



OpenFusion JacORB 3.7.0.0

Release Notes

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

<http://www.microfocus.com>

Copyright © Micro Focus 2009-2016. 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.

2016-05-26

Contents

Micro Focus OpenFusion JacORB 3.7.0.0 Release Notes	5
Installation	5
Operating Systems Supported	5
Features Specific to OpenFusion JacORB.....	5
OpenFusion IMR	5
Transparent corbaloc URLs	6
ThreadPools for connections	6
Shared threadPool for request processing.....	6
IDL compiler improvement	6
Slow ORB initialisation and high thread usage when using JSSE.....	6
Server/client keep alive	6
Monitoring TCP and SSL connections.....	6
SecureRandom initialisation.....	7
Additional information for threads	7
API change for SSLSessionEvent	7
JacORB timeout properties to set in Notification Service scenarios:	7
JacORB ImR and OpenFusion Name and Notification Service as Unix services.	7
JacORB IORMutator.....	8
New Features	9
3.7.0.0: This Release.....	9
3.0.0.0.....	10
2.3.0.3.....	11
2.3.0.2.....	11
2.3.0.1.....	12
2.3.0.0.....	12
2.1.3.3.....	13
2.1.3.2.....	13
2.1.3.1.....	14
2.1.3.....	15
2.1.2.....	15
2.1.1.....	15
2.1	15
2.0 beta	16
User Documentation.....	17
Known Issues	18

RMI issue with JDK 1.5.0	18
Encoding of indirections	18
JSSE idle_timeout with JDK1.3.....	18
Shutting down the OpenFusion ImR.....	18
Using the compactTypecodes property	18
Resolved Issues.....	19
Issues Resolved in this Release	19
Issues Resolved in Version 3.0.0	20
Issues Resolved in Version 2.3.1.0	21
Issues Resolved in Version 2.3.0.4	23
Issues Resolved in Version 2.3.0.3	27
Issues Resolved in Version 2.3.0.2	28
Issues Resolved in Version 2.3.0.1	29
Updates and SupportLine	31
Further Information and Product Support	31
Disclaimer.....	31

Micro Focus OpenFusion JacORB 3.7.0.0

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 *Setup* program. These programs can be run using a Graphical User Interface (GUI Mode) or from the command line (Command Line Mode) which enables the installation to be run from a script.

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 both products.

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 new OpenFusion Implementation Repository (IMR) is used by OpenFusion JacORB to locate and activate CORBA object implementations. The new OpenFusion IMR was designed so that it can also be used to locate object implementations based on OpenFusion TAO or OpenFusion e*ORB. The OpenFusion IMR was designed to provide a single unifying IMR that can be used across OpenFusion ORBs. However, in this initial release of the OpenFusion IMR, support will only be 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. (Added JacORB 2.1.3+)

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. (Added JacORB 2.1.3+)

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. (Added JacORB 2.1.3+)

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`. (Added JacORB 2.1.3+)

Monitoring TCP and SSL connections

The `jacorb.properties` file now includes new properties to allow a developer to create listeners to monitor TCP and SSL connections. The developer has to implement an interface (that uses the Java `EventListener` pattern) in order to receive notifications upon

for example TCP connections, SSL sessions. See the *JacORB Programming Guide* for more information on JacORB Network Event Logging. (Added JacORB 2.1.3+)

TCPConnectionEvent and SSLSessionEvent are extended to include a method to return the local IP. (Added JacORB 2.1.3.1+)

SecureRandom initialisation.

To allow developers to plug-in their own `java.security.SecureRandom` initialisation when using SSL. See the *JacORB Programming Guide* for more information on `jacorb.security.randomClassPlugin`. (Added JacORB 2.1.3+)

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 programming guide for more information. (Added JacORB 2.1.3+)

API change for SSLSessionEvent

Change to API for SSLSessionEvent to receive and provide cause of Exception. SSL logging also provides exception cause. (Added JacORB 2.1.3+)

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 ImR and OpenFusion Name and Notification Service as Unix services.

JacORB ImR and OpenFusion Name and Notification Service can be installed as an RPM (RedHat Package Manager) and integrated as Unix services.

- An RPM called `openfusion` installs the distribution to the `/opt/openfusion` directory and an associated script called `openfusion` to the `/etc/init.d` directory to ensure that the services are run as Unix services on machine boot.
- A service wrapper script called `serviceWrapper.sh` is installed with the RPM into the `/opt/openfusion/bin` directory. This script is responsible for starting the JacORB ImR and OpenFusion Services, detecting multiple start attempts, configuring a timeout and detecting if a Service is already running. This script takes the following options:

`[-t <timeout>]`: The timeout value that the script waits for a service to start. If not specified, it defaults to 180 seconds. The timeout value should be specified in seconds.

