbix Code Generation Toolkit 3.0.1
- OrbixCOMet 3.0.1
- OrbixNames 3.0.2
- Orbix Wonderwall 3.0.1

## Orbix 3.0.1 and Later Versions

This section describes changes made in Orbix 3.0.1.

## Development Environments

This section describes the compiler and operating system versions that Orbix 3.0 has been built and tested with. The following applies to both multi-threaded and single-threaded variants of Orbix 3.0.

### Solaris 7

Orbix 3.0.2 has been built on Solaris 7 using SPARC C++ compiler version 5.0 using the following patches

- 106327-05 - Solaris 7 Patch to C runtime
- 107295-01 - Patch for WorkShop IPE 5.0
- 107355-02 - Patch for dbx
- 107289-03 - Patch for C 5.0 compiler
- 107311-06 - Patch for C++ 5.0 compiler
- 107357-04 - Patch for Compiler Common

These patches should be installed on your system.

### Solaris 2.5.1

Orbix 3.0.1 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<sup>th</sup>.

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 was carried out on Solaris 2.5.1 and 2.6 and SPARC compiler version 4.2.

The Orbix 3.0.1 libraries contain RTTI information.

### **Year 2000 compliance**

If you are using either Solaris 2.5.1 or 2.6, the Solaris recommended patch cluster should be installed as well as the patches below. More information about the recommended patch cluster is available at:

<http://online.sunsolve.sun.co.uk/pub-cgi/uk/pubpatchpage.pl>

### **Solaris 2.5.1 Y2K Patches**

- 103948-02 SunOS 5.5.1: accounting patch
- 103566-40 OpenWindows 3.5.1: Xsun patch
- 104463-03 SunOS 5.5.1: /usr/bin/date patch
- 104490-05 SunOS 5.5.1: ufsdump and ufsrestore patch
- 104816-01 SunOS 5.5.1: usr/sbin/sar patch
- 104818-01 SunOS 5.5.1: /usr/bin/passwd patch
- 104820-01 SunOS 5.5.1: /usr/lib/saf/listen patch
- 104822-01 SunOS 5.5.1: usr/lib/libadm.so.1 and usr/lib/libadm.a patch
- 104824-01 SunOS 5.5.1: usr/vmsys/bin/initial patch
- 104854-02 SunOS 5.5.1: troff macro patch
- 104873-04 SunOS 5.5.1: /usr/bin/uustat and other uucp fixes
- 105016-01 SunOS 5.5.1: usr/lib/libkrb.a and usr/lib/libkrb.so.1 patch
- 105675-01 SunOS 5.5.1: /usr/sbin/auditreduce patch
- 105701-02 SunOS 5.5.1: sysidsys unzip patch
- 104918-01 OpenWindows 3.5.1: y2000 filemgr patch
- 104995-01 OpenWindows 3.5.1: imagetool patch
- 104093-07 OpenWindows 3.5.1: mailtool patch
- 104977-01 OpenWindows 3.5.1: perfmeter patch

### **Solaris 2.6 Y2K Patches**

- 105210-18 libc & watchmalloc patch
- 105393-07 at and cron utility patch
- 105464-01 OpenWindows 3.6: multiple xterm fixes
- 105621-09 libbsm patch

105800-05 /usr/bin/admintool patch

106193-03 sysid unzip patch

106828-01 /usr/bin/date patch

## HP-UX 10.20

Orbix 3.0.1 has been built and tested on HP-UX10.20 using the Cfront compiler version A.10.36 with patch phss-13124 installed. This product has also been built and tested with ANSI compiler A.01.18.

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.

### Threading Support on HP-UX

In order to use Orbix on HP-UX, the operating system threading support should have been installed using is the 'install and core OS' for HP-UX 10.20 CD and the subset DCE programming environment.

#### Year 2000 compliance

The following patch clusters should be installed on HP:

##### 10.20 Series 700

Y2K1020S700

##### 10.20 Series 800

Y2K1020S800

## 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.

### HP-UX 11.00 Patches

The following patch bundles were used for HP-UX 11.00:

64-bitDevLibs	B.11.00	Cross Platform Development Kit
B3901BA	B.11.00	HP C/ANSI C Developer's Bundle for HP-UX 11.00 (S800)
B3913DB	B.11.01.06	HP aC++ Compiler (S800)
B5456CA	C.01.17.01	HP-UX Development Kit for Java*
B5457DA	C.01.15.04	HP-UX Runtime Environment for Java*

## Orbix 3.0.2 Release Notes

---

B5458DA	C.01.17.02/tmp/117r	HP-UX Runtime Environment for Java*
HPUXEng32RT	B.11.00	English HP-UX 32-bit Runtime Environment
KernDev	B.11.00	HP-UX Kernel Developers Kit
XSWGR1100	B.11.00.43	HP-UX Extension Pack, December 1998
Y2K-1100	B.11.00.B0315	HP-UX Core OS Year 2000 Patch Bundle

In addition, the following were patches were used for HP-UX 11.00:

