

---

## Orbix 6.3.14 Release Notes

# Table of Contents

Introduction	4
New Features and Enhancements	5
Log4J 2 Upgrade	5
Removal of Orbix 6 GUI components	6
Java Version Support	6
Platform and Compiler Support	7
Upgrade to OpenSSL 3.x	7
CORBA Compliance	8
Platforms and Compilers	8
Deprecated Features	8
Unsupported Features	8
Migration from Previous Versions	9
Upgrading and TLS 1.3	9
Migrating away from Java 7	10
Additional steps required for Solaris deployments	10
OTS 2PC feature removed	11
Known Issues	12
Issue installing with JDK 17	12
Installer cannot load a valid Java virtual machine	12
MIOP issues on Linux platforms	13
Java TLS 1.3 half-close and synchronization issue	13
Java 11 and security	13
Using TLS with older JDKs	13
Microsoft Visual Studio 2015 compiler	14
Benign warning when launching the Windows installer	14
Oracle Solaris Studio 12.4 and 12.5 compiler	15
Deployment problem on Windows 7 or Windows 2008 R2 VM on VMWare	15
Aurea Actional integration feature removal	15
Spaces in install path and itant	15

Compiling 64-bit C++ applications	16
Resolved Issues	16
Issues resolved in previous HotFixes	16
Disclaimer	18

# Introduction

---

These release notes contain information about the Orbix 6.3.14 release from Rocket Software. They contain information that might not appear elsewhere in the documentation. Read them in their entirety before you install the product.

# New Features and Enhancements

---

Orbix 6.3.14 includes the following new features and enhancements:

## Log4J 2 Upgrade

---

Orbix 6.3.14 now uses Log4J 2.18.0 for logging, with a modified version of the Log4J 1.2 API bridge. The Log4J 1.2 API bridge allows code to still use the Log4J 1.2 API, and the following modifications were made:

- Log4J1 appenders are now supported.
- Log4J1 compatibility mode is enabled by default; if the `log4j1.compatibility` system property is not set, it is assumed to be true. This means that Log4J1-style XML and properties files are automatically handled.
- The `BasicConfigurator.configure()` method works as it did in Log4J1 (adding a console appender to the root logger).

The following changes in behaviour are of note:

- In Log4J1, the default logging level is `DEBUG`; in Log4J 2, it is `ERROR`.
- In Log4J2, the root logger now has a console appender by default. In Log4J1, the root logger had no appenders by default.
- The 1.2 API bridge does not handle parent loggers completely. In Log4J1 and Log4J2, a logger named `A.B` would have logger `B` and the root logger as parents. In the 1.2 API bridge, all non-root loggers only have the root logger as a parent. The main effect of this is that appenders on parent loggers other than the root logger will not be invoked when a logger is called; only appenders on the logger itself and the root logger will be invoked.
- The 1.2 API bridge does not support additivity. Calling the `Category` class's `setAdditivity()` method does not function as expected and should be strictly avoided.
- Appenders created within the Log4J2 core are not visible via the 1.2 API bridge (they will not be returned by the `Category` class's `getAppenders()` method). The main examples of Log4J2 appenders are the default console appender that is placed on the root logger, and appenders created via the Log4J1-style configuration file (such as `log4j.properties`).

## Removal of Orbix 6 GUI components

---

The following GUI components are no longer automatically installed:

- Orbix Management Service Console (Administrator Web Console)
- Orbix Configuration Authority Web Console
- Notification Service Console
- Telecom Log Service Console
- Trader Service Console
- Orbix Configuration Explorer

They are now available as an optional component. To install, please download and extract the GUI components archive and follow the installation instructions contained therein.

The GUI components have been separated from the product as they do not fully comply with the Section 508 and WCAG accessibility requirements and guidelines. It is intended that future product releases will include updated GUI components that comply with the relevant accessibility guidelines.

## Java Version Support

---

Orbix 6.3.14 has been rebuilt using Java 8 and no longer supports Java 7.

Support for Java 11 has been added to AIX, HP-UX 11iV3, and Windows.

Support for Java 21 has been added for AIX, Linux, and Windows.

Support for Open JDK 11 has been added to Solaris 11 Intel.

Support for Open JDK 17 has been added to the Linux and Windows platforms.

