opentext | CORBA An OMG Standard

Discover the Future of CORBA

OpenFusion JacORB 3.9.0.0

Release Notes

2023-09-07

Contents

Micro Focus OpenFusion JacORB 3.9.0.0 Release Notes

Micro Focus's OpenFusion JacORB release is a full binary distribution that is easily installed and configured, with no additional compilation required following installation. The full source code tree for OpenFusion JacORB is also included. This gives users the ability to extend the ORB, perhaps to add support for non-standard functionality or add new features to more precisely match their requirements.

Installation

In previous releases of OpenFusion, OpenFusion CORBA Services and OpenFusion JacORB were included in the same installer. As of this release, OpenFusion JacORB and OpenFusion CORBA Services have separate installers. Both products are installed using a Java-based installer program.

If you intend to use OpenFusion CORBA Services with OpenFusion JacORB, you should install OpenFusion JacORB first, and then install CORBA Services in the same directory. See the *Micro Focus OpenFusion CORBA Services Product Guide* for full installation instructions for CORBA Services.

You can install OpenFusion JacORB using an interactive graphical user interface (GUI Mode) or using commands entered on the command line (Command Line Mode). Using GUI Mode is generally the more popular method, however the Command Line Mode is useful when automating the installation with a script.

The JacORB installation file is called *`Setup_JacORB_39_JDKn.jar`*, where *n* is one or two digits indicating the Java version: *`Setup_jacORB_39_JDK8.jar`* for JDK 8, *`Setup_JacORB_39_JDK11.jar`* for JDK 11, and so on.

Installing JacORB Using GUI Mode

1. Follow the instructions on the Micro Focus Web site to select the appropriate *`Setup_JacORB_39_JDKn.jar`* file for installing OpenFusion JacORB.
2. Run the *`Setup_JacORB_39_JDK11.jar`* file (***without*** any options) from the command line, as follows:

```
%  java -jar Setup_JacORB_39_JDK11.jar
```

   This will display the installer's graphical user interface.
3. Follow the instructions displayed in the GUI, selecting the services and components you want to install.

Installing JacORB Using Command Line Mode

1. Follow the instructions in *Step 1* under Installing JacORB Using GUI Mode above.
2 Run the *Setup_JacORB_39_JDKn.jar* file ***with*** the options shown below:

```
%  java -jar Setup_JacORB39_JDK11.jar <-list | [<install_dir>
[components]]>
```

where

- *-list* will list all available services and components, without performing the installation
- *<install_dir>* is the directory where OpenFusion JacORB is to be installed
- *[components]* is the list of components and services to be installed; if no components are specified, then all components will be installed

Example 1: List all available services and components

```
%  java -jar Setup_JacORB39_JDK11.jar -list
```

Example 2: Install all services and components to /MicroFocus/myOF

```
%  java -jar Setup_JacORB39_JDK11.jar /MicroFocus/myOF
```

Example 3: Install the JacORB component to /MicroFocus/myOF

```
%  java -jar Setup_JacORB39_JDK11.jar /MicroFocus/myOF JacORB3
```

Operating Systems Supported

For a full list of supported platforms, see
http://supportline.microfocus.com/prodavail.aspx

Features Specific to OpenFusion JacORB

Micro Focus has added a number of valuable features to OpenFusion JacORB that are not available in the standard Open Source JacORB distribution. These features include those described in the following sections.

OpenFusion IMR

The OpenFusion Implementation Repository (IMR) is used by OpenFusion JacORB to locate and activate CORBA object implementations. The OpenFusion IMR was designed so that it can also be used to locate object implementations based on OpenFusion TAO. It also provides a single unifying IMR that can be used across OpenFusion ORBs. However, in this release of the OpenFusion IMR, support is only provided for CORBA

servers written with OpenFusion JacORB. The OpenFusion IMR provides a number of advanced features including:

- Load balancing between multiple server object instances. This feature can also be used to provide basic fail over between servers objects.
- Support is provided for fail over between IMR instances. Multiple OpenFusion IMR instances can be started, if one instance fails then any OpenFusion JacORB server object references registered with the first instance will be automatically re-registered with the second OpenFusion IMR instance and client requests will continue to be processed as normal.
- Support is provided for auto-activation of server objects. Any persistent object references registered with the OpenFusion IMR can be automatically started if they are not already running when the IMR receives a client request.
- Support for fail over between IMR instances across different network subnets.