PHCO_12772	B.11.00.00	HP AutoRAID Manager delivery
PHCO_13204	1.0	cumulative PFS patch
PHCO_13213	1.0	Patch for /sbin/init
PHCO_13214	1.0	cumulative newgrp(1) patch
PHCO_13217	1.0	csh(1) patch
PHCO_13279	1.0	Fix install/configure for patches
PHCO_13283	1.0	libc cumulative patch
PHCO_13284	1.0	cumulative 10.20 libc compatibility support
PHCO_14585	B.11.00.07	HP AutoRAID Manager cumulative patch
PHCO_18102	1.0	cumulative 10.20 libc compatibility support
PHCO_18132	1.0	Year 2000 HP-UX Operating System Patch Tool
PHKL_13052	1.0	pstat(2) number of processors limit removal
PHKL_13200	1.0	JFS fixes for panics and a hang
PHKL_13201	1.0	JFS vx_olt_ilstadd() panic fix
PHKL_13202	1.0	JFS extended map inodes cause fsck failure
PHKL_13219	1.0	64 Bit HPA3519A Graphics Patch
PHKL_13290	1.0	32bit binary compatibility on 64bit systems
PHNE_13076	1.0	LAN product patch
PHNE_13208	1.0	fixes for NFS system and processes hang
PHNE_16017	1.0	NFS Kernel General Release/Performance Patch
PHSS_13390	1.0	HP aC++ runtime library components (A.03.05)
PHSS_16587	1.0	HP aC++ runtime libraries (aCC A.03.13)
PHSS_17571	1.0	ld(1) and linker tools cumulative patch

## IRIX 6.5

Orbix 3.0.1 has been built and tested on IRIX 6.5 using the MIPSpro compiler 7.2.1.

## 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.

## Win-95 and Win-98

Orbix 3.0.1 supports development and deployment of clients on these platforms. This includes clients that receive callbacks from servers.

## New Features in Orbix 3.0.1 and Later Versions

This section describes new features added to Orbix 3.0.1 and later versions.

### Server Context Handler API Changes

Testing of thread-per-request filters and service context handlers uncovered a flaw in the thread safety of the current mechanism. As a result of this flaw, requests can corrupt service context data for other requests in certain conditions.

There is now a new service context handler class:

```
class ITDECLSPEC ServiceContextHandler {
private:
CORBA::ULong m_context_id;
public:
~ServiceContextHandler() {};
ServiceContextHandler ( CORBA::ULong ContextId,
CORBA::Environment &Env = CORBA::IT_chooseDefaultEnv ());

virtual CORBA::Boolean incomingRequest
CORBA::Request&incomingRequest,
CORBA::Environment &Env = CORBA::IT_chooseDefaultEnv ( )
{ return 1;}
virtual CORBA::Boolean outboundRequest (
CORBA::Request &incomingRequest, CORBA::Environment &Env =
CORBA::IT_chooseDefaultEnv ( )
{ return 1;}
virtual CORBA::Boolean incomingReply (
CORBA::Request &incomingRequest,
CORBA::Environment &Env = CORBA::IT_chooseDefaultEnv ( )
{ return 1;}
virtual CORBA::Boolean outboundReply (
CORBA::Request &incomingRequest,
CORBA::Environment &Env = CORBA::IT_chooseDefaultEnv ( )
{ return 1;}
virtual CORBA::ULong context_id()
{ return m_context_id; }
};
```

Refer to the Orbix directory demos/servicecontext for an example of how to use this class.

## Bugs Fixed in Orbix 3.0.1 and Later Versions

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:

- **Incident ID**

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.

- **PR Number**

Not all bugs fixed have a PR number (the number assigned by IONA support when a call is logged).

- **Synopsis**

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

Incident ID	PR Number	Synopsis
51970	220557	Forward declaration of interfaces in different IDL files in Orbix 3.0c creates duplicate code inside .hh file.
51984	220540 220878 221122 221368 221888 221904 223531	HP-UX product only: shared library hard-coded into Orbix 3.0 binaries.
52028		<code>resolve_initial_references()</code> does not expand <code>IT_NAMES_SERVER_HOST</code> from <code>IT_LOCAL_DOMAIN</code> .
52059		An endian problem occurs in interoperability between Orbix and OrbixWeb when an IOR is encoded on the wire followed by a long integer.
52068		Orbix now allows users to specify the IOR returned by <code>resolve_initial_references()</code> . For example, to add an IOR for a naming service (this can be for an OrbixNames server or any other server) the following entry can be added to the Orbix configuration information:  <code>Common.Services.NameService="IOR:...";</code>  When the application then calls <code>resolve_initial_references()</code> , the ORB will return a proxy for this IOR. This will work for any other service. Replace 'NameService' in the configuration entry with the service name that will be passed to <code>resolve_initial_references()</code> .
52238		The daemon was statically linked on NT.  Persistent Java servers were not recognized by the Orbix daemon.

## Known Problems, Workarounds, and Tips

This section summarizes known issues and tips relating to Orbix 3.0.1 and later versions.

### Const\* Char Warning

Orbix 3.0.2 for Solaris 7: A const \* char warning may be flagged in the server side `_DISPATCH` routine. This will be corrected in future releases.

### SPARC Compiler Stream Bug

Orbix 3.0.2 for Solaris 7: In the demos in `opt/iona/demos` (if `opt/iona` is your `root` directory) you might need a double return to enter values in the demos. This is due to a bug in the SPARC compiler stream files in the `getline` function. This is a known error and the equivalent bug id is Bug Id: 4280541.

### To Remove Inlining

Orbix 3.0.2 for Solaris 7: When compiling demos use a `-d` flag in the Makefiles to get rid of inlining.

### Porting Existing Code

Orbix 3.0.2 for Solaris 7: You must regenerate stub code when porting existing code.

### Use of Underscore in IORs

Use of the underscore character is not supported in IORs.

### IIOP 1.1 and SSL

IIOP 1.1 IOR profiles do not contain the SSL tagged component.

### Performance of Anys

In Orbix 3.0 there is a known performance issue relating to the use of the `CORBA::Any` data type. This issue has been partially addressed in Orbix 3.0.2

Check the IONA knowledge base for updates at:  
[www.iona.com/online/support/kb/index.html](http://www.iona.com/online/support/kb/index.html).

