

---

# Orbix 2.3.4

## Release Notes

### September 1999

---

## Contents

<b>Introduction</b>	<b>2</b>
<b>Development Environments</b>	<b>2</b>
<i>Solaris</i>	2
Year 2000 Compliance	2
Solaris 2.5.1 Y2K Patches	3
<i>NT</i>	3
<b>Compatibility with Other IONA Products</b>	<b>4</b>
<b>New Features in Orbix 2.3.4</b>	<b>4</b>
<i>Host Lookup Caching</i>	4
Caching for Clients and Servers	4
Using the Host Lookup Cache API	5
Using the Cache in the Orbix Daemon	6
<b>Bugs Fixed in Orbix 2.3.4</b>	<b>7</b>
<b>Further Information</b>	<b>14</b>

## Introduction

This Orbix 2.3.4 release covers the following platforms:

Product ID	Platform
s1178	Orbix C++ Solaris 2.5.1 Single-threaded
s1190	Orbix C++ HP-UX 10.20 Multi-threaded HP Native 10.20
s1192	Orbix C++ HP-UX 10.20 Single-threaded HP C-Front 10.22
s1193	Orbix C++ Solaris 2.5.1 Multi-threaded
s1194	AIX 4.1.5
s1228	Orbix C++ Win32
s1300	Orbix C++ HP-UX 10.20 HP ANSI C++ 1.09
S1341	Orbix C++ DEC UNIX 4.0 D
s1408	Orbix C++ for HP-UX 11.00 Multi-threaded

This document contains information about Orbix 2.3.4, including build information, information about new features, and details of bugs that have been fixed in this release. These release notes describe the changes across all platforms.

Orbix 2.3.4 has been built and tested with the relevant operating system (OS) patches for Y2K issues. This version of Orbix synchronizes the bug fixes across platforms, which simplifies compatibility issues for users.

## Development Environments

This section describes the compiler and operating system versions that Orbix 2.3.4 has been built and tested with. The following applies to both multi-threaded and single-threaded variants of Orbix 2.3.4.

### Solaris

Orbix 2.3.4 has been built and tested with Solaris 2.5.1 using the SPARC C++ compiler version 4.1.

### Year 2000 Compliance

The recommended Solaris patch cluster should be installed as well as the patches that follow. 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

## NT

Orbix 2.3.4 has been built and tested on NT 4 with service pack 3 installed, using the VC++ compiler version 5. In addition, a Microsoft Y2K patch has been installed.

Orbix 2.3.4 has also been tested on Windows NT4 with service pack 4.

For details of the Y2K issues with Windows NT, please refer to [http://www.microsoft.com/technet/year2k/product/user\\_list.asp](http://www.microsoft.com/technet/year2k/product/user_list.asp). The details of the English language version of the patch can be found at [http://www.microsoft.com/technet/year2k/product/user\\_view68488EN.htm](http://www.microsoft.com/technet/year2k/product/user_view68488EN.htm)

## Compatibility with Other IONA Products

Orbix 2.3.4 is compatible with the following IONA products:

- OrbixTalk 2.0.3 for Solaris
- OrbixTalk 2.0.3 for Win NT
- OrbixEvents 1.0.2 for Solaris
- OrbixEvents 1.0.2 for Win NT

## New Features in Orbix 2.3.4

This section describes new features added to Orbix 2.3.4.

### Host Lookup Caching

Orbix uses the `gethostbyname()` and `gethostbyaddr()` system calls, and also provides a cache for storing successful host lookups. By default, the internal host lookup cache is turned off, but you can enable it by following the steps outlined in this section.

### Caching for Clients and Servers

When the cache is enabled, you can use an API for flushing the entire contents or removing a particular value. This feature only works for clients and servers that set a variable in their environment before runtime. This feature is designed so that you can remove entries that you believe to be invalid; for example, because of using of DHCP. The removal of stale entries is the responsibility of the user.

#### Activating the Host Lookup Cache

To activate the host lookup cache, set `IT_CACHE_LOOKUP` to `YES` in your environment, using one of the following methods:

- Using the `orbix.cfg` file.
- Explicitly setting it in your environment.