Transparent corbaloc URLs

The `jacorb.properties` file now includes references to allow transparent corbaloc URLs for the OpenFusion CORBA Services.

ThreadPools for connections

The `jacorb.properties` file now includes new properties to limit the thread used for client and server side connections. See the *JacORB Programming Guide* for more information on `jacorb.connection.client.max_receptor_threads`, `jacorb.connection.server.max_idle_receptor_threads` and `jacorb.connection.server.max_receptor_threads`.

Shared threadPool for request processing

Initially JacORB provided a thread pool per POA. Using the new property `jacorb.poa.thread_pool_shared` allows one thread pool per JVM instance.

IDL compiler improvement

An option has been added to the IDL compiler that will generate `toString()` and `equals()` methods on generated stubs for Structs.

Slow ORB initialisation and high thread usage when using JSSE

Under certain platforms (such as J2ME CDC platforms) when the JSSE initializes its random number generator it may spawn a large number of threads and/or ORB initialisation is slow.

OpenFusion JacORB comes with a plugin system to resolve this issue. An implementation of the org.jacorb.security.ssl.sun_jsse.JSRandom interface can initialise SecureRandom as required. Two examples are provided in the org.jacorb.security.ssl.sun_jsse package in the JacORB source code.

Server/client keep alive

The `jacorb.properties` file now includes new properties to allow servers and clients to be kept alive. See the *JacORB Programming Guide* for more information on `jacorb.connection.server.keepalive` and `jacorb.connection.client.keepalive`.

Monitoring TCP and SSL connections

- The `jacorb.properties` file now includes new properties to allow a developer to implement an interface, using the Java EventListener pattern, which will create

listeners to monitor TCP and SSL connections and receive notifications. See the *JacORB Programming Guide* for more information on JacORB Network Event Logging.

- TCPConnectionEvent and SSLSessionEvent are extended to include a method to return the local IP.

SecureRandom initialisation

The `jacorb.security.randomClassPlugin` allows developers to plug-in their own `java.security.SecureRandom` initialisation when using SSL. See the *JacORB Programming Guide* for more information on this plug-in.

Additional information for threads

A new property `jacorb.enhanced_thread_name` has been added to configure additional information for threads. Specifically, it adds connection endpoints and time (in milliseconds) that the thread started to the Thread name. See the *JacORB Programming Guide* for more information.

API change for SSLSessionEvent

A change has been made to the API for SSLSessionEvent to receive and provide the cause of exceptions. SSL logging now also provides the exception cause.

JacORB timeout properties to set in Notification Service scenarios:

- A consumer attempts to connect to the Notification Service behind a badly configured firewall. Set `jacorb.connection.client.connect_timeout` to timeout the attempted connection.
- A push_structured_event call to a consumer takes a long time. Set `jacorb.connection.client.pending_reply_timeout` to allow the reply to timeout.
- A machine is powered down before a Notification Service client performs a disconnect call on a socket and the socket remains open. Set `jacorb.connection.client.idle_timeout` so that the connection will be closed.

JacORB IORMutator

An enhancement has been added to allow the developer to alter incoming and outgoing objects at a low level within the ORB. This is useful for scenarios where a user is running with legacy network elements which have multiple, identical IP address, for example.

Note that the IORMutator should be used with caution since it operates at the CDRStream level, which makes it easy to break the ORB and cause unpredictable behaviour.

New Features

The changes made to JacORB at version 3.9.0.0 are listed below. Note that bug fixes are listed in the Resolved Issues section.

- Updated to be in line with open-source JacORB 3.9
- SLF4J has been updated from 1.7.6 to 1.7.14.
- There are now separate builds of OpenFusion JacORB 3.9 for JDK 8, 11 and 17. OpenFusion JacORB 3.9 can only be installed and used with the appropriate JDK. For example, to use OpenFusion JacORB 3.9 with JDK 11, the JDK 11 installer file needs to be installed using JDK 11.
- CORBA RMI-IIOP is only available in the JDK 8 build (the required support for CORBA RMI-IIOP in the JDK was removed after JDK 8).