## Length of Configuration Strings

The configuration files used by Orbix consist mainly of "name = value" pairs where the "value" is written as a string. The maximum length of a string literal is 256 characters. If you need to specify a longer string, you can do this by concatenating several string literals together with the "+" operator. For example:

```
foo = "the way to represent a" + "very long string is to" +  
      "split it up into" + "smaller strings and then" +  
      "concatenate them together with the + operator";
```

This limitation that a string literal cannot be longer than 256 characters will be most noticeable if you want to embed an IOR inside a configuration file. In this case, you might need to split the IOR into several strings and join them together with the "+" operator, as illustrated in the above example.

# Orbix Code Generation ToolKit 3.0.1

This section describes changes made in Orbix Code Generation Toolkit 3.0.1.

## New Features

The following new features have been added in this release.

### Java Genies

The Orbix Code Generation Toolkit includes Java code generation genies for use with OrbixWeb. Using these genies, you can generate start point code for client-server applications, or generate complete client-server examples, based on the input IDL.

In addition to the application Java source files, the genie also generates a makefile that facilitates building the application. The makefile can be in `nmake` or `make` format. You should therefore have one of these installed.

Client-server applications in Java can lead to large projects with extensive source trees. To facilitate ease of building (and rebuilding), a Java dependency generator utility called JavaDeps is included in this release (see "Third Party Contributions" on page 12). This is a pure Java application that takes as input a list of Java sources in your project and generates a makefile dependency list for inclusion in the generated makefile. Any changes to the existing source tree only require the changed sources and their dependents to be recompiled, thereby saving a considerable amount of time. The dependency file need only be generated once provided there are no additional sources added or removed from the tree.

### Building a Java Application

To build a typical Java application, perform the following steps:

1. Enter the following command:

```
idlgen java_genie.tcl -all t.idl
```

This produces the following output:

```
t.idl:
idlgen: creating MyPackage/Config.java
idlgen: creating idlgen/PrintFuncs.java
idlgen: creating idlgen/MyPackage/foo/Printbar.java
idlgen: creating idlgen/RandomFuncs.java
idlgen: creating
idlgen/MyPackage/foo/Randombar.java
idlgen: creating idlgen/RandomMyPackage.java
idlgen: creating MyPackage/foo/barCaller.java
idlgen: creating MyPackage/foo/barLog.java
idlgen: creating MyPackage/foo/_clt_op_barLog.java
idlgen: creating MyPackage/foo/_srv_op_barLog.java
idlgen: creating MyPackage/fooLog.java
idlgen: creating classes/MyPackage.cfg
idlgen: creating MyPackage/foo/barImpl.java
idlgen: creating MyPackage/foo/Smartbar.java
```

```
idlgen: creating
MyPackage/foo/Smartbar_Factory.java
idlgen: creating MyPackage/Loader.java
idlgen: creating client.java
idlgen: creating classes/MyPackage.html
idlgen: creating server.java
idlgen: updating Makefile
```

2. To generate the makefile dependency list, enter `make depend`. This performs the following:

```
C:\Iona\bin\idlj -N -jP MyPackage t.idl
```

```
IDL Compiler: t.idl
perl C:\Iona\bin\create_src_list.pl .
C:\Iona\IDLgen\src\java\idlgen
C:\jdk1.1.8\bin\java -classpath
C:\Iona\bin\JavaDeps\jdeps.jar;..\classes;C:\Iona\lib\OrbixWeb.jar;C:\Iona\lib\uitoolkit.jar;C:\jdk1.1.8\lib\classes.zip smr.JavaDeps.JavaDeps -d .\classes -f .\java_src.lst > .deps
```

3. To build the complete project, enter `make`.
4. To register the server persistently, enter `make putit`.
5. To run the server persistently, enter `make runserver`. This is best done in separate window.
6. Enter `make runclient`.

All IDL compiler-generated sources are placed in the default output directory `java_output/NoPackage` and belong in the package `NoPackage`. (this may be changed by editing the `idlgen.cfg`). In addition, a sub package can be created by specifying a package name with the IDL compiler `-jP <package name>` option.

---

**Note:** Users with MSVC++ installations can use `nmake` in place of `make` in the steps outlined in this section.

---

## Third Party Contributions

This release of the Orbix Code Generation Toolkit contains Win32 ports of the GNU C++ preprocessor and GNU make utilities from the Cygwin project at Cygnus ([www.cygwin.com](http://www.cygwin.com)). The JavaDeps utility from Steve Robbins ([steve@nyongwa.montreal.qc.ca](mailto:steve@nyongwa.montreal.qc.ca)) is included also. The JavaDeps Home Page can be found at the following URL:

<http://www.cs.mcgill.ca/~steve/software/JavaDeps/>

## Incidents Cleared in This Release

This section describes the incidents cleared in this release. All incidents are cross platform unless otherwise stated. The incidents are described in terms of **Incident ID**, **PR Number**, and **Synopsis**, as described on page 8.

Incident ID	PR Number	Synopsis
-------------	-----------	----------

51929 - Incorrect group permissions set on Solaris package installation.

## Documentation Errata

This section details some documentation errors in the 3.0c Orbix Code Generation Toolkit Programmer's Guide.

p.50 – arguments to `idlgen -jp` should be `'-jP'`

This page details making an application with the Java genies which is incorrect, please see section *Building an Application* above for details on how to make a genie generated application.

p.53 – The Java genie does not come with a `-batch` option. Also the `-jp` option should read `-jP`.

pp.54,62,64 – The `'-jp "myPackage" '` should read `'-jP "myPackage" '`

p.61 – The `java_genie` does not come with a `-batch` option. Also the `-jp` option should read `-jP`.

p.246 – Section title "Equality Functions". Opening line mentions `cpp_print`, `cpp_random` and `cpp_equal`. Should be `java_print`, `java_random` and `java_equal` respectively.

