Sun Java System Message Queue Release Notes

Version 3 2005Q1 (3.6)

Part Number 819-0064-12

These release notes contain important information available at the time of release of Sun Java™ System Message Queue 3 2005Q1 (3.6). New features and enhancements, known issues and limitations, and other information are addressed here. Read this document before you begin using Message Queue.

The most up-to-date version of these release notes can be found at the Sun Java System Message Queue documentation web site: `http://docs.sun.com/coll/MessageQueue_05q1`. Check the web site prior to installing and setting up your software and then periodically thereafter to view the most up-to-date release notes and product documentation.

These release notes contain the following sections:

Third-party URLs are referenced in this document and provide additional, related information.

NOTE Sun is not responsible for the availability of third-party Web sites mentioned in this document. Sun does not endorse and is not responsible or liable for any content, advertising, products, or other materials that are available on or through such sites or resources. Sun will not be responsible or liable for any actual or alleged damage or loss caused by or in connection with the use of or reliance on any such content, goods, or services that are available on or through such sites or resources.

Release Notes Revision History

Table 1 Revision History

Date	Description of Changes
August 1, 2005	Add Hardware and Software Requirements for Windows operating systems. Add Windows information in the section "Installation Issues" on page 21.
May 1, 2005	Update the section "Issues Related to the Next Major Release of Message Queue" on page 16 with new information. Add an Interface Stability statement in the Compatibility section.
January 24, 2005	Update in synch with revenue release of Sun Java System Message Queue 3 2005Q1 and Java Enterprise System 2005Q1.
November 8, 2004	Initial release of *Sun Java System Message Queue Release Notes*.

About Message Queue 3 2005Q1 (3.6)

Sun Java System Message Queue is a full-featured message service that provides reliable, asynchronous messaging that conforms to the Java Messaging Specification (JMS) 1.1. In addition, Message Queue provides a host of features that go beyond the JMS specification to provide for the needs of large-scale enterprise deployments.

This section includes:

- "What's New in This Release" on page 2
- "Hardware and Software Requirements" on page 5

What's New in This Release

New features in Message Queue 3 2005Q1 (3.6) include:

- "Dead Message Queue"
- "No Acknowledge Mode"
- "Client Message Body Compression"

- "Connection Failure Detection (Client Runtime Ping)"
- "Certificate Management: C-API NSS Tools"
- "Support for C-API Basic Authentication"
- "64-bit C-API Support"

These are described in the following sub-sections.

Dead Message Queue

Message Queue automatically creates a specialized destination at broker startup that is used to store dead messages for diagnostic purposes. A *dead message* is one that is removed from the system for a reason other than normal processing or explicit administrator action. A message might be considered dead because it has expired, because it has been removed from a destination due to memory limit overruns, or because of failed delivery attempts.

An administrator can configure destinations to either discard dead messages or place them in the dead message queue. When placed in the dead message queue, additional property information is written into the message, providing an administrator with information about the cause of death. A client developer can also set a property value when creating a message that determines whether the message should be placed in the dead message queue were it to die.

For more information, see the *Message Queue Administration Guide*.

No Acknowledge Mode

The `NO_ACKNOWLEDGE` acknowledgement mode is an extension to the JMS API. Normally, the broker waits for a client acknowledgement. That acknowledgement must be made programmatically if the client has specified `CLIENT_ACKNOWLEDGE` or it can be made automatically, by the session, if the client has specified `AUTO_ACKNOWLEDGE` or `DUPS_OK`. If a consuming client specifies the `NO_ACKNOWLEDGE` mode, the broker discards the message as soon as it has sent it to the consuming client. This feature is intended for use by non-durable subscribers consuming non-persistent messages, but it can be used by any consumer.

Using this feature improves performance by reducing protocol traffic and broker work involved in acknowledging a message. This feature can also improve performance for brokers dealing with misbehaving clients who do not acknowledge messages and therefore tie down broker memory resources unnecessarily. Using this mode has no effect on producers.

There is no support for the `NO_ACKNOWLEDGE` mode in C clients. For more information, see the *Message Queue Developer's Guide for Java Clients*.