`-start|-stop|-status`: A command to start, stop or get the status of the services.

The `-t` parameter is optional. The ImR, Name and Notification Services are started and stopped sequentially by the `-start` or `-stop` option.

- So, to start the services, you need to run `/etc/init.d/openfusion -start` (or reboot the machine) which calls onto the `serviceWrapper.sh` script with the `-start` option. If you want to configure the timeout value, edit the `openfusion` script to put `-t <timeout>` before `-start`.
- The `openfusion` script supports `chkconfig` and the services are active in levels 2,3,4,5. The script executes last in the boot sequence (or first when shutting down).

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 are listed below according to the JacORB release version number. Note that bug fixes are listed in the [Resolved Issues](#) section.

3.7.0.0: This Release

- ORB
 - Fix threading issues in the ORB core
 - Improved client handling of references with multiple profiles.
 - Fix for threading issue with CodeSet (BZ994).
 - Fix for threading issue with Configuration (BZ991).
 - Fix for `read_fixed` (BZ996).
 - Fix for nested invocations and timeouts (BZ940).
 - Further improvements to network address retrieval code. Real interfaces (e.g. Ethernet/WLAN) will be prioritised over virtual interfaces (e.g. VMWare, VirtualBox, Docker) via the JVM system property `jacorb.network.virtual` which defaults to `VirtualBox,VMWare,vboxnet,docker`.
 - Fixes for IPv6/MIOP parsing.
 - Remove `deferredArrayQueue` to avoid stream corruption/improve performance.
 - Fix for client reconnecting to restarted servers (BZ967).
 - Fix for close issues in GIOPConnection (BZ1002).
 - Add compatibility check for special operations.
 - Ensure IIOP close doesn't throw exceptions.
 - Handle reconnecting via a closed connection to a persistent server (BZ975).
 - Fix for CodeSet tag appearing twice in multi-profile IORs.
 - Deprecate defining `org.jacorb.orb.standardInterceptors.IORInterceptorInitializer` as a standard interceptor; it is now implicitly defined by `jacorb.codeset QoS`.
 - Fix for 4 byte UTF-8 conversion (BZ969).
 - Codeset configuration: by default they are now enabled.
 - Fix for ORB instances sharing Current information.
 - Fix for forward request handling for special operations (BZ#979).
 - Fix for connection closure/server restart with multiple connections (BZ#986).
 - Default Sync scope for oneways is now configurable via `jacorb.default_sync_scope` property.
 - Add new `getDelegate` to `ClientRequestInfoImpl` Public API.
 - Fix interceptor `BAD_PARAM` minor codes (BZ961).
 - Default to closing client connections completely on error.
 - Rename QoS `jacorb.delegate.disconnect_after_systemexception` to `jacorb.connection.client.disconnect_after_systemexception`. (BZ967)
 - Implement `ContextList` (BZ968).
 - Split `jacorb.jar` into `jacorb.jar` and `jacorb-omgapi.jar`.
 - The OMG stubs are now held solely within `jacorb-omgapi`.
 - `PrintIOR` can now interpret `TAG_RMI_CUSTOM_MAX_STREAM_FORMAT`.
 - Fix for built-in interceptors and local calls (BZ956/957).
 - Add new `isLocalInterceptors` to `*RequestInfoImpl` Public API.
 - Add public ctor to `CDROutputStream` to allow specifying a predetermined buffer size, as well as a `releaseBuffer` method to let the application take ownership of the buffer from the stream for use after the stream goes out of scope. (BZ958)
 - Implement the method `validate_connection`.

- Support setting the ORB ID using the `-DORBId` option.
 - Support multi-network interfaces using `-ORBListenEndpoints` option.
 - Add public API for Client/ServerRequestInfo.
 - Add fix for NIO with large requests (BZ939).
 - Fix for UCS-2 codeset encoding.
 - Fix for RMI-IIOP Exception marshalling.
 - Only call `removeContext` if `addContext` has been called (Bug 925).
 - Call `_this_object` on Servants for `ServantManager` created objects.
 - When selecting an address for localhost attempt to find non-loopback first for both IPv4 and IPv6.
 - Fix for portable interceptor ordering issue with colocated calls.
- POA
 - Add spec-compliant minor codes for POA raised system exceptions.
 - Improve performance of thread pool on high-volume systems.
 - Add fix for exception in interceptors breaking requestprocessor thread. (BZ946)
 - Add fix for requestprocessor pooling. (BZ946)
- IDL Compiler
 - Add flag to turn off timestamps in generated code.
 - Updated IDL files to <http://www.omg.org/spec/CORBA/3.1.1>
 - Updates for IDL to Java Language Mapping 1.3 (formal/2008-01-11)
 - Fix for `i2jpackage` option causing rejection of valid module definition.
- TAO ImR/NS Compatibility
 - Implement the new property `"jacorb.use_tao_imr"`.
 - Support redundancy for failing over to a secondary ImR or a NS service when the current service is no longer available. JacORB will iterate the profiles the ImR-ified IOR while attempting to re-establish conversations with the targeted ImR or Naming Service (NS).
 - JacORB is inter-operable with the TAO ImR by setting the new property `"jacorb.use_tao_imr"` to `"on"`.
 - Add demo and integration test suites for TAO ImR, and TAO NS.