## Known Problems, Workarounds and Tips

This section summarizes known problems, workarounds and tips with the Orbix Code Generation Toolkit. A list of the known limitations of IDLgen is provided in Chapter 1 of the *Orbix Code Generation Toolkit Programmer's Guide*.

## Configuration Issues

On Win32 platforms, update the `PATH` environment variable to include the following path:

```
orbix_install_dir\contrib\Cygwin
```

This allows you to access the third-party utilities.

On Win32 platforms, edit the `idlgen.preprocessor.cmd` option in the `idlgen.cfg` file. Change this command to the following:

```
cmd = "<orbix_install_dir>\contrib\Cygwin\cpp.exe";
```

## Setting the JDK Location

It is important to set the install location details of your JDK appropriately in the file `idlgen.cfg`. To do this, edit the Java section of the file `idlgen.cfg` located in the `OrbixOTM config` directory.

## C++ Wizard Genies

On Win32 platforms, the C++ wizard genies `cppwfull.tcl` and `cppwpre.tcl` are incorrectly located in the `genies` directory. They should be located in a sub-directory of the `genies` directory called `wizards`. To rectify this:

1. Change directory to `IDLgen\genies`
2. `mkdir wizards`
3. `move cpppre.tcl wizards`
4. `move cppfull.tcl wizards`

## Java Genies

For users of the Java genies, the class path is incomplete in the generated makefile. Consequently, you must edit the file `gen_makefile.bi` located in the following directory

### Win32

```
orbix_install_dir\IDLgen\genies\java_genie
```

### Solaris

```
orbix_install_dir/IDLgen/genies/java_genie
```

Edit the file as follows:

1. Add the following to line 171 (which has the `OW_OW_JAR` definition):

```
OW_NS_JAR =  
@[gen_dirname $(OW_JAR_DIR)/OrbixNames.jar $platform@
```

2. Change line 175 from:

```
OW_CLASSES = $(OW_OW_JAR)
```

to:

```
OW_CLASSES = $(OW_OW_JAR)@$SEP$(OW_NS_JAR)
```

3. Change the `runserver` target on line 344 from:

```
$(JAVA) $(PKG_NAME).server $(CLASS_DIR)/$(PKG_NAME).cfg
```

to:

```
$(JAVA) $(JAVA_FLAGS) $(PKG_NAME).server  
$(CLASS_DIR)/$(PKG_NAME).cfg
```

4. Change the `runclient` target on line 340 (unmodified) from:

```
$(JAVA) $(PKG_NAME).client localhost true
```

to:

```
$(JAVA) $(JAVA_FLAGS) $(PKG_NAME).client localhost true
```

## Licensing

You must install the license key supplied in order to use the Orbix Code Generation Toolkit. You can install the key with the executable `licence.exe`, which is in the `bin` subdirectory of the OrbixWeb installation directory.

The executable file to be licensed is named `idlgen.exe`, and can be found in the same directory.

# OrbixCOMet Desktop 3.0.1

This section describes OrbixCOMet Desktop 3.0.1.

## Development Environments

OrbixCOMet 3.0.1 has been built and tested with Orbix 3.0.1 and OrbixWeb 3.2. This release of OrbixCOMet completely supercedes any previous release of OrbixCOMet.

OrbixCOMet Desktop 3.0.1 is built and tested on Windows 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 versions of Windows NT earlier than 4.0. This is because DCOM is not available on those versions.

OrbixCOMet Desktop supports both the Automation/CORBA mapping and the COM/CORBA mapping as specified in the COM/CORBA Interworking Document (*ORBOS-97-09-01*).

### Automation Clients

Orbix COMet Desktop 3.0.1 has been tested with Automation client applications built with the following:

- PowerSoft PowerBuilder Version 6.0
- Borland Delphi 3/4

When using Delphi4, Inprise recommend that you make a call to `Application.Initialise()`; before making any COM calls. This includes any calls to OrbixCOMet.

- Microsoft Visual Basic Version 5.0 (SP3)
- Microsoft Visual Basic Version 6.0
- Microsoft Visual C++ 6.0 (SP2)
- Microsoft Excel97
- Microsoft Internet Explorer 4.0 or higher with VBScript

### COM Client Applications

Orbix COMet Desktop 3.0.1 has been tested with COM client applications built with the following:

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

### CORBA Server Applications

Orbix COMet Desktop 3.0.1 has been tested with CORBA server applications built with the following:

- Orbix 2.3c03
- Orbix 3.0
- Orbix 3.0.1
- OrbixWeb 3.2

### CORBA Client Applications

Orbix COMet Desktop 3.0.1 has been tested with CORBA client applications built with the following:

- Orbix 3.0.1

OrbixWeb 3.2 (using JDK 1.1.7)

## New Features

This section describes new features added to OrbixCOMet 3.0.1.

### SSL Support

OrbixCOMet 3.0.1 has been fully integrated with OrbixSSL 3.0. When OrbixSSL 3.0 is installed and SSL support is enabled, OrbixCOMet 3.0.1 supports secure communications using SSL.

OrbixCOMet's new SSL support opens up the domain of SSL-secured CORBA clients and servers to COM/Automation clients and servers. With OrbixCOMet applications using SSL support, all on-the-wire communication using IIOP is secure. Consider the OrbixCOMet recommended deployment scenario of COM/Automation clients or servers using OrbixCOMet in-process and connecting to an SSL-secured CORBA client or server. In this scenario, all on-the-wire communication is performed using SSL-secured IIOP.

This new feature provides OrbixCOMet SSL applications with the cornerstone security attributes of Authentication, Privacy and Integrity with little or no extra code.

The use of secure IIOP between OrbixCOMet and CORBA clients and servers does not have any impact on deployment scenarios where DCOM on-the-wire is also used (that is, where OrbixCOMet is used out-of-process by the COM/Automation clients or servers).