Client Message Body Compression

The developer can specify that the body of a message can be compressed. Message compression and decompression is handled entirely by the client runtime and does not affect the broker. Therefore, applications can use this feature with a previous version of the broker, but they must use version 3 2005Q1 (3.6) of the Message Queue client runtime library.

Advantages and Limitations of Compression

Although message compression has been added to improve performance, such benefit is not guaranteed. Benefits vary with the size and format of messages, the number of consumers, network bandwidth, and CPU performance. For example, the cost of compression and decompression might be higher than the time saved in sending and receiving a compressed message. This is especially true when sending small messages in a high-speed network. On the other hand, applications that publish large messages to many consumers or who publish in a slow network environment, might improve system performance by compressing messages.

Message consumers deployed with client runtime libraries that precede version 3 2005Q1 (3.6) cannot handle compressed messages. Clients configured to send compressed messages must make sure that consumers are compatible. C clients cannot currently consume compressed messages.

For more information, see the *Message Queue Developer's Guide for Java Clients*.

Connection Failure Detection (Client Runtime Ping)

Message Queue 3 2005Q1 introduces a new `ConnectionFactory` attribute named `imqPingInterval`. The `imqPingInterval` attribute specifies the frequency of a ping operation from the client runtime to the broker. By periodically testing the connection, the client runtime can preemptively detect a failed connection. If the ping operation fails, the client runtime throws an exception to the client application's exception listener object. If the application does not have an exception listener, the application's next attempt to use the connection fails.

For more information, see the *Message Queue Administration Guide*.

Certificate Management: C-API NSS Tools

The C-API uses NSS (Network Security Services) libraries to support SSL. These libraries provide APIs and utilities for developing secure applications. These utilities include tools to manage keys and certificate databases. In Message Queue 3.5, we asked developers to use Mozilla to manage NSS keys and certificates. In Message Queue 3 2005Q1, administrators can use the NSS `certutil` tool to generate the needed keys and certificates.

For more information, see the *Message Queue Developer's Guide for C Clients*.

Support for C-API Basic Authentication

Message Queue 3 2005Q1 C-API supports the `basic` authentication type. Previous releases of Message Queue did not support the `basic` authentication type.

64-bit C-API Support

Message Queue now contains 64-bit C-API support on the Solaris/SPARC platform. For more information about enabling 64-bit C-API support, see the *Message Queue Developer's Guide for C Clients*.

Hardware and Software Requirements

This section specifies or describes the hardware and software required for this release of Message Queue. Table 2 lists hardware and software requirements for Solaris SPARC operating systems.

Table 2 Solaris SPARC Hardware and Software Requirements

Component	Platform Requirement
Operating System	Solaris 8, Update 7 Solaris 9, Update 7 Solaris 10 (Includes 32 and 64 bit support for Solaris 8, Solaris 9, and Solaris 10)
CPU	Sun UltraSPARC™ **Note:** To ensure proper operation of Message Queue, you should install all required Solaris Patches for Java 2 Platform, Standard Edition 5.0. For the latest information about the patches and to download the recommended and required patches, see: `http://java.sun.com/j2se/1.5.0/download.jsp`
RAM	256 Mbytes
Disk space	The compressed installation file is approximately 27 Mbytes. The temporary working directory used for extracting the installation files requires an additional 27 Mbytes. The installed product requires approximately 27 Mbytes of hard drive space. Message Queue, however, may need more space if the broker stores persistent messages locally.

Table 3 lists hardware and software requirements for Solaris x86 operating systems.

Table 3 Solaris x86 Hardware and Software Requirements

Component	Platform Requirement
Operating System	Solaris 9, Update 7 Solaris 10 (Includes 32 and 64 bit support for Solaris 9 and Solaris 10)
CPU	Intel Pentium 2 (or compatible) **Note:** To ensure proper operation of Message Queue, you should install all required Solaris Patches for Java 2 Platform, Standard Edition 5.0. For the latest information about the patches and to download the recommended and required patches, see: `http://java.sun.com/j2se/1.5.0/download.jsp`
RAM	256 Mbytes
Disk space	The compressed installation file is approximately 27 Mbytes. The temporary working directory used for extracting the installation files requires an additional 27 Mbytes. The installed product requires approximately 27 Mbytes of hard drive space. Message Queue, however, may need more space if the broker stores persistent messages locally.