## Platform and Compiler Support

---

Orbix 6.3.14 has added support for the following platforms:

- RedHat Enterprise Linux 9.x
- Windows 11
- Windows Server 2022

Orbix 6.3.14 has added support for the following compilers:

- gcc 11.2 on RedHat Enterprise Linux 9.x
- Visual Studio 2022 on Windows 11 and Windows Server 2022

## Upgrade to OpenSSL 3.x

---

For the latest release of Orbix 6, the underlying OpenSSL build has been updated to version 3.0.14.

This means, by default, there will be a higher level of security due to OpenSSL disallowing many legacy ciphersuites and algorithms.

This may result in existing deployments needing to have any or all of the following changed:

- TLS Protocol (by default TLS 1.0 and TLS 1.1 are now disabled).
- TLS Ciphersuite (by default DES and RC4 ciphers now disabled).
- TLS Certificates (certificates signed by SHA-1 are no longer allowed by default).

If you are affected by any of these default changes to OpenSSL, then the correct course of action would be to upgrade certificates and/or alter protocol and ciphersuite values in your configuration. For details on making these changes, see the *Orbix 6.3.14 Security Guide*.

However, if doing this is unfeasible then an alternate solution may be to lower the default security level back to zero, which will allow some legacy settings.

This can be achieved by adding the following line to relevant areas of your domain configuration file:

```
plugins:openssl:openssl_security_level = "0";
```

All demonstration certificates and CA lists which are shipped with Orbix 6 (and found under `$PRODUCT_DIR/asp/6.3/etc/tls/x509`) have been updated accordingly for this release.

Further details regarding OpenSSL upgrades to 3.0 can be found on the [OpenSSL website](#).

# CORBA Compliance

---

Orbix 6.3 complies with the following specifications:

- CORBA 2.6
- GIOP 1.2 (default), 1.1, and 1.0
- C++ Language Mapping (formal/99-07-41)
- IDL-to-Java Language Mapping (formal/99-07-53)
- Object transaction service (OTS) 1.1 and 1.2

## Platforms and Compilers

---

For a full list of platforms and compilers supported by Orbix 6.3.14, see the *Orbix 6.3.14 Platform Support Notes*.

## Deprecated Features

---

The following features are deprecated. Deprecated features are supported for only backwards compatibility, and may be removed in a future release:

- Firewall Proxy Service

The Firewall Proxy Service is deprecated as of Orbix 6.3.12.

## Unsupported Features

---

The following features were no longer supported as of Orbix 6.3.11:

- Legacy cipher suites

Advances in cryptanalysis mean that many older encryption methods can no longer be considered secure, and cipher suites using such methods cannot be recommended and may even not be supported by recent JDK versions. Therefore, Orbix 6.3 no longer supports the following:

- Any null-encryption cipher suite



- Any export-strength cipher suites

See the *Orbix 6.3.14 Security Guide* for more information on cipher suites.

## Migration from Previous Versions

---

To upgrade to Orbix 6.3.14 from existing Orbix 6.3.x installations:

- Back up existing installations before you upgrade.
- Go to the Orbix 6.3.x directory and run the Orbix 6.3.14 installer. The Orbix installer overwrites the existing version.
- If you experience any difficulty with running existing applications, see the following sections:
  - [Upgrading and TLS 1.3](#)
  - [MIOP issues on Linux platforms](#)
  - [Java TLS 1.3 half-close and synchronization issue](#)

For details on installing Orbix 6.3.x service packs, see the *Orbix Installation Guide*. For details on migrating from earlier Orbix versions, see the migration and upgrade documentation on the Rocket Software documentation site.

## Upgrading and TLS 1.3

---

Updating Orbix 6.3 to the new TLS 1.3 support settings by default may cause an issue when installing on top of an existing deployment, if no TLS 1.3 ciphers are enabled. As these TLS 1.3 ciphers are new and completely different from TLS 1.2 ciphers they did not exist previously and therefore were not present in Orbix 6.3.11 (or earlier versions) by default.

If a mismatch occurs, you may see a message similar to the following when re-starting your servers or services:

```
IDL:omg.org/CORBA/NO_PERMISSION:1.0: minor = 0x49540A41
```

```
(IT_IIOP_TLS:SPECIFIED_POLICIES_INCOMPATIBLE), completion status = NO
```

This means that you need to change either of the following configuration variables:

If you wish to continue using your existing cipher suites and use TLS 1.2, then change your protocol variable setting as follows:

```
policies:mechanism_policy:protocol_version = ["TLS_V1_2"];
```

If you wish to upgrade your cipher suites to use TLS 1.3, update your cipher suites variable to the new default value:

```
policies:mechanism_policy:ciphersuites = ["TLS_AES_256_GCM_SHA384", "RSA_WITH_AES_256_CBC_SHA256",  
"RSA_WITH_AES_256_CBC_SHA", "RSA_WITH_AES_128_CBC_SHA", "RSA_WITH_AES_128_CBC_SHA256"];
```

For more information on these variables and their possible settings, see the *Orbix 6.3.14 Security Guide*.