### Making OrbixCOMet Applications Secure

SSL support is added to OrbixCOMet applications using a combination of calls to a new OrbixCOMet interface (D)IOrbixSSL and by specifying configuration information in the COMetSSL configuration scope (see `orbixSSL.cfg`). In the case of CORBA clients talking to existing DCOM servers, SSL support can be added

using COMetSSL configuration scope settings alone.

The key attribute of an SSL-secured application is its association with an X.509 certificate. A trusted Certificate Authority signs an X.509 certificate. The association between an SSL application and a trusted X.509 Certificate enables authentication of the application. Stored within each certificate is an encrypted private key. A private key is used in conjunction with a public key to facilitate privacy of communication as the private key/public key combination is used to encrypt the messages between the client and server. To retrieve (and decrypt) the private key stored within the certificate requires the use of a private key password.

Hence an application with access to a certificate and the password for the certificate's private key can be initialized as an SSL application. Of course, clients and servers can inspect certificates presented from SSL-enabled applications and reject connection attempts based on custom criteria.

The new OrbixCOMet SSL interface ((D)IOrbixSSL) can be used by COMet applications to specify the password/certificate combination that is to be used to SSL-enable the application. A pointer to the OrbixCOMet SSL interface is obtained by a call to GetOrbixSSL on the existing (D)IOrbixORBObject interface. Applications must then call (D)IOrbixSSL::InitSSL to initialize the SSL library. (See the Demos section below for a list of OrbixCOMet SSL-enabled demos). The process of specifying the private key password/ X.509 certificate combination to be used by the application is achieved by calling (D)IOrbixSSL::SetPrivateKeyPassword and (D)IOrbixSSL::InitScopeSSL.

InitScopeSSL takes as a parameter an IONA configuration scope parameter. For example, `objSSL.InitScope("COMetSSL.Demos")`.

This parameter identifies a scope in the OrbixSSL configuration file that contains a specification of the SSL security policy settings that will be implemented by OrbixSSL on behalf of users of this policy (For reference consult OrbixSSL documentation). If this policy contains a value for the configuration setting `IT_CERTIFICATE_FILE`, this is the certificate that SSL associates with the application as a result of the call to InitScopeSSL. Scopes and policies can be specified on a per-application basis in the OrbixSSL configuration file. From the following OrbixSSL configuration file excerpt, you can see the specification of the certificate associated with the scope `COMetSSL.Demos`. All of the OrbixCOMet SSL demos use this scope.

```
COMetSSL
{
  # This cert is used by COMet SSL clients and servers
  Demos
  {
    # Password for the demosever cert is
    "demoseverpassword"
    IT_CERTIFICATE_FILE= _ssl_dir +
    "\certs\demo\demosever";
  };
};
```

Alternatively, the file specification of the certificate can be passed explicitly as a parameter to the (D)IOrbixSSL::SetSecurityName method.

Note that the password must be specified before the certificate is loaded by the SSL library as a result of a call to InitScopSSL or SetSecurityName. Also, private key passwords should not be "hardcoded" into applications. Instead, where necessary users should be prompted to enter their passwords.

As with all OrbixSSL applications, OrbixCOMet clients and servers are required perform their SSL initialization before either binding to secure CORBA Servers or registering themselves as secure CORBA Servers.

### OrbixCOMet SSL Handler DLLs

OrbixCOMet Handler DLLs can be used to inject extra custom SSL functionality into COMet applications. OrbixSSL provides the facility to register a C/C++ callback function that will be invoked by OrbixSSL during client/server authentication. This callback function is passed details of the X.509 certificate of the application being accessed. Custom checks can then be made on the certificate in the implementation of the callback function. If the callback function returns `TRUE`, this signals the SSL that the certificate is acceptable. If it returns `FALSE`, OrbixSSL aborts the connection attempt, throwing an authorization failure exception.

This is an example application of OrbixCOMet SSL Handler DLLs that is illustrated in the SSLHandler demo contained in the `corbasrv` demo directory. This Handler DLL is used by both the secure COM grid client and the secure VB Bank client to check that the certificate used by their respective servers is the demosever certificate.

### Secure CORBA Clients Accessing Existing DCOM Servers

OrbixCOMet SSL support also enables secure CORBA clients to connect to existing DCOM servers (that is, third-party DCOM servers). Currently, OrbixCOMet facilitates CORBA clients connecting to third-party DCOM servers using the OrbixCOMet Generic Custom Surrogate (`CustSur.exe`) which acts, in this scenario, as a CORBA server (see the OrbixCOMet documentation for further details).

If the CORBA client is secure, COMet (hosted by `CustSur.exe`) needs to initialize itself as a secure CORBA server. To instruct an instance of `CustSur` to initialize itself as a secure CORBA server, the `-1` switch must be passed to `CustSur` as a command line parameter.

For example, the current fortune demo CORBA client makefile registers `CustSur` as a CORBA server with the following command:

```
putit fortune "C:\IONA\COMet_3.0\custsur.exe -t 20000"
(-t — specify server time out in milliseconds)
```

To make the fortune server secure simply requires the `-1` switch to be added to the command line:

```
putit fortune "C:\IONA\COMet_3.0\custsur.exe -1 -t 20000"
```

To run the fortune server persistently, simply add the `-1` switch to the command line in addition to the `-s` switch (required to specify the server name):

```
C:\IONA\COMet_3.0\custsur.exe -1 -s "fortune" -t 20000
```

Instructing `CustSur` to initialize itself as a secure CORBA server does not complete the entire process. `CustSur`, like any other secure CORBA server, must be associated with an X.509 certificate and have access to the corresponding private key password.