Table 4 lists hardware and software requirements for Linux operating systems.

Table 4 Linux Hardware and Software Requirements

Component	Platform Requirement
Operating System	Red Hat Advanced Server 2.1, Update 2 RedHat Advanced Server 3.0, Update 1 **Note:** The Message Queue software distribution includes the required JRE version at the time of release.
CPU	Intel Pentium 2 (or compatible)
RAM	256 Mbytes
Disk Space	The zip file containing the product is approximately 18 Mbytes. The installed product requires approximately 18 Mbytes of hard drive space. Message Queue, however, may need more space if the broker stores persistent messages locally.

Table 5 lists hardware and software requirements for Windows operating systems.

Table 5 Windows Hardware and Software Requirements

Component	Requirements
Operating system	Windows XP Professional SP2 Windows 2000 all editions, SP4 or higher Windows Server 2003, Enterprise Edition
CPU	Intel Pentium 3
RAM	256 Mbytes
Disk space	The installation file is approximately 150 Mbytes. The temporary directory used for extracting the installation files requires an additional 160 Mbytes. The installed product requires approximately 45 Mbytes (125 Mbytes if a new Java runtime must be installed). Message Queue, however, may need more space if the broker stores persistent messages locally.

Message Queue 3 2005Q1 also depends upon other technologies, as indicated in Table 6. Other versions or vendor implementations can also be used but they are untested by Sun Microsystems and therefore not supported.

Table 6 lists and describes the basic components that you must install in order to be able to develop and run Message Queue clients.

Table 6 Message Queue 3 2005Q1 Basic Product Support Matrix

Platform/Product	Used For	Supported Platform/Product Version
Java Runtime Environment (JRE) (Sun Microsystems versions only)	Message Queue broker (message server) and Message Queue administration tools	Java Runtime Environment 1.4.2_05 Java 2 Platform, Standard Edition, 5.0 (1.5): • Solaris 8 (SPARC), Solaris 9 (SPARC and X86), and Solaris 10 (SPARC and X86). • Linux Red Hat Advanced Server 2.1, Update 2, and Linux Red Hat Advanced Server 3.0, Update 1. • Windows XP Professional SP2, 2000 all editions SP4, Windows Server 2003 Enterprise Edition.
Java Software Development Kit (JDK), Standard Edition (Sun Microsystems production versions only)	Java client development and deployment (Java SOAP/JAXM clients are supported only on JDK 1.4.2 and 1.5)	JDK 1.4.2_05 Java 2 Platform, Standard Edition, 5.0 (1.5): • Solaris (same platforms as JRE) • Linux (same platforms as JRE) • Windows (same platforms as JRE)

Table 7 lists and describes the components that you can install to provide additional support for a Message Queue client. You might just need some of the components listed. For example, if you are not writing a C client to Message Queue, you will not need any of the components required for C client support.

Table 7 Message Queue 3 2005Q1 Optional Product Support Matrix

Product	Used For	Supported Product Version
LDAP Directory Server	Message Queue user repository and administered object support	Sun Java System Directory Server Version 5.2 SP 3
Web Server	HTTP and HTTPS support	Sun Java System Web Server, Enterprise Edition Version 6.1 SP 4
Application Server	HTTP and HTTPS support	Sun Java System Application Server, Enterprise Edition 8.1
Database	Plugged-in persistence support	PointBase, Version 4.8 Oracle 9i, Version 9.2
JNDI (Java Naming and Directory Interface)	administered object support	• JNDI Version 1.2.1 • LDAP Service Provider Version 1.2.2 • File System Service Provider Version 1.2 Beta 3 (supported for development and testing, but not for deployment in a production environment.)
C Compiler and compatible C++ runtime library	Message Queue C client support	• Solaris: Sun WorkShop 6, Update 2 or later C++ compiler with `-compat=5`, Sun WorkShop C compiler • Linux: gcc/g++ 2.96 • Windows: Microsoft Windows Visual C++ 6.0, SP3
NSPR (Netscape Portable Runtime)	Message Queue C client support	Version bundled with Java Enterprise System 2005Q1. For the Platform Edition, this is installed as a shared package.
NSS (Network Security Service)	Message Queue C client support	Version bundled with Java Enterprise System 2005Q1. For the Platform Edition, this is installed as a shared package.