## Migrating away from Java 7

---

Orbix 6.3.14 no longer supports Java 7.

If the existing Orbix deployment that you want to upgrade to Orbix 6.3.14 was deployed using Java 7, you need to take the steps outlined below to migrate away from Java 7 to a supported Java release.

During deployment, the Orbix configuration tool `itconfigure` generates scripts that are used to set the Orbix development environment and to start/stop Orbix services. These scripts contain variables that point to the location where the JDK shared libraries (on UNIX) or DLLs (on Windows) reside.

For each deployed domain `<domain_name>`, the following script must be updated, replacing the existing Java 7 paths with their Java 8 or later equivalents:

```
$IT_PRODUCT_DIR/etc/bin/_env
```

Update the shared library path variable to a Java 8 or later location

## Additional steps required for Solaris deployments

---

In addition to the steps outlined in the section above, the following steps are required to migrate to Java 8 or later. These additional steps are required due to the fact that Solaris JDKs are only provided as 64-bit JDKs (32-bit JDKs are not available for Java 8 or later).

For each deployed domain `<domain_name>`, the following scripts need to be updated as described below:

```
$IT_PRODUCT_DIR/etc/bin/<domain_name>_env
```

The `PATH` variable must be updated to point to the 64-bit binaries first: i.e. modify `$IT_PRODUCT_DIR/asp/6.3/bin` in the `PATH` variable to be:

```
$IT_PRODUCT_DIR/asp/6.3/bin/amd64
```

(for Solaris x86\_64)

```
$IT_PRODUCT_DIR/asp/6.3/bin/sparcv9
```

(for Solaris SPARC)

```
$IT_PRODUCT_DIR/etc/bin/start_<domain_name>_services
```

Lines containing `$IT_PRODUCT_DIR/asp/6.3/bin/<it_service_name>` (where `<it_service_name>` is equal to `itlocator`, `itnaming`, etc.) must be updated to contain the following string instead:

```
$IT_PRODUCT_DIR/asp/6.3/bin/amd64/<it_service_name>
```

(for Solaris x86\_64)

```
`$IT_PRODUCT_DIR/asp/6.3/bin/sparcv9/`
```

(for Solaris SPARC)

```
$IT_PRODUCT_DIR/etc/bin/stop_ _services
```

Lines containing `$IT_PRODUCT_DIR/asp/6.3/bin/itadmin` must be updated to contain the following string instead:

```
$IT_PRODUCT_DIR/asp/6.3/bin/amd64/itadmin
```

(for Solaris x86\_64)

```
$IT_PRODUCT_DIR/asp/6.3/bin/sparcv9/itadmin
```

(for Solaris SPARC)

## OTS 2PC feature removed

---

The Object Transaction Service (OTS) feature supporting fully recoverable 2PC (2-phase commit) transactions, using one or more resource managers, has been removed from this release (Orbix 6.3.14). Both the OTS 2PC standalone service and the OTS 2PC application plug-in have been removed from the product.

The OTS Lite feature is still available. OTS Lite provides a lightweight transaction coordinator supporting the 1PC (1-phase commit) protocol. It is available as an application plug-in and requires minimal configuration and administration. It can be used by applications with a single resource manager.

## Known Issues

---

Orbix 6.3.14 may be affected by the following known issues:

### Issue installing with JDK 17

---

When installing with JDK 17 using the graphical interface, you may encounter an issue where the installation stalls at the **Choose Install Set to Install** panel. In this case, you can try one of the following:

- Perform a silent installation using JDK 17. See the "*Installing Silently*" section in the **Installation Guide** for instructions.
- Install with a different supported JDK, such as JDK 11. At the **Choose Java Virtual Machine** panel, navigate to the JDK 17 Java executable. After installation, ensure the `JAVA_HOME` and `PATH` environment variables are set appropriately for JDK 17.
- Perform a graphical installation using JDK 17 and set the `_JAVA_OPTIONS` environment variable to `"--add-opens=java.desktop/sun.awt.shell=ALL-UNNAMED"` before launching the installer. On Windows, if the graphical installer is launched by double-clicking the installer in File Explorer, or by running it at a non-administrator command prompt, be sure to set `_JAVA_OPTIONS` (along with the necessary `JAVA_HOME` and `PATH` variables) using the **Environment Variables** window first. If using an administrator command prompt, the environment variables can be set in the command prompt itself. After installation, the `_JAVA_OPTIONS` environment variable can be unset/removed as it is not required when using JDK 17 with Orbix 6.