3.0.0.0

- ORB
 - Removed `finalize` from `Delegate` to improve performance and scalability. Client code is now responsible for calling `_release` on CORBA Objects as necessary (JAC#741).
 - Fix so the optimised key cache will auto-switch off for both client and server if it detects an older `jacorb` version (JAC#690).
 - Added `jacorb.key.cacheSize` property to allow the optimised key cache to be configured and improved purging algorithms (JAC#656).
 - Enhanced `disableServiceContextNegotiation` property to disable on both client and server side (and fixed its spelling) (JAC#656).
 - Removed `sync` from `GIOPConnection.getTransport` to fix deadlock with callback issue (JAC#649).
 - Enhancement to implement the correct use of portable interceptors when dealing with local calls, including the handling of `ForwardRequests` (JAC#647).
 - Added dummy implementation of `Delegate::_get_component`, which returns null (JAC#645).

- Enhancement to add the `jacorb.enableNullString` option to marshall null strings (JAC#639).
- An implementation of the `Object::create_request` operation has been added with an exception list (JAC#631).
- Fix for `NullPointerException` access in `org.jacorb.orb.Delegate` (JAC#772).
- Allow `OAIAddr` to be specified when using `PortRangeSocketFactory` (JAC#629).
- Split `jacorb.disableServiceContextNegotiation` into a separate property from `jacorb.interop.comet` (JAC#577).
- Added `Object._get_component ()` to OMG stub code from OMG 03-01-02 (JAC#489).
- JacORB now implements the Endorsed Standards Override Mechanism as described in <http://java.sun.com/j2se/1.5.0/docs/guide/standards/index.html> (RTJ#736).
- POA
 - Modified `AOMRemovalThread` to shutdown on POA destruction (JAC#696).
- IDL Compiler
 - An enhancement has been made to the idl compiler to implement the correct use of portable interceptors when dealing with local calls (JAC#647). idl generated code must be recompiled as the idl compiler changes will now cause failures for cases involving local objects and portable interceptors. This is because:
 - The `SEND_REPLY` or `SEND_EXCEPTION` points on the server side will not get called if a call is invoked on a local server. This will result in a `BAD_INV_ORDER` exception when trying to access attributes that are not available because the call status has not been set to `send_reply` or `send_exception`.
 - If a `ForwardRequest` is thrown from `servant_previnoke` (and so the `ServantObject` is null) a `CORBA.UNKNOWN` exception will be thrown with the message "local invocations not supported".
 - Some interception points will not get called, as described above and also at the `RECEIVE` stages on the client side if an invocation is made to a local servant.
 - Enhancement to the IDL compiler to generate a `toString ()` implementation for Union types (JAC#643).
 - Enhancement to ant `JacIDL` task to pass `sloppyidentifiers` and `i2jpackagefile` through, add error checking and add comments to generated code to silence Eclipse warnings (JAC#608).
- Bug fixes

See the [Resolved Issues](#) section.

2.3.0.3

- Bug fixes

See the [Resolved Issues](#) section.

2.3.0.2

- Bug fixes

See the [Resolved Issues](#) section.

2.3.0.1

- Bug fixes

See the [Resolved Issues](#) section.

2.3.0.0

- Bug fixes

See the [Resolved Issues](#) section.

- ORB

- add IORMutator plugin.
- add property to disable client orb policies to improve performance.
- change default connect_timeout to 90000 (JAC#217).
- new property: jacobc.codeset. By default this is on. If off, JacORB will ignore all codeset component info profiles and will turn off explicit marshalling with codesets (re Strings). Fixes for char transmission.
- New properties are available for assigning protocol-independent addresses:
 - OAddress=<protocol_name>://<protocol_address> for example:
OAddress=iiop://myhost:1234
 - OAIAddr and OAPort are still used to make IIOP-specific addresses but are superceded by OAddress.
 -
 - jacobc.ior_proxy_address=<prot_name>://<prot_addr> supercedes jacobc.ior_proxy_host and jacobc.proxy_port
 - jacobc.imr.ior_proxy_address

Note that OASSLPort is still necessary when configuring a specific SSL port number even when using OAddress.