Known Issues

Issue	Solution / Workaround
RMI issue with with JDK 1.5.0	When you build JacORB with JDK 1.4 and run JacORB with JDK 1.5, some unknown and indirection offset errors occur in the RMI area with the following tests: orb.rmi.RMITest.test_exception RMITest.test_vectorToValueArray RMITest.test_getException RMITest.test_referenceSharingWithinCollection

Encoding of indirections	An enhancement has been added to the JacORB code to allow the encoding of indirections to be enabled and disabled via a flag in the config file. The default is for indirection encoding to be enabled.
	An entry has been added to the JacORB properties file:
	# Turn off indirection encoding for repeated typecodes. This fixes
	# interoperability with certain broken ORB's eg. Orbix E2A
	jacorb.interop.indirection_encoding_disable=off

JSSE idle_timeout with JDK1.3	Due to bugs with Sun's JSSE implementation when used in conjunction JDK 1.3, you may experience COMM_FAILURE exceptions when there are pending messages and the JacORB idle_timeout property is set.
Shutting down the OpenText ImR	org.jacorb.imrutility.imr.RepositoryImpl.shutdown method has been amended to ignore the boolean paramter passed to this method. It will always shutdown the ImR immediately and future implementations will have the method API changed to remove the boolean parameter.
Using the compactTypecodes property	Configuration of JacORB using the compactTypecodes property can save bandwidth when marshalling type code information over the wire. If compactTypecodes is set to off, then no bandwidth saving action is undertaken. If compactTypecodes is set to on, then names, member types and member names are removed. While this improves performance and saves bandwidth it may cause interoperability issues with another ORB that does not handle compacted typecodes.

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 followed by the Customer Incident Numbers (in parentheses). RPIs that have numbers only (and no text) are included to confirm that the RPIs have been fixed, since no further information is required.

3.9 ORB

- Embed fatal memory issues within the comm failure for diagnostic purposes.
- Moved setting of configure flag before queue listeners are added to remove exception condition. (Matthew Ruggiero).
- Fix ArrayIndexOutOfBoundsException in TaggedComponentList.removeComponents (Richard Hash).
- Added "virbr0" to the default "jacorb.network.virtual" system property. (Richard Hash).
- NameService. Fixed a bug that happened randomly because of modifying a map while enumerating it (sasuu).
- NotificationService. Fixed a bug where adding a new message filter or filter matching in NotificationService caused.
- ConcurrentModificationException while iterating filters. (Pulla Johannes).
- Misc
  Documentation improvements by Leif Gruenwoldt.
  Added example of using an enum with an any (Phil Mesnier).

3.8 ORB

- Fix for null ptr in IIOPAddress (BZ1025).
- Improve network ordering (BZ1023).
- Add VMware for network ordering (BZ1024).
- Fixes for byte order selecting.
- Add ASCII and MacRoman as aliases for the ISO-8859-1 codeset.
- Fix for bidirectional IIOP where the client creates a callback IOR with alternate addresses (BZ1026).
- Add jacodbg script to enable attaching a remote debugger to a shell-started process.
- Fix to prevent COMM_FAILURE in concurrent handling of ForwardRequest in Portable Interceptors (BZ1014).

User Documentation

New documentation for this release is available online, at https://www.microfocus.com/en-us/support/documentation.

Updates and SupportLine

Our Web site gives up-to-date details of contact numbers and addresses.

Further Information and Product Support

Additional technical information or advice is available from several sources. The product support pages contain a considerable amount of additional information, such as:

- The WebSync service, where you can download fixes and documentation updates.
- The Knowledge Base, a large collection of product tips and workarounds.
- Examples and Utilities, including demos and additional product documentation.

To connect, enter *http://www.microfocus.com* in your browser to go to the Micro Focus home page.

Note: Some information may be available only to customers who have maintenance agreements.

If you obtained this product directly from Micro Focus, contact us as described on the Micro Focus Web site, *www.microfocus.com*. If you obtained the product from another source, such as an authorized distributor, contact them for help first. If they are unable to help, contact us.

Disclaimer

This software is provided "as is" without warranty of any kind. Micro Focus disclaims all warranties, either express or implied, including the warranties of merchantability and fitness for a particular purpose. In no event shall Micro Focus or its suppliers be liable for any damages whatsoever including direct, indirect, incidental, consequential, loss of business profits or special damages, even if Micro Focus or its suppliers have been advised of the possibility of such damages. Some states do not allow the exclusion or limitation of liability for consequential or incidental damages so the foregoing limitation may not apply.