These are the recommended means of turning the cache on.

### Enabling Cache Flushing

To enable the ability to flush the cache, set `IT_FLUSH_CACHE` to `YES` in your environment, using one of the following methods:

- Using the `Orbix.cfg` file.
- Explicitly setting it in your environment.

These are the only methods for allowing the user to flush the cache in their code. For example, using your `Orbix.cfg` file, the entries would be as follows:

```
IT_CACHE_LOOKUP YES
IT_FLUSH_CACHE YES
```

Using your environment, the entries would be as follows:

```
export IT_CACHE_LOOKUP=YES
export IT_FLUSH_CACHE=YES
```

### Using the Host Lookup Cache API

You can get and set the value for `IT_CACHE_LOOKUP` in your code, using the following APIs:

```
static unsigned int GetConfigValue(const char * name, char
*& value);
```

```
static unsigned int SetConfigValue(const char * name, char *
value);
```

When using `SetConfigValue()` with the name `IT_CACHE_LOOKUP`, the call may fail if a flush is in progress. It is the responsibility of the caller to check the return value.

These APIs are thread safe. You can use them as often as necessary. The following is an example of using these APIs:

```
char *cache_value;
if(CORBA::Orbix.SetConfigValue("IT_CACHE_LOOKUP", "YES"))
    cout << "Successfully turned the cache ON" << endl;
else {
    cout << "Could not turn cache on as flush is in
progress" << endl;
}
CORBA::Orbix.GetConfigValue("IT_CACHE_LOOKUP", cache_value);
if( strcmp(cache_value, "YES"))
    cout << endl << "CACHE is OFF" << endl;
else
    cout << endl << "CACHE is ON" << endl;
```

You can flush the contents of the host lookup cache in your code using the following API:

```
unsigned char CORBA::Orbix::flushLookupCache(const char* s);
```

where *s* is a string representing a host name or IP address,

or

where *s* is `NULL` when you wish to flush the entire cache.

The return value indicates a success or failure to the flush. The call fails under the following circumstances:

1. Cache entry not found.
2. Flush already in progress.
3. Flush capability not enabled.

For example,

```
if(CORBA::Orbix.flushLookupCache("rubbish"))
    cout << "Just removed rubbish from cache" << endl;
else
{
    cout << "Failed to flush rubbish from cache."
         << "Either " << endl
         << "1. Entry not found." << endl
         << "2. Flush already in progress." << endl
         << "3. Flush capability not enabled." << endl;
}

if(CORBA::Orbix.flushLookupCache()
    cout << endl << "Just flushed entire cache" << endl;
```

## Using the Cache in the Orbix Daemon

You can also activate the host lookup cache when using the Orbix daemon (`orbixd`). You must first set the value `IT_CACHE_LOOKUP` to `YES` in your environment by either using the `orbix.cfg` file or explicitly setting it in your environment.

### The Orbix Daemon -a Switch

To activate the host lookup cache, start the daemon using the `-a` switch. For example,

```
orbixd -a
```

This is not recommended if you need to turn the cache off or flush it at any stage.

## Bugs Fixed in Orbix 2.3.4

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

- **Patch Number**

This is the patch for a particular platform that included a fix for the bug referenced.

- **Bug 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.

Patch Number	Bug ID	PR Number	Synopsis
s1408-23c03-04 s1194-23c03-04 s1300-23c03-04	347	24809 168078 169199 203735	Provide <code>putid1 -I</code> functionality.
s1408-23c03-04 s1194-23c03-04 s1300-23c03-04	625	113847 121375 139390	The <code>objectKey</code> is not being reset when either the marker or server is being changed.
s1408-23c03-04 s1194-23c03-04 s1300-23c03-04	10122	30154 111224 116862 119658 135727 141076 146763 201489 209244	Orbix use of host names with IIOp is now not case sensitive in <code>bind()</code> calls and can handle aliases. This previously caused problems where the server spins and /or many connections are opened between a client and a server.
s1408-23c03-04 s1194-23c03-04 s1300-23c03-04	10460	130726 132681 148067 163419 205134	The insertion of a struct into an Any and invoking an operation with the Any passed in as a parameter to the server causes memory leaks for Orbix 2.3.