### Installer cannot load a valid Java virtual machine

---

When running the installer on Windows, you may encounter an error indicating that no valid Java virtual machine could be loaded. In this case, check that Java is properly installed and that the Java bin directory is on the system path. (Having the Java `bin` directory on a current user path is not sufficient, as the installer runs in elevated mode and might not inherit the user environment.)

Alternatively, you can use the installer's `LAX_VM` parameter to select a Java virtual machine which is present on the system. After the `LAX_VM` parameter, you need to specify the absolute path of the Java executable. For example:

```
microfocus_orbix_6.3.14_win_64_vs2015.exe LAX_VM "%JAVA_HOME%/bin/java.exe"
```

## MIOP issues on Linux platforms

---

On newer Linux platforms, Orbix applications using the Multicast Inter-ORB Protocol (also called MIOP or EGMIOP) may not work correctly when initially installed because of the default restrictive network rules of the system firewall service.

We recommend running the following command to configure the firewall service to permit MIOP messages:

```
firewall-cmd --permanent --zone=public --add-protocol=udp
```

## Java TLS 1.3 half-close and synchronization issue

---

Orbix 6.3.14 may be affected by a known JDK issue in which applications do not work well with the TLS 1.3 half-close policy. If, after migrating from TLS 1.2 to TLS 1.3, some of your applications hang awaiting further operations, then you are probably seeing this issue.

See <https://bugs.openjdk.java.net/browse/JDK-8208526> for more details.

## Java 11 and security

---

Java 11 also includes a number of security updates, and for older applications the current deployed security certificates may not work out of the box (for example, the key size, or signature algorithm strength may not satisfy the default security configuration of the JDK). Where feasible, we recommend upgrading any security certificates and cipher suites to work with the default security configuration in Java 11.

## Using TLS with older JDKs

---

Older JDKs (such as versions of JDK8 before update 162) do not install the Java Cryptography Extension (JCE) Unlimited Strength Jurisdiction Policy Files by default.

These files are needed to support the SHA-256 ciphers which are recommended for greater security and as such are now enabled in Orbix by default, so we recommend that you download and install these unlimited strength `.jar` files.

If for some reason you do not wish to install these files on top of older JDKs, you will need to make the following change:

1. Remove the 256 ciphers from their cipher suite lists. This would involve updating the `tls.xml` file in `$OrbixInstallDir/asp/6.3/etc/conf/tls.xml` for future deployments.

2. If you wish to deploy the secure configuration repository (CFR) and the security service then, in addition to the above change, you must also add a cipher list that omits the stronger ciphers. Add this under the `is2_inFile` section in the same `tls.xml`.

For example:

```
policies:mechanism_policy:ciphersuites list RSA_WITH_RC4_128_SHA RSA_WITH_RC4_128_MD5
RSA_WITH_DES_CBC_SHA RSA_WITH_3DES_EDE_CBC_SHA RSA_WITH_AES_128_CBC_SHA
```

## Microsoft Visual Studio 2015 compiler

---

Orbix 6.3.14 supports Microsoft Visual Studio 2015 version 14.0.25431.01 Update 3 or later versions. Support is not provided for earlier versions than Microsoft Visual Studio 2015 version 14.0.25431.01 Update 3 because of compiler issues discovered during testing.

We advise customers to install Visual Studio 2015 using the Visual Studio 2015 Web installer (not the ISO installer) to avail themselves of the latest fixes available from Microsoft.

### Older security algorithms with recent JDKs

Recent JDKs may by default disable the use of what are considered legacy algorithms. This includes older protocols, ciphers, digests and also may include insisting that key sizes used are above a certain size. This is due to progress in cryptanalysis which has rendered some of these older algorithms no longer strong enough.

We highly recommend that any certificates used in secure Orbix applications that are signed using older functions, such as with an MD5 digest signature, are regenerated to use at least a SHA-2 digest signature.

## Benign warning when launching the Windows installer

---

When installing Orbix 6.3.14 on Windows, the installer may issue a warning about a missing `java.dll`, or a registry key specifying the wrong version of Java. This is a benign warning and can be safely ignored.