- ORB configuration is more flexible now:
 - a new, common properties file orb.properties is considered.
 - multiple ORBs in a single JVM can now have separate configurations.

(Please consult the *JacORB Programming Guide* for details).

- IDL

- optimized generated code when using octet sequences.
- The IDL compiler can now generate DII stubs, using the new option "-diistub".

- Tools

- Updated 'dior' utility to decode realtime policies.

- Security

- SecurityLevel 2 and IAIK support removed.

- Documentation

- Enhanced description of configuration properties.
- Improved SSL documentation.

- Logging

- singleton ORB uses a logfile if the property jacobc.logfile is set.

- General
 - JacORB now supports IPv6.
 - removed scalability bottleneck due to superfluous synchronization.

2.1.3.3

- Bug fixes
 - See the [Resolved Issues](#) section.
- ORB
 - Minor performance improvement for `enhanced_thread_name`.
 - CDRStream typecode marshaling cleaned up.
 - Any: added better support for Streamable values.
 - Added `org.jacorb.orb.factory.FixedAddressSocketFactory` that binds client sockets to the address specified by the property "OAIAddr".
 - Logging additions for debugging.
 - Minor enhancement to allow release of client receptor threads in callback situations not using Bidirectional Policy.
- IDL
 - Use any speed accessors like `insert_long/extract_long` in Helper classes for aliased basetypes.
 - Enhance comment in idl generated classes.
- ImR
 - Add option so that ImR can be remotely shutdown.

2.1.3.2

- Bug fixes
 - See the [Resolved Issues](#) section.
- ORB
 - enhanced TCP and SSL logging information by also providing local IP address and port.
 - Change to API for `SSLSessionEvent` to receive and provide cause of Exception. SSL logging also to provide exception cause.
 - optimizations for interceptors within Delegate.
 - Added `AcceptorExceptionHandler` for notifications by the serversocket acceptor thread. NOTE: The default behaviour now is to shutdown the ORB whenever an Error or `SSLException` is thrown by `accept()` It will only shutdown for `SSLExceptions` on the first invocations of `accept()` . This property may be turned off. See the *Acceptor Exception Event Plugin* paragraph under section 3.3 *Configuration Properties* of the *JacORB Programming Guide*.
 - Added new configuration property `jacorb.enhanced_thread_name` to allow additional information to be added temporarily to the Thread name.
- Documentation
 - Removed IDL compiler from JavaDoc generation.
 - Improved command line and property documentation.

2.1.3.1

- Bug fixes
 - See the [Resolved Issues](#) section.
- ORB
 - Improved performance.
 - Support for client protocol properties added.
 - Pass `SystemException` message back in `ExceptionDetailMessage` service context.
 - new property: `jacorb.ORB_initializer.fail_on_error`. Control if errors caused by an `ORB_initializer` will make `ORB.init()` fail or not.
 - new property: `jacorb.listener.server_socket_timeout`. This is a workaround for Java bug #4344135 and is only useful in conjunction with the SI&C SSL socket factories.
 - new properties: `jacorb.connection.client.max_idle_receptor_threads`, `jacorb.connection.client.max_receptor_threads`, `jacorb.connection.server.max_idle_receptor_threads` and `jacorb.connection.server.max_receptor_threads` to limit the threadpools used for client and server side connections.
 - new property: `jacorb.ignoreComponentInfoProfiles`. This forces JacORB to ignore all codeset component info profiles.
 - new properties: `jacorb.connection.server.keepalive` and `jacorb.connection.client.keepalive` to configure TCP KeepAlive on sockets.
 - new properties: `jacorb.net.tcp_listener` and `jacorb.security.ssl.ssl_listener` for specifying the respective listener interfaces to receive event notifications.
 - Pass `SystemException` message back in `ExceptionDetailMessage` service context.
- POA
 - new property: `jacorb.poa.thread_pool_shared`. This is to share a single thread pool amongst all the POAs.
- IMR
 - Added support for failover across subnets using the new properties `jacorb.imr.identifier` and `jacorb.imr.other_imrs`.
- IDL
 - Add option (`-genEnhanced`) to generate `toString/equals` on stubs. Only implemented in `StructType` so far.
- Security
 - new property: `jacorb.security.randomClassPlugin`. Used by developers to plugin their own `SecureRandom` initialisation.
 - Change to improve information returned by `SSLSessionEvent`.
- Documentation
 - Added chapter on threads to the *JacORB Programming Guide*.
 - Enhanced logging documentation.