## Orbix 2.3.4 Release Notes

Patch Number	Bug ID	PR Number	Synopsis
s1408-23c03-04 s1194-23c03-04 s1300-23c03-04	10680	112424 122744 124676 129764 137066 141670 144221 144914 146203 148824 152703 158500 162955 165268 174499 185959 207545	Typecode information returned from the Interface Repository is always <code>tk_null</code> for any parameters of a method.
s1408-23c03-04 s1194-23c03-04 s1300-23c03-04	10883	121299 133072	Need to distinguish between manager and 'straightforward' types.
s1408-23c03-04 s1194-23c03-04 s1300-23c03-04	11440	112195 133663 133709 136980 138364 176974 190421	IDL compiler does not scope correct with module when multiple inheritance is used.
s1408-23c03-04 s1194-23c03-04 s1300-23c03-04	12180	116042 127405 129191 143033 147998 148626 205689	Union causes <code>put idl</code> to core dump if discriminator is enum.
s1408-23c03-04 s1194-23c03-04 s1300-23c03-04	12240	133779 148901 150943 166849 187216 214599	IDL problem with reopening modules.
s1408-23c03-04 s1194-23c03-04 s1300-23c03-04	12340	130419 140701 144060 198538	Unable to pass an object reference in a struct.
s1408-23c03-04 s1194-23c03-04 s1300-23c03-04	13120	125616	<code>CORBA::ORB::addForeignFD()</code> core dumps.
s1190-23c03-06 s1193-23c03-06	13300	25208 218217 220728 220874	Two new environment variables were added to the configuration file: <code>IT_CLIENT_SERVER_BASE</code> and <code>IT_CLIENT_SERVER_RANGE</code> . These restrict the ports a client can use when allowing Orbix clients to work through a firewall.

## Orbix 2.3.4 Release Notes

Patch Number	Bug ID	PR Number	Synopsis
s1408-23c03-04 s1194-23c03-04 s1300-23c03-04	13340	127686	TypeCode::equal() incorrectly returns false on some large typecodes.
s1408-23c03-04 s1194-23c03-04 s1300-23c03-04	13500	125046 126132 145740 145850 145877	IT_daemon::removeUnsharedMarker is corrupting the Implementation Repository file.
s1408-23c03-04 s1194-23c03-04 s1300-23c03-04	13601	128057 137878 139464 142071 224697	Leak caused when a struct with an Any is copied.
s1408-23c03-04 s1194-23c03-04 s1300-23c03-04	13800	125128 154122	Exceptions containing object reference cause client core dump.
s1408-23c03-04 s1194-23c03-04 s1300-23c03-04	13820	126667 144280 155292	IDL-generated code does not compile.
s1408-23c03-04 s1194-23c03-04 s1300-23c03-04	14001	111059 140131 186421 187306	Memory leak which occurs when an Any is inserted into an Any.
s1408-23c03-04 s1194-23c03-04 s1300-23c03-04	14081	112075 155215	When the client calls a method that should return an Any (which is a structure containing a union of structures), an exception is returned which states that there is not enough information to extract the ANY.
s1408-23c03-04 s1194-23c03-04 s1300-23c03-04	14200	27583	User-defined exceptions do not work correctly if they contain an Any.
s1408-23c03-04 s1194-23c03-04 s1300-23c03-04	14700	123818 125172	IDL -O crashes the IDL compiler.