Bugs Fixed in This Release

Table 8 describes the bugs fixed in Message Queue 3 2005Q1 (3.6):

Table 8 Fixed Bugs in Message Queue 3 2005Q1

Bug Number	Description
2091749	Broker gets Null Pointer Exception on shared thread pool under load.
2092975	Persistent store should continue to load if deserialization fails on one destination.
2092976	Periodic error processing message to topic when a consumer with a selector was being closed as the message was being routed through the system.
4888259	`vmargs` are getting overriden if we pass more than one to `imqbroker` on Windows.
4934433	Packet code should defend against large packets.
4941127	Destination will not completely load if a message exceeds individual msg size limit
4949398	`imqcmd query dst` reports wrong numbers while a destination is being loaded
4953659	Message Queue client never calls `ExceptionListener.onException` on half open connection.
4962906	Broker should detect stale database connection and reconnect.
4969880	`Receive(timeout)` returns `null` before timeout expires with frequently closing connections.
4970719	Messages with acks in prepared transaction may be resent to consumers.
4983525	Error creating producer on autocreated destination in linux 3.0 system.
4987799	Message Queue Resource Adapter and broker needs to support shared subscriptions for Application Server 8.1 Enterprise Edition Clusters.
4989708	`imqbrokerd` always behaves as if `imq.transaction.autorollback` is `true`.
4992645	Message Queue Resource Adapter: JMS Resource Adapter needs to support connection pooling.
4996776	Broker throws `ConcurrentModificationExceptions` with high client load plus metrics.
5004868	Deleted transactions are not removed from JDBC database.
5014570	QBrowser example does not work with Platform Edition.
5024685	Access Control Lists: `queue.create.deny.user=*` and `imq.autocreate.queue=true` interact poorly.
5025241	Durable subscriber with `noLocal=true` receives self-published messages.
5037962	Mismatch between master broker's state and log causes deadlocks at startup.
5042763	File store code should use `FileChannel.force(false)` to improve synchronous IO performance.

Table 8 Fixed Bugs in Message Queue 3 2005Q1 *(Continued)*

Bug Number	**Description**
5046995	Message Queue Java client protocol negotiation implementation does not work reliably.
5053565	Message Queue 3.5 SP1: Broker spins and burns CPU in a cluster with a master.
5063625	Queue receiver stops receiving messages with remote messages intended for it in two-broker cluster.
6021000	Deadlock in broker when adding and removing consumers under heavy load.
6057402	Master broker deadlock on restart if it had removed destination with durables.
6057450	`imqcmd`: Cannot set `maxNumBackupConsumers` to 0 for queues.
6155087	Broker with bad broker name and durable subscriber in cluster hangs on shutdown.
6155091	JDBC: "`-reset store`" takes a long time with large messages and Oracle
6157943	`imqbrokerd` service does not shutdown cleanly on Windows 2000 reboot.
6165743	Message Queue Resource Adapter: Repeated use of `connection.open/close` under stress could cause `IllegalStateException`.
6165984	`imqSSLIsHostTrusted` is ignored when `imqAddressList` is used to specify client to make SSL connection.
6170578	Message Queue Resource Adapter:`ManagedConnectionFactory` should inherit the properties that are set in the Resource Adapter java bean.
6170831	Message Queue Resource Adapter:MDB undeployment timing problem could cause Null Pointer Exception if messages are still being received.
6174532	JMS brokers redeliver messages that have already been consumed by MDBs.
6178549	Reload `cls` loses configured broker list if `-D imq.cluster.url` is used on command line
6189214	`MQ_MESSAGE_ID_HEADER_PROPERTY` (C) in received message differs from `JMSMessageID` in sent msg (Java).