2.1.3

- Bug fixes

See the [Resolved Issues](#) section.
- ORB
 - Enhance `Any` equals for value comparison.
 - `DynValue` idl added.
- IDL
 - Added `CosProperty.idl`
- Documentation
 - Improve property documentation.
- General
 - Removed Appligator as no longer supported.

2.1.2

- Bug fixes

See the [Resolved Issues](#) section.
- ORB
 - There is a new property `jacorb.connection.client.connect_timeout`, which specifies a timeout for the initial TCP/IP connection attempt.
 - Removed property `jacorb.security.change_ssl_roles` and the associated "Change SSL Roles" feature.
 - The new property `jacorb.poa.check_reply_end_time` specifies whether any `ReplyEndTimePolicy` (or `RelativeRoundtripTimeoutPolicy`) should also be checked on the server side. This requires that the clocks of the client and server machine are synchronized to the same order of magnitude as the timeout. The default for this property is off, unlike previous releases, where this check was always done.
- IDL
 - Added detail to `idl -v` output.
 - IDL compiler has a new option `-in <file>` with which it will accept files that do not end in `".idl"`

2.1.1

- Bug fixes

See the [Resolved Issues](#) section.
- ORB
 - Added property to disable client orb policies for performance.

2.1

- Bug fixes

See the [Resolved Issues](#) section.

- ORB
 - Implementation of the ORB PolicyManager interface to set QoS policies ORB-wide.
 - CDR 1.2 Portable Interceptor codec.
 - Added codeset negotiation.
 - Removed `jacorb.timestamp` (now handled by `jacorb.log.default.log_pattern`).
 - Added `WrongTransaction` to idl.
 - Added DIOP IOR decoding ability.
 - Performance Improvements.
 - You can now use IORInterceptors to add new profiles to IORs (rather than just adding components to profiles that were created by the ORB). To achieve this, the IORInfo object that is passed to IORInterceptors now implements a special JacORB interface, `org.jacorb.orb.portableInterceptors.IORInfoExt`. This interface provides methods to look up existing profiles and add new ones to an IOR.
 - Rename property `jacorb.connection.server_timeout` to `jacorb.connection.server.timeout`, for consistency.
 - Added property `"jacorb.connection.client.timeout_ignores_pending_messages"` which controls if client-side idle timeouts take care of pending messages or not.
 - Added property `"jacorb.connection.client.retry_on_failure"` which controls if network failures on existing connections should yield a `COMM_FAILURE` or should trigger a remarshaling of all pending messages.
- POA
 - Added synchronization to the POAManager public methods (JAC#752).
- IDL Compiler
 - Removed deprecated option `-i`.
 - `"-cldc10"` option to generate J2ME/CLDC1.0 compatible stub code.
 - new option `-unchecked_narrow`.
- Logging
 - Better structured logger name spaces.
- InterfaceRepository
 - Documentation and initialisation improvements.
- IMR
 - New IMR added.
 - Documentation
 - Updated and merged JacIDL HTML into *JacORB ProgrammingGuide*.
 - Updated logging component descriptions.

2.0 beta

- ORB
 - All QoS timing policies are implemented.
 - `SyncScopePolicy` is implemented.
 - The transport layer now conforms to the draft of the *Extensible Transport Framework* (ETF, mars/2003-02-01) (see the *JacORB Programming Guide* for details).

- Two new properties, `jacorb.poa.queue_wait` and `jacorb.poa.queue_min` allow the server to block when the request queue is full.
- Fix for generation of some basic type sequence holders.
- Added server-side connection management (see the *JacORB Programming Guide*).
- Added support for GIOP CloseConnection messages.
- Two new properties, `jacorb.poa.queue_wait` and `jacorb.poa.queue_min` allow the server to block when the request queue is full.
- Added port range server socket factory support.
- Logging
 - Added property 'jacorb.logfile.maxLogSize' to support rolling logs.
- IDL Compiler
 - Fixed inheritance error (bug 345).
 - Now using the Apache logkit loggin API.

User Documentation

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

Known Issues

RMI issue 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.

The following 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

Because of 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 OpenFusion ImR

The `org.jacorb.imrutility.imr.RepositoryImpl.shutdown` method has been amended to ignore the boolean parameter 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.

Issues Resolved in this Release

- 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 J2ME Lightweight Naming Service in rebind/unbind.