This warning is issued because the installer does a thorough search across the system for a usable version of Java with which to launch the installer. When an incomplete installation is found, the warning may be issued.

Such an incomplete Java installation is typically a leftover install of Java JRE installed via the "Java Update" mechanism. In order to ensure that the installation is wiped properly, run the "Java Update" installer, and at the end of the installation, agree to uninstall previous older JRE installations.

## Oracle Solaris Studio 12.4 and 12.5 compiler

---

Oracle Solaris Studio 12.4 and 12.5 compiler is not certified with Orbix 6.3.14. A compiler issue was uncovered while certifying Orbix 6.3.8 with Studio 12.4. The compiler issue relates to an inconsistent behavior in mangling symbol names between Studio 12.4 and earlier compiler versions.

The compiler vendor fixed this defect in Studio 12.6 and Orbix 6.3.14 is certified to work with Studio 12.6.

We advise customers to refrain from using Oracle Solaris Studio 12.4 and 12.5 with Orbix 6.3.14.

## Deployment problem on Windows 7 or Windows 2008 R2 VM on VMWare

---

You might encounter an intermittent failure to deploy services on virtual machines with only 1 CPU. This problem does not occur when the virtual machine has 2 or more CPUs.

## Aurea Actional integration feature removal

---

The Orbix integration feature with Aurea Actional Application Performance Monitoring has been removed from Orbix 6.3.14.

## Spaces in install path and itant

---

If your Orbix installation path contains spaces, and you use the `itant` tool to build the Java demos, the following message might appear in the console output:

```
C:\Program%20Files\Micro%20Focus\Orbix\asp\6.3\demos\corba\demo.xml could not be found
```

This is a benign message and can be ignored. The Java demos build successfully.

## Secure CFR domain with replicated services

In a secure configuration repository (CFR)-based domain with replicated Orbix services, CFR replica sets can not be automatically shrunk. This issue does not occur in an insecure CFR-based domain. If you have to remove CFR replicas in a secure CFR-based domain, please contact Orbix technical support.

## Compiling 64-bit C++ applications

When compiling 64-bit applications with the C++ Sun Studio 12 Update 2 compiler on a Solaris x86 platform, there may be issues relating to compiling certain demos delivered with Orbix. The issue relates to a known compiler bug in the C++ compiler. Oracle is aware of this issue, and as a workaround suggests compiling the code with the `-O1` flag instead of using the debug `-g` flag.

## Resolved Issues

The resolved issues that customers have reported are listed in this section. The numbers that follow each issue are the Reported Problem Incident number or an Octane defect number followed by the Customer Incident Numbers (where applicable) in parentheses. RPIs/Octane defects that have numbers only (and no text) are included to confirm that the RPIs/Octane defects have been fixed, since no further information is required.

### Issues resolved in previous HotFixes

This section includes issues that were fixed in HotFixes to Orbix 6.3 SP12, and are now incorporated into SP14.

- The Orbix 6 IDL compiler has been enhanced such that it can ignore annotations that may appear in an IDL file. This change ensures that the IDL compiler can continue to compile IDL files as if they did not contain any IDL-RS annotations.

641246

- OpenSSL server cipher preference is now configurable. Consider the following hypothetical lists:

Server side ciphers: AES-256, AES-128

Client side ciphers AES-128, AES-256

The Orbix 6 default is client cipher preference. During hand- shake negotiation, the chosen cipher would be AES-128.

If Orbix 6 is configured for server cipher preference, the cipher chosen during the handshake would be AES-256.

Configure server cipher preference as follows:

```
plugins:openssl:cipher_server_preference = "true";
```

138109

- OpenSSL "middle box" capability is now configurable. TLSv1.3 handshakes may encounter problems when flowing through network hardware that does not understand TLSv1.3. The Orbix 6



default is to run with middle box capability turned on, providing a handshake flow that looks more like TLSv1.2. To turn off middle box capability, set the following configuration item:

```
plugins:openssl:enable_middlebox_compatibility = "false";
```

140105

# Disclaimer

---

© 2024 Rocket Software, Inc. or its affiliates. All Rights Reserved.

The only warranties for products and services of Rocket Software, Inc and its affiliates and licensors ("Rocket Software") are as may be set forth in the express warranty statements accompanying such products and services. Nothing herein should be construed as constituting an additional warranty. Rocket Software shall not be liable for technical or editorial errors or omissions contained herein. The information contained herein is subject to change without notice.