## Orbix 2.3.4 Release Notes

Patch Number	Bug ID	PR Number	Synopsis
s1190-23c03-01 s1408-23c03-04 s1193-23c03-01 s1194-23c03-04 s1300-23c03-04	22203	153572 154532 186950 218902 221720	<p>Calling oneway methods continuously causes server to leak memory. To reduce the memory leaks caused by invoking a large number of oneway operation calls, a new API has been created which is called the <code>setMaxEventQueueSize(size)</code>;</p> <p>This API restricts the number of requests a server handles at any given time. Setting the queue size is completely under the server's control. It inherits from <code>CORBA::BOA</code>.</p> <p>Example:</p> <pre>// // Server side // In srv_main.cc file //</pre> <p>Before calling <code>impl_is_ready()</code> must declare what the <code>qsize</code> will be:</p> <pre>unsigned short qsize = 100;  CORBA(Orbix).setMaxEventQueueSize(qsize);</pre> <p>If you set this and run <code>top</code>, you will see the difference in memory leakage.</p>
s1190-23c03-06 s1193-23c03-03	24120	152673 165389 176586 214692 217692 219890 224632	<p><code>USER_EXCEPTIONS</code> caused the server to core dump when an operation took an invalid object reference.</p>
s1193-23c03-10	33800	176129 180655 186600 196850 220802	<p>When a nil object reference is passed as a parameter, the unmarshalling code in the receiving side creates a null proxy. When this null proxy is created, its reference count should be 1. However, the null proxy is actually created with a reference count of 0.</p>
s1190-23c03-02 s1408-23c03-04 s1193-23c03-02 s1194-23c03-04 s1228-23c03-02	35984	186186 216570	<p>Piggyback data not cleared after request data is shorter than previous invocation (uses stale data).</p>
s1190-23c03-01 s1408-23c03-04 s1193-23c03-01 s1194-23c03-04 s1300-23c03-04	38320	195367	<p><code>putidl</code> core dumps when dealing with a complex data structure.</p>
s1193-23c03-05	40080	200983	<p>Client or server crashes when running multiple threads.</p>

## Orbix 2.3.4 Release Notes

Patch Number	Bug ID	PR Number	Synopsis
s1190-23c03-02 s1408-23c03-04 s1193-23c03-02 s1194-23c03-04 s1300-23c03-04	42240	203955	Problem when putting a <code>port 0</code> in IOR when binding to Naming Service (POOP protocol).
s1190-23c03-03 s1408-23c03-04 s1193-23c03-02 s1194-23c03-04 s1300-23c03-04	43440	211834 215214	Two foreign IORs with different object keys hash to the same proxy in the COT table.
s1190-23c03-01 s1408-23c03-04 s1193-23c03-01 s1194-23c03-04 s1300-23c03-04	44780	188447 205124 209150 218942	UDP messaging problem: client process hangs if another client on another machine terminates (CTRL-C).
s1190-23c03-03 s1408-23c03-04 s1193-23c03-02 s1194-23c03-04 s1300-23c03-04	44920	207419	Server has memory leak when client exits; when using the TIE approach, thread filter and <code>cond_wait()</code> .
s1190-23c03-03 s1408-23c03-04 s1193-23c03-02 s1194-23c03-04 s1228-23c03-02	47400	206309 207810 224538	Timeout value for non-existent function does not work.
s1190-23c03-06  s1193-23c03-06	47760	205690	When SSL libraries are used, the <code>psit</code> command does not show an auto-launched server.
s1408-23c03-04 s1194-23c03-04 s1300-23c03-04	47940	N/A	IOR callback port bug that would cause callback requests on OrbixSSL C++ clients to fail if bi-directional IOP was not being used. An Orbix server that attempts to callback this client would instead contact the daemon.
s1190-23c03-01 s1408-23c03-04 s1193-23c03-01 s1194-23c03-04 s1300-23c03-04	48020	212555	A listener that uses a thread-filter has the potential to lose messages if it crashes while processing.
s1228-23c03-02	48580	212539	If the binding is done explicitly, providing a hostname, the server will eventually hang up, and in turn causes the clients to block.
s1408-23c03-04 s1194-23c03-04 s1300-23c03-04	48840	N/A	Bi-directional IOP causing requests to go to the wrong server.
s1408-23c03-04 s1194-23c03-04 s1300-23c03-04	49120	N/A	OrbixSSL C++ Clients incorrectly determines the SSL port number for IORs of a different-endian architecture to the client machine. This bug manifests itself in an OrbixSSL client attempting to invoke on the reverse endian interpretation of the port number. This bug does not happen when <code>bind()</code> using IOP is being used, also it does not happen if there is no <code>TAG_SSL_SEC_TRANS</code> component in the IOR.