JAC-68

- 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

Issues Resolved in Version 3.0.0

- Added ImR POA registration/deregistration so that two Implementation Repositories can run on different hosts and the Notification Service can register with both ImRs.
JAC832
- Fix incorrect processing of type ID of derived type in `orb.Any.read_value ()`.
JAC832
- Fix to prevent high CPU consumption in a MIOP client connection's threads.
JAC829
- Fix where binding and rebinding could lead to an infinite loop with an SSL connection.
JAC824
- Fix for MIOP domains and object groups processing.
JAC823

- Added `RequestAnalyser` and `RequestProcessorAnalyser` plugins for custom handling of requests. Added new documentation, section 6.3 of the *JacORB Programming Guide*. Supplied an example `RequestAnalyser` and two `RequestAnalyserProcessor` implementations. Added property `jacorb.poa.checkRequestIDs`. When set, this forces replies on the same connection to be sent back in the same order, by checking the RequestIDs. This property is designed to work in correspondence with the `RequestAnalyser` plugin. Default = `off`.

JAC822

Issues Resolved in Version 2.3.1.0

- Fix to allow configuration of SSL keystore types and algorithms.

JAC818

- New MIOP ETF transport added.

JAC810

- Fix `toString` on `Union` and refactor `UnionType`.

JAC801

- Fix for incorrect child POA processing.

JAC798

- Add NAT and firewall network configuration section to the *JacORB Programming Guide*.

JAC794

- Fix for local interceptors to prevent client `PICurrent` being overwritten by a server `PICurrent`.

JAC788

- Fix for decoding and output of IORs with multiple `TAG_ALTERNATE_IIOP_ADDRESS`.

JAC779

- Fix for locating valid `ImRs` in ORB and in `IMRLocator`.

JAC777

- Fix for potential deadlock in `ImRRebindRetryStrategy` when communication with current `Implementation Repository` is broken by network errors.

JAC777

- Fix for `NegativeArraySize` exception caused by `BufferManager` usage in `CDROutputStream`.

JAC774

- Pluggable Expansion policy added to Buffer Manager to allow control of memory allocation. Options are:
 - `log` (default). For small replies, Buffer Manager allocates buffers bigger than requested to prevent frequent buffer reallocation and reduce memory usage. For buffer sizes in excess of 180MB, Buffer Manager allocates buffers with exactly the requested sizes
 - `linear` (pluggable). Returns a buffer with exactly the requested size
 - `double` (pluggable). Returns a buffer with twice the requested size

Stream closure and logging improved.

JAC773

- Fix for NullPointer access in `org.jacorb.orb.Delegate`.

JAC772

- Fix for high CPU load issue on Windows hosts caused by default configuration re-reading.

JAC770

- Remove DDS files from JacORB distribution.

JAC765

- Fix for IDL compiler issue where a union with an enum discriminator could send through a non-defined value.

JAC755

- Support for typeprefix has been added to the idl compiler.

JAC753

- Support for attribute exceptions (`getraises/setraises`) has been added to the JacORB idl compiler.

JAC753

- Added synchronization to the POAManager public methods.

JAC752

- Enhance the build options that are bundled with the JacORB source. For more information, extract the JacORB source file `<DISTRIB_DIR>/src/src.tar.gz` and, using Ant 1.7.1, run `ant -projecthelp` on the top level `build.xml` file. See <http://ant.apache.org/manual/running.html#options> for more details.

JAC749

- Copy the `idl`, `dior` and `jaco` scripts from the source tree to the distribution bin directory and patch them so that they work in the new location.

JAC749

Issues Resolved in Version 2.3.0.4

- Remove `finalize` from `Delegate` to improve performance and scalability. Client code is now responsible for calling `_release` on CORBA Objects as necessary.

JAC741

- Fix IDL compiler issue with constant long long generation.

JAC737

- Fix for multithreading and fault-tolerant retries.

JAC723

- Fix that allows `IORMutator` class caching to prevent `CDRInputStream` and `CDROutputStream` threads blocking.

JAC722

- Fix to propagate `objectId` to `ServerRequestInfo` for local calls.

JAC704

- Fix for non-configured `CDRInputStream` when created with an `ORBSingleton`.

JAC703

- Modified `AOMRemovalThread` to shutdown on POA destruction.

JAC696

- Fix so that the enhanced context key system will automatically switch itself off if it detects an older JacORB version. This avoids the requirement of using the `disableServiceContextNegotiation` property, although it is still available.

JAC690

- Fix to ensure that current contexts are added and cleared correctly when handling embedded calls.

JAC685

- Fix an issue that was preventing JacORB servers from registering with the TAO ImR when the TAO ImR runs on a little endian machine.

JAC681

- Fix an issue where creating a persistent POA with an already active POA manager instance causes duplicate registrations with the ImR.

JAC680

- Change to ensure that the `REQUEST_DURATION_POLICY` timeout is not overwritten if an interceptor makes an embedded invocation.