The association between an instance of `CustSur` — impersonating a particular CORBA server — and a certificate is made using the entries in the `COMetSSL.CustSur` scope within `orbixSSL.cfg`. This scope has entries for each CORBA server that `CustSur` is set up to impersonate. Each SSL policy within this

scope is identified by the server name. Therefore, for the above example, when CustSur is asked to initialize itself securely as the fortune CORBA server, it looks for a certificate and other SSL security policy configuration information within the configuration scope `COMetSSL.CustSur.fortune`.

Secure CORBA servers have two options for retrieving passwords at run time. The first is through user input where the server prompts the user to specify the private key password for the certificate when it is launched. Alternatively, the server receives a password from the KDM at start-up time. CustSur supports both methods.

The KDM (Key Distributed Management) is a persistent CORBA server supplied with OrbixSSL (see the OrbixSSL documentation). It provides a mechanism whereby passwords can be securely and dynamically supplied to servers at server start-up time, thereby negating the requirement for user intervention through password entry. The KDM maintains an encrypted database that stores server names and private key password pairs. Entries can be added to the KDM's database using the `putKDM` CORBA client utility. For example:

```
putKDM fortune demoseverpassword
```

The KDM must be started persistently before the Orbix daemon is started. When the Orbix daemon launches a server, it checks to see if there is an entry for that server in the KDM. If there is, then the password is sent on to the OrbixSSL runtime in the server.

CustSur's SSL initialization code checks to see if it has received a password from the KDM via the Orbix daemon. If no password has been obtained from the KDM (see `(D)IOrbixSSL:HasPassword`), CustSur displays a dialog prompting the user for a password.

### TypeMan and the secure IFR

TypeMan is an IFR client. If the IFR is configured to run as a secure CORBA server, TypeMan needs to operate as a secure CORBA client when it is used to look up type information in the IFR.

To SSL-enable TypeMan, set the OrbixCOMet configuration setting (in `config\OrbixCOMet.cfg`) to

```
COMet.TypeMan.TYPEMAN_SSL_ENABLED = 'yes'  
(default is 'no').
```

The X.509 certificate used by TypeMan is specified by the SSL security configuration policy `COMetSSL.TypeMan`.

### OrbixCOMet SSL Demos

Secure Orbix CORBA Grid client/server (`demo\corbasrv\ssl_grid`).

An SSL-enabled version of the standard C++ grid demo.

Secure Orbix CORBA Bank client/server (`demo\corbasrv\ssl_idl_demo`).

SSL-enabled version of the standard C++ bank demo.

Visual Basic Bank client (`demo\vb\bank`).

The existing COMet bank client. This demo can now be initialized as a

secure client by ticking a checkbox.

Visual Basic Bank server (demo\vb\bank).

The existing COMet bank server, now with the ability to serve secure bank clients by ticking a checkbox.

OrbixCOMet SSL Handler DLL (demo\corbasrv\SSLHandler).

Demonstrates use of a COMet Handler DLL to inject extra OrbixSSL C++ authentication functionality into COMet applications. The secure VB Bank client and the secure COM grid client use this handler DLL.

COM Grid client (demo\com\ssl\_grid).

SSL-enabled version of the standard C++ COM grid client.

Secure Orbix CORBA fortune client using existing DCOM Server (demo\corbaclient\ssl\_fortune).

Secure CORBA client which illustrates the use of CustSur as a generic secure CORBA server.

### OrbixCOMet SSL Interface (D)IOrbixSSL

InitSSL ([in,out] VARIANT \*IT\_Ex, [retval,out] int\* nRet )

Initializes SSL library. Must be called before any other SSL methods.

InitScopeSSL ([in] BSTR cPolicyName, [in,out] VARIANT \*IT\_Ex, [retval,out] int\* nRet )

Used to specify the OrbixSSL configuration scope from which to read the SSL security policy for the application.

SetSecurityName ( [in] BSTR cCertName, [in,out] VARIANT \*IT\_Ex, [retval,out] int\* nRet )

Used to explicitly specify the file spec of the X.509 certificate to be used by the application. The private key password for an application must be specified before calling SetSecurityName or InitScopeSSL.

GetSecurityName ( [in,out] VARIANT \*IT\_Ex, [retval,out] BSTR\* cCertName )

Retrieves the file spec of the X.509 certificate being used by the application.

SetPrivateKeyPassword ( [in] BSTR cPassword, [in,out] VARIANT \*IT\_Ex, [retval, out] int\* nRet )

Used to specify the password to be used to decrypt the private key contained within the application's certificate.

HasPassword ( [optional,in,out] VARIANT \*IT\_Ex, [retval,out] VARIANT\_BOOL \*IT\_retval)

Returns `TRUE` if a private key password for the application has already been specified, either by means of the KDM or as a result if a previous call to SetPrivateKeyPassword. Used by the VB Bank Server demo to determine if a password has been obtained from the KDM and consequently to decide whether or not to prompt the user for a password.

## Incidents Cleared in this Release

This section describes the incidents cleared in this release. All incidents are cross platform unless otherwise stated. The incidents are described in terms of **Incident ID**, **PR Number**, and **Synopsis**, as described on page 8.

Incident ID	PR 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— <code>ts2idl</code> generates wrong MIDL.
52103		Narrowing a derived object using <code>Idispatch()</code> .
52178		<code>Typeman</code> crashes when priming type store with <code>MSOUTL8.OLB</code> (Microsoft Outlook express).
52179		Problem with unions with enum discriminator.
52180		Warning messages from <code>typeman</code> when priming with some type libs.
52194		Crash on creating/returning sequence of structures containing sequences of structures.

## OrbixNames 3.0.1 and Later Versions

This section describes changes in OrbixNames 3.0.1 and later versions.

### Development Environments

Development environment information for OrbixNames 3.0.2 is the same as that described for Orbix 3.0.2 on page 3.