## Orbix 2.3.4 Release Notes

Patch Number	Bug ID	PR Number	Synopsis
s1190-23c03-03 s1408-23c03-04 s1193-23c03-02 s1194-23c03-04 s1228-23c03-02	50000	214924	No bounds checking being done when passing IDL bounded strings.
s1228-23c03-02	50021	198987 220912	<p>Deadlock when loaders use application-level-locks.</p> <p>When Orbix invokes the registered loaders (for example as part of a look-up caused by an incoming object reference that needs to be loaded), it locks its Object Table to prevent a race condition. This locking of the Object Table, however, results sometimes in a lock-order deadlock with an application-level lock.</p> <p>For example:  Thread1 (in a loader) acquires application-level lock  [Object Table lock acquired, tries to acquire application-level lock].  Thread2 (not in a loader) acquires application-level lock and then calls a function that locks the Object Table (for example a function that creates a new proxy)  [application-level lock acquired, tries to acquire Object Table lock].  Deadlock when servant destructors use application-level locks.</p> <p>The very same problem arises in servant (Orbix object) destructors, since they are called while Orbix has the Object Table locked. This patch lets you redefine the lock (recursive mutex) used by Orbix to protect its Object Table. In this way you can collapse the two locks which deadlock (the Object Table lock and your application-level lock) into one. To do so, derive a class from <code>IT_ObjectTableExternRecursiveMutex</code> and override the <code>lock()</code> and <code>unlock()</code> functions.</p> <p>These functions should respectively call <code>lock</code> and <code>unlock</code> on an application lock. Then create an instance of the class and register it using <code>IT_ObjectTableExternRecursiveMutex::set_instance()</code>.</p> <p>For example:</p> <pre>#include &lt;it_object_table_exterm_recursive_mutex.h&gt; class MyRecursiveMutex : public IT_ObjectTableExternRecursiveMutex { public: virtual void lock() { global_app_recursive_mutex.lock(); } virtual void unlock() { global_app_recursive_mutex.unlock(); } }; int main() { MyRecursiveMutex my_rmutex; IT_ObjectTableExternRecursiveMutex::set_instanc e(&amp;my_rmutex); ... }</pre>

## Orbix 2.3.4 Release Notes

Patch Number	Bug ID	PR Number	Synopsis
s1190-23c03-02 s1190-23c03-03 s1408-23c03-04 s1193-23c03-02 s1194-23c03-04 s1300-23c03-04	50240	152303	Purify reports errors in the client and server when the client passes a struct that contains a nil object reference.
s1190-23c03-06 s1193-23c03-06	51180	218617	The <code>putidl</code> command fails for a large nested sequence of structures when in operation parameters are used.
s1190-23c03-01 s1408-23c03-04 s1193-23c03-01 s1194-23c03-04 s1300-23c03-04	51375	216649	The Orbix daemon does not show per-client-pid servers after restart when the checkpointing ( <code>-c checkpointfile</code> ) switch is used.
s1193-23c03-05	51488	217388	IDL compiler core dumps if the IDL file contains an undefined type.
s1190-23c03-06 s1193-23c03-06	51535	216864	Orbix reads configuration values from the <code>orbix.cfg</code> configuration file instead of the registry, even when the <code>IT_USE_CONFIG_FILE</code> is set to <code>no</code> .
s1194-23c03-02	51616	N/A	The <code>putit -j</code> flag did not exist, it is needed to launch Java servers from the C++ daemon.
s1193-23c03-03	51618	217527	Addresses problem introduced after patch 21 of Orbix2.3MT server that prevented Interop Context clauses and Anys from working.
s1193-23c03-05	51733	218332	An Objects reference count is being corrupted when a server has many threads accessing the Object.
s1193-23c03-05	51964	N/A	General memory leakage in Orbix demos.
s1192-23c03-08	52311	221099	User signals handlers are incorrectly restored internally by Orbix.
s1190-23-c03	52369	222679	A fix for a memory leak in the Orbix libraries when using OrbixEvents. There is also a change put in to stop the server CPU usage going up to near maximum of the machines total.

## 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>