Important Information

This section contains the latest information that is not contained in the core product documentation. This section covers the following topics:

- Installation Notes
- Compatibility Issues
- Documentation Updates for Message Queue 3 2005Q1 (3.6)

Installation Notes

Refer to the *Message Queue Installation Guide* for information about pre-installation instructions, upgrade procedures, and all other information relevant to installing Message Queue, Platform Edition on the Solaris, Linux, and Windows platforms.

Refer to the *Sun Java Enterprise System Installation Guide* for information about pre-installation instructions and all other information relevant to installing Message Queue, Enterprise Edition on the Solaris and Linux platforms.

Refer to the *Sun Java Enterprise System Upgrade and Migration Guide* for information about upgrade and migration instructions relevant to upgrading to Message Queue 3 2005Q1 (3.6) on the Solaris and Linux platforms.

Installing Missing Solaris Packages (Message Queue 3 2005Q1 Enterprise Edition only distribution (subset distribution) with Java Enterprise System Installer)

The Message Queue 3 2005Q1 Enterprise Edition Distribution for Solaris does not include the packages `SUNWtlsu` (Network Security Services Utilities) and `SUNWtlsux` (Network Security Services Utilities 64-bit SPARC only).

These packages are necessary only to support SSL C clients. A previous version of these packages may already be installed on your system, but you should update those packages to the version that is included in Java Enterprise System 2005Q1.

Workaround
You can get the version of `SUNWtlsu` and `SUNWtlsux` that is included with Java Enterprise System 2005Q1 from the Message Queue 3 2005Q1 Platform Edition bundles at the Sun Java System website. After you download the Message Queue files, follow the instructions in the section “Installing Message Queue” in Chapter 2, “Solaris Installation” of the *Message Queue Installation Guide*. In Step 8, answer **`n`** (no) and proceed to Step 9 to install just the `SUNWtlsu` and `SUNWtlsux` packages on your system.

Compatibility Issues

This section covers compatibility issues in Message Queue 3 2005Q1 (3.6).

Issues in Message Queue 3 2005Q1 (3.6)

The following sections describe issues that affect: all platforms, Solaris and Linux platforms, and Linux platforms only.

All Platforms

Interface Stability Sun Java System Message Queue uses many interfaces that can help administrators automate tasks. Appendix B in the *Message Queue Administration Guide* classifies the interfaces according to their stability. The more stable an interface is, the less likely it is to change in subsequent versions of the product.

Deprecation of Password Options The following options have been deprecated for security reasons:

- `-p`
- `-password`
- `-dbpassword`
- `-ldappassword`

If a password is specified as a part of a command such as the following:

```
imqcmd query bkr -u admin -p adminpassword
```

a user could see the administrator password while listing or querying a machine's processes (e.g. using `ps` on Solaris).

Use the `-passfile` option instead. The passfile option is explained in the chapter on security in the *Message Queue Administration Guide*.

Solaris and Linux Platforms

JDK 1.3 Client Support The following packages contain jar files used to support JNDI and JSSE for client development and deployment on JDK 1.3. These packages are shipped with Message Queue, Platform Edition only. These packages will not be delivered in a future release. These packages are not installed by default:

- `SUNWiqsup`--Solaris platform
- `sun-mq-sup`--Linux platform

For more information on installing packages separately, see the *Message Queue Installation Guide*.

Linux Platforms

This section describes the changes that affect the packages and installed locations on the Linux platform.

New Linux Install Locations The location of Message Queue installation directories has changed for Linux.

Table 9 shows the default installation directories for Message Queue RPMs on Linux

Table 9 Default install locations of Message Queue 3 2005Q1 RPMs

RPM Package Name	Default Install Location
`sun-mq-config`	`/etc/opt/sun`
`sun-mq-var`	`/var/opt/sun`
all other RPMs	`/opt/sun`

For more information, see Appendix A, "Operating System-Specific Locations of Message