JAC677

- Fix so that the idl compiler detects bounds violations for the string type.
JAC676
- Change to request handling to ensure that Relative Roundtrip Timeout will bound the entire user invocation including any transparent re-inocations following LOCATION_FORWARD or other retryable exception conditions and also embedded invocations.
JAC670
- Fix delegate bind issue to prevent cached object delegate being set to null.
JAC663
- Fix a memory leak in Delegate `pending_replies` if an error occurs when reading the reply.
JAC662
- Fix for Current overwriting in local calls between Client/ServerRequestInfo.
JAC660
- Added `jacorb.key.cacheSize` property to allow the optimised key cache to be configured and improved purging algorithms.
JAC656
- Fix spelling of `disableServiceContextNegotiation` property and property now functions on client and server side.
JAC656
- Implementation of `get_component` operation.
JAC653
- Fix to reduce the number of threads created when deactivating an object; `backport-util-concurrent.jar` has been added and should be used when the JDK is pre 1.5.
JAC652
- Improved documentation for threads and SSL / TLS in the *JacORB Programming Guide*.
JAC652 / JAC651
- Fix for attribute setting of the `jacorb.security.ssl.client.protocols` and `jacorb.security.ssl.server.protocols` properties.
JAC651

- Synchronisation fix for `GIOPConnection.getTransport`.

JAC649

- Enhancement to implement the correct use of portable interceptors when dealing with local calls, including the handling of `ForwardRequests`.

JAC647

- Fix to improve exception handling in `POA::reference_to_id`, which returned a bad oid value under certain conditions.

JAC646

- Enhancement to add a dummy implementation of `Delegate::_get_component()`, which returns null.

JAC645

- Enhancement to the IDL compiler to generate a `toString()` implementation for Union types.

JAC643

- Fix a memory leak in `ObjectKeyMap`.

JAC642

- Enhancement to add the `jacorb.enableNullString` option to marshall null strings.

JAC639

- An implementation of the `Object::create_request` operation has been added with an exception list.

JAC631

- Fix for `NullPointerException` when multiple threads reference the same CORBA Object, one of those threads releases that reference and another thread attempts to invoke the reference.

JAC630

- Fix to allow `OAIAddr` to be specified when using `PortRangeSocketFactory`.

JAC629

- Fix for `InterruptedException` in `IIOPListener` from `LogKit` and native code.

JAC622

- Fix for IDL generated code where the execution of `toString()` throws `NullPointerException` when a class contains an array of `char`.

JAC617

- Fix for OpenFusion Naming Service hang when the service is stopped and started several times.

JAC609

- Enhancement to ant JacIDL task to pass sloppyidentifiers and i2jpackagefile through, add error checking and add comments to generated code to silence Eclipse warnings.

JAC608

- Fix issue on Windows where values are not escaped in `jaco.bat`.

JAC605

- Fix for initialising the ORB with properties to avoid a `java.lang.ClassCastException`.

JAC600

- JacORB now implements the Endorsed Standards Override Mechanism as described in <http://java.sun.com/j2se/1.5.0/docs/guide/standards/index.html>.

RTJ736

- In the `DynUnion.to_any ()` operation, a null pointer exception occurs because no check is made that the member is not null.

JAC590

- Fix to add `Object._get_component ()` to OMG stub code from OMG 03-01-02.

JAC589

- The idl generated `BooleanUnionHelper::read ()` operation is incorrect: if demarshalling a union with discriminator set to true, the result will have discriminator set to false.

JAC581

- A JacORB fault tolerant client invokes a oneway call on an IOGR in a TAO fault tolerant server. This causes the server to crash and a second TAO server now becomes the primary one. Since the client set the `SYNC_WITH_SERVER` policy, JacORB should retry the request on the second profile in the IOGR and hence contact the new primary. Instead, JacORB throws a `COMM_FAILURE` for the first profile in the IOGR and does not retry the request on the new primary. The client does not receive an exception as it is a oneway request.

JAC579

- The `jacorb.disableServiceContextNegotiation` property has been created as a separate property to `jacorb.interop.comet`. This disables the sending of an extra service context on initial negotiation, which in turn disables JacORB's optimised object key handling, but it allows the real object id to be detected, for example, by the Recording tool, which searches the network for CORBA requests

and filters according to object id. The default value is `off`.

JAC577

- A JacORB client contacts an agent and makes a request. The agent replies with a location forward to a reference in a JacORB server. The client interacts with the reference in the JacORB server, the server goes down, the client correctly receives an `org.omg.CORBA.TRANSIENT` exception and the client falls back to the initial IOR in the agent. At a later date, the agent establishes that the same permanent reference in the JacORB server is reachable again and forwards the JacORB client on to it. However, JacORB remembers that the server IOR failed before and an `org.omg.CORBA.TRANSIENT` exception is raised again. This behaviour is incorrect by default, according to chapter 15.6 of the CORBA specification. The JacORB client should interact with the reference in the JacORB server again when it receives a location forward from the agent. However, if the JacORB server is still down, JacORB contacts the agent, which performs a location forward to the JacORB server, which is down. JacORB should then raise an `org.omg.CORBA.TRANSIENT` exception to the client, rather than again attempt to connect to the JacORB server in an infinite loop.