### Compatibility with Other IONA Products

It is not possible for OrbixNames 3.0.2 and the Java naming service supplied with OrbixWeb to share the same Bindings Repository.

## New Features in OrbixNames 3.0.1 and Later Versions

This section describes the new functionality and major changes added in OrbixNames 3.0.1 and later versions.

### SSL Support

OrbixNames can now be run securely using OrbixSSL. Before running OrbixNames securely, you must install OrbixSSL and enable SSL support.

### -j Switch

The OrbixNames server is a Java application. On platforms other than Solaris, you can instruct the server to pass command-line switches directly to the Java interpreter. To do this, use the `-j` switch to the OrbixNames server.

For example, if you want to increase the virtual memory used by the interpreter when running OrbixNames, start the server as follows:

```
ns -j -mx9000000
```

### Documentation

The OrbixNames user documentation has been updated for this release. The OrbixNames user documentation is a single volume, called the *OrbixNames Programmer's and Administrator's Guide*.

### Configuration

Each Orbix 3 service has its own configuration file. The OrbixNames configuration variables are scoped and defined in the file `orbixnames3.cfg`.

## Orbix 3.0.2 Release Notes

---

The command-line options available with OrbixNames can also be set in `orbixnames3.cfg`. The relevant variables are as follows:

Variable	Description
<code>IT_NAMES_CACHE_SIZE</code>	The number of contexts to be cached by OrbixNames. For example:  <code>IT_NAMES_CACHE_SIZE = "10";</code>
<code>IT_NAMES_DIAGNOSTICS</code>	Sets the diagnostic level within the OrbixNames server. For example:  <code>IT_NAMES_DIAGNOSTICS = "0";</code>
<code>IT_NAMES_HASH_TABLE_SIZE</code>	Sets the initial size of a context's hash table. For example:  <code>IT_NAMES_HASH_TABLE_SIZE = "23";</code>
<code>IT_NAMES_THREAD_POOL</code>	The number of threads to be created to handle the invocations to the Naming Service. For example:  <code>IT_NAMES_THREAD_POOL_SIZE = "10";</code>
<code>IT_NAMES_TIMEOUT</code>	The length of time in milliseconds after which the Naming Service will timeout. For example, to set an infinite timeout:  <code>IT_NAMES_TIMEOUT = "-1";</code>
<code>IT_NS_HASH_TABLE_LOAD_FACTOR</code>	The factor by which a context's hash table is increased when full. For example:  <code>IT_NS_HASH_TABLE_LOAD_FACTOR = "0.5f";</code>

The main configuration variable set is described in *OrbixNames Programmer's and Administrator's Guide*. In addition to these variables, it is now possible when using OrbixNames to configure the format of an IOR, with respect to its host address part. The IOR can contain either the IP address or a host name. The `OrbixNames.IT_USE_HOSTNAME_IN_IOR` variable determines this characteristic. The default value is `TRUE`. With this value, a host name appears in an IOR. Setting the value to `FALSE` causes the IOR to contain an IP address.

## Incidents Cleared in OrbixNames 3.0.1 and Later Versions

This section describes the incidents cleared in this release. All incidents are cross platform unless otherwise stated. The incidents are described in terms of **Incident ID**, **PR Number**, and **Synopsis**, as described on page 8.

Incident ID	PR Number	Synopsis
24780	160367	The command <code>del_group</code> , when used with the <code>-n</code> switch, core dumps if the name exists but the group has already been deleted.
26180	163401	The operation <code>CosNaming::NamingContext::OBfactory()</code> is not described in the documentation.
26960	163477	A port number 0 appears in the Naming Service IOR when the OrbixNames server is automatically launched.
28400	162529	The command <code>putncns</code> core dumps if incorrect parameters are specified.
30940	176192	The <code>lsns</code> command, when used with the <code>-h</code> switch, hangs the OrbixNames server if the case of the host name is incorrect.
34560	187359	The operation <code>resolve()</code> returns object references that were previously removed from the Naming Service.
36580	189815	OrbixNames 1.1 does not work with proxified IORs created by Orbix Wonderwall <code>iortool</code> .
38000	192357	The command <code>putnewncns</code> crashes when a name of more than 600 bytes is specified.
51475	216644	The marker in the <code>ObjectKey</code> in a string format IOR should be in the form <code>module/interface</code> . OrbixNames 1.1 formatted this as <code>module_interface</code> .
52119	221571	Cannot specify an IOR for any Orbix service.
52172	221731	When creating a federation of name spaces, the command <code>putncns</code> generates an <code>Unexpected system exception 12003</code> (Java exception).
52217	222042	When <code>rebind_context()</code> is passed a zero-length name, the Naming Service outputs <code>Unexpected system exception 12003</code> (Java exception).
52308	222254	The command <code>reputncns</code> does not work and gives an unexpected exception.

## Known Problems, Workarounds and Tips

This section describes the known issues and suggested workarounds for OrbixNames 3.0.1 and later versions.

### OrbixNames Utilities

The C++ OrbixNames utilities have a switch `-h`. However this switch does not work. To change the host that is to be used by a utility, explicitly set the desired host value using the variable `IT_NAMES_SERVER` in the `orbixnames3.cfg` file.

In addition, you cannot use the utility `killit` to stop the OrbixNames server when using SSL.

### Interoperability Problem

The OrbixNames server assumes the existence of a non-empty principal in IOP messages received from connecting clients. Some ORBs do not always send principal information from clients. Clients developed using these ORBs fail to connect to the OrbixNames server.

