Discover the Future of CORBA

OpenFusion JacORB 3.7.1.0

Readme

Micro Focus
The Lawn
22-30 Old Bath Road
Newbury, Berkshire RG14 1QN
UK

https://www.microfocus.com

Copyright © Micro Focus 2009-2021. All rights reserved.

MICRO FOCUS, the Micro Focus logo, and Micro Focus product names are trademarks or registered trademarks of Micro Focus Development Limited or its subsidiaries or affiliated companies in the United States, United Kingdom, and other countries. All other marks are the property of their respective owners.

2021-04-14

Contents

Micro Focus OpenFusion JacORB 3.7.1.0 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_JacORBnn.jar`*, where *nn* is two digits indicating the Java version: *`Setup_JacORB16.jar`* for JDK 1.6, *`Setup_JacORB17.jar`* for JDK 1.7, and so on.

Installing JacORB Using GUI Mode

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

```
%  java -jar Setup_JacORB16.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_JacORBnn.jar` file ***with*** the options shown below:

```
%  java -jar Setup_JacORB16.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_JacORB16.jar -list
```

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

```
%  java -jar Setup_JacORB16.jar /MicroFocus/myOF
```

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

```
%  java -jar Setup_JacORB16.jar /MicroFocus/myOF JacORB3
```

Operating Systems and Processors Supported

This section details the operating systems and processors that are supported by this release.

Operating System	Operating System Version	Processor
Red Hat Enterprise Linux	6.x	Intel x86-64
Red Hat Enterprise Linux	7.x	Intel x86-64
Solaris	10.x	Intel x86-64
Solaris	10.x	SPARC
Solaris	11.x	Intel x86-64
Solaris	11.x	SPARC
SUSE Linux Enterprise Server	12.x	Intel x86-64
Windows	7	Intel x86-x64
Windows	8.1	Intel x86-x64
Windows	10	Intel x86-x64
Windows Server 2008	R2	Intel x86-x64
Windows Server 2012	R2	Intel x86-x64

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 3.7.1.0 are listed below. Note that bug fixes are listed in the Resolved Issues section.

- JacORB is rebuilt with OpenJDK.

User Documentation

New documentation released with this Service Pack is available online, from https://supportline.microfocus.com/productdoc.aspx.

Known Issues

For known issues with OpenFusion JacORB, see the *Known Issues* section of the release documentation that is installed as part of the product, available at `/docs/JacORB/html/knownIssues.html`.

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.

Issues Resolved in this Release

- Socket was not closing when `deliverConnection()` fails.

  1099402 (2820586)

- Extra logging has been added to show when a socket has been closed.

  1099756 (2820586)

- A race condition in the request analyser processor clean-up code could result in out-of-order request processing when the DefaultRequestAnalyserProcessor was configured by setting the `jacorb.poa.requestanalyserprocessor.enable_removal` property. Requests from a client connection should have been processed within a server strictly one after another by a single thread, but in some circumstances they were being processed by more than one thread and not strictly in FIFO order. This is now fixed. Refer to the user documentation of `jacorb.poa.requestanalyser` and `org.jacorb.poa.DefaultRequestProcessorAnalyser` for background details of this ORB behavior.

  1097325 (2802062)

- JacORB 3.7.0 supports Java 7.x and 8.x on Red Hat Enterprise Linux.

  1096095 (2792787)

- Added a `maxMessageBufferSize` parameter to limit allocated buffer sizes to address a vulnerability issue with GIOP headers over telnet.

  JAC-80

- Fix for server IIOP connection's `close()` method implementation.

  JAC-75

- Fix for invocation context loss with nested colocated calls.

  JAC-64

- Added write lock timeouts to GIOP connections for resolving potential thread count build up issue.

  JAC-63

- Fix for portable interceptor ordering issue with colocated calls.

  JAC-8

- Fixed profile selection in bind/rebind for group IOR (IOGR).

  JAC-62

- Fixed code generation for the default case of boolean-switched unions.

  JAC-60

- Fixed high CPU consuming cause in RequestAnalyser.

  JAC-51

- Fix for loop in GIOPConnection when underlying transport closing fails with exception.

  JAC-45

- Added `jacorb.connection.client.ungraceful_shutdown` property. If set to `true`, the client connection manager will invoke the shutdown method on the connection before closing. The shutdown implementation will break the connecting thread's retries cycle (after the current retry finishes) and will release synchronization locks to allow other threads to close.

  JAC-44

- Fix for naming context implementation from static to non-static 'orb' field.

  JAC-41

- Fixed waiting for request threads completion in `ORB.shutdown ()`.

  JAC-38

- Fix for GIOP 1.0 / 1.1 message header writing.

  JAC826

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 Software Licenses and Downloads (SLD) website at *https://sld.microfocus.com/mysoftware/index* is where you can download products and patches for them. A user ID and password are required to access the Software Licenses and Downloads website.

  To obtain a user ID and password, sign up online or contact Customer Support by email or telephone.
- The Knowledge Base, a large collection of product tips and workarounds.
- Examples and Utilities, including demos and additional product documentation.

To connect, enter *https://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.

Micro Focus is a registered trademark.
Copyright © Micro Focus 2009-2021. All rights reserved.