JAC574

- An 'already defined' error occurs in the idl compiler if a `struct` is typedef'd to an array outside the scope of a module.

JAC572

- An SSL-enabled server cannot communicate with an SSL-enabled client when bi-directional GIOP is enabled.

JAC563

-

Issues Resolved in Version 2.3.0.3

- A Repositoryid using pragma prefix with more than one space before id causes an idl compiler parse error.

JAC569

- Property `jacorb.connection.server.reuse_address` added. This enables the socket option for `SO_REUSEADDR`. Default is `off`.

JAC567

- The `read ()` and `extract ()` methods in Helper classes generated by the JacORB idl compiler can cause memory leaks with RTOrb.

JAC555

- Enhancement to allow the `toString` method on idl-generated Struct stubs to print out the contents of a 2-dimensional array.

JAC550

- Property `jacorb.dns.eager_resolve` added. If `on`, allows eager, rather than lazy, DNS resolution. Default is `on`.

JAC542

- If `use_imr=on`, the Name Service does not start with `ssl`. A property `jacorb.imr.passIORcomponent` has been added. If enabled, the IOR component information is also accessible to the ImR. The property is set to `off` by default. A side effect is that the component information is now added twice to the IOR, resulting in a bigger IOR.

JAC531

- JacORB had a static dependency on the `javax.naming` package but J2ME does not include this package.

JAC526

- Resource temporarily unavailable message received: to work around this, attempt to avoid creating network connections for in-process communications by utilising loopback protocol. (Note that this is not the network stack loopback device).

JAC524

- `jac_imr` does not start on Windows.

JAC520

- A deadlock occurs when two resources try to access `org.jacorb.poa.RequestController`.

JAC519

- AMI exceptions do not work if the IDL is specified with `pragma` prefix.

JAC511

- A `COMM_FAILURE` occurs when narrowing a reference to the Name Service using `SSL`.

JAC486

Issues Resolved in Version 2.3.0.2

- The `CosNamingME` package is missing from the JacORB ME distribution.

JAC508

- The `Object.set_policy_overrides ()` method applies policies on the current object and returns the current object; it should create a copy of the current object, apply policies to this copy and return the copy.

JAC493

- Holder logger is not being set, which causes a `NullPointerException`.

JAC492

- Creating an empty any and calling `create_input_stream ()` can lead to an `ArrayIndexOutOfBounds`.

JAC491

- Holder classes for user-defined exceptions are not generated by the idl compiler.

JAC480

- `org.jacorb.naming.NameServer` is not in `jacorb.jar` when you build the source.

JAC470

- The Interface Repository is not persistent; its IOR's object key changes when the Interface Repository restarts.

JAC465

- A repository id is built unnecessarily when marshalling a valuetype, which causes a latency overhead.

JAC449

Issues Resolved in Version 2.3.0.1

- The `ValueBaseHelper` class has no implementation.

JAC444

- The JacORB idl compiler does not allow you to use the same variable name as type name.

JAC441

- `PortableServer.Servant` type becomes an int type when the JacORB idl compiler runs with an `RTOrb` idl file, but not with a `JacORB` idl file.

RTJ519

- A custom `ssl` socket factory cannot be instantiated.

JAC417

- A null pointer exception occurs on the `POAUtil.ExtractOID ()` method when a null reference is passed to the `POA.reference_to_servant ()` method.

JAC415

- Multiple empty log files are created during the log file initialisation routine.

JAC384

- The *JacORB Programming Guide* does not contain information about the `jacorb.security.keystore_password` property, which is needed if you want to use the `ImR` with `SSL` enabled.

JAC381

- Using the SocketFactories with the new JDK 1.4 socket API, the wrong exception type is thrown when a timeout is hit.

JAC375

- Unable to load IDL file into JacORB Interface Repository where #pragma prefix "jacorb.ir" is defined in IDL file.

JAC367

- Marshalling error when interop with the Oracle (formerly Sun) JDK 1.4 ORB.

JAC352

- The `-omg` option on the JacORB idl compiler should be removed and replaced with `i2jpackage` in `idl/build.xml` and documented in the *JacORB Programming Guide*.

JAC72

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.

Micro Focus is a registered trademark.
Copyright © Micro Focus 2016. All rights reserved.