### Java Runtime Environment Problem

On HP-UX, you may encounter problems when running the server on a very heavily loaded system. If you are using an old version of the Java Runtime Environment (JRE), the server sometimes crashes. This is due to a problem in the JRE. To work around the problem, pass the `-nojit` switch to the Java interpreter as follows:

```
ns -j -nojit
```

This problem does not occur with version 1.1.6 of the JRE, which is shipped with Orbix 3.0.1 on HP-UX.

### Changing the OrbixNames Port Number

The configuration variable `IT_NS_PORT` in `orbixnames3.cfg` sets the port number on which applications communicate with the OrbixNames server. This has the same default value as the Orbix daemon port variable `IT_DAEMON_PORT`. The default is 1570. If you wish to change the OrbixNames port value, set the value of `IT_NS_PORT` in `orbixnames3.cfg`.

The `IT_NS_PORT` value affects only Java applications. In addition, if the Orbix daemon locates the OrbixNames server for your applications, ensure that the value of `IT_NS_PORT` is the same as `IT_DAEMON_PORT`. If the OrbixNames server runs without the Orbix daemon, it is not necessary to synchronize these values.

## Orbix Wonderwall 3.0.1

This section describes changes made in Orbix Wonderwall 3.0.1.

### Licensing

This release of Orbix Wonderwall requires that you license the IIOP proxy with your Orbix 3.0 license key. The installation script attempts to do this. However, if you enter an invalid license key, the proxy will fail at start-up. To enter a new license key, run the following command on UNIX platforms:

```
<Orbix Wonderwall dir>/bin/install_licence  
                                <Orbix Wonderwall dir>/iioproxy <key>
```

The equivalent command on Windows NT is:

```
<Orbix Wonderwall dir>\bin\licence.exe  
                                <Orbix Wonderwall dir>\bin\iioproxy.exe <key>
```

### Development Environments

Orbix Wonderwall supports the same environments as those described for Orbix on page 3.

To use the GUI tools, you must have either OrbixWeb 3.2 or Orbix 3.0.1 installed. You must also have specified the location of one of these products during installation. If this was not done, you must update the `ORBIX_DIR` location in your `wonderwall\demo\demo_setup.bat` file to point to the root directory of either your Orbix 3.0.1 or OrbixWeb 3.2 installation.

### Compatibility

Orbix Wonderwall is designed to interoperate with any CORBA ORB that implements version 1.0 or 1.1 of the CORBA Internet Inter-ORB Protocol (IIOP).

## New Features in Orbix Wonderwall 3.0

This section describes the new functionality and major changes added in Orbix Wonderwall 3.0.1.

### NT-Specific Changes

Wonderwall on NT is now made up of an executable and two helper DLLs. These DLLs are called the following,

```
iioproxy301.dll
```

```
itts.dll
```

When moving Orbix Wonderwall from its default location, you must ensure that you either keep the DLLs in the same directory as the executable; alternatively, they must be within your path.

## SSL-Tagged Component Support in iortool

The `iortool` utility now has an extra flag that causes it to add an SSL-tagged component to a proxified IOR. The flag is `-sslport`.

Here is an example of how you might use this feature:

```
iortool -ior -proxify -host bloom -port 16000
      -sslport 3636 \ grid.ior > grid_proxified.ior
```

## SSL Details Added to Configuration GUI

The configuration now has an SSL tab that adds configuration details that are relevant for an SSL-enabled Wonderwall.

## Incidents Cleared in Orbix Wonderwall 3.0

Incident ID	PR Number	Synopsis
52094	220640	Objects specified using the bind syntax in Orbix Wonderwall's configuration file do not work with Orbix 3.0c servers.

## Known Problems, Workarounds and Tips

This section describes known problems, workarounds, and tips for Orbix Wonderwall 3.0.1.

### Fragmented Replies and HTTP Tunneling

Sending fragmented `Reply` messages from IIOP 1.1 servers over a HTTP-tunneled connection is not yet supported.

### Timing Out of Servers with Transformers

If an activated server that requires use of a server transformer times out or is stopped, Orbix Wonderwall attempts to send a transformed message to the server's activation port. This port is associated with the `orbixd` or `orbixdj` process and causes the daemon to fail with an unmarshalling error. This in turn causes the server to be unavailable to the client.

### Host Names and Orbix 2.3c

Orbix 2.3c does not use the host name in an IOR, but uses the host name contained in the object key instead. If you intend to use Orbix 2.3c clients to contact Orbix or OrbixWeb servers behind Orbix Wonderwall, with proxified IORs, you must run the Wonderwall and server on the same host, but using different ports. This problem is fixed in Orbix 3.0.

## Contacting an Unregistered Server

The OrbixWeb 3.0 activator, `orbixdj`, produces the following stack trace if Orbix Wonderwall tries to bind to a server that is not registered in the Implementation Repository:

```
java.lang.NullPointerException
  at IE.Iona.OrbixWeb.CORBA.ServerRequest.target(ServerRequest.java)
  at IE.Iona.OrbixWeb.Activator.DJAuthenticationFilter.
    inRequestPreMarshal(DJAuthenticationFilter.java)
  at IE.Iona.OrbixWeb.CORBA.ServerRequest.
    inRequestPreMarshal(ServerRequest.java)
  at IE.Iona.OrbixWeb.CORBA.ServerDispatcher.
    dispatchSpecial(ServerDispatcher.java)
  at IE.Iona.OrbixWeb.CORBA.BOA.processRequest(BOA.java)
  at IE.Iona.OrbixWeb.CORBA.BOA.processOneEvent(BOA.java)
  at IE.Iona.OrbixWeb.CORBA.BOA.processEvents(BOA.java)
  at IE.Iona.OrbixWeb.CORBA.EventHandler.run(EventHandler.java)
  at java.lang.Thread.run(Thread.java)
```

This is fixed in OrbixWeb 3.0 patch 2 and later releases of OrbixWeb.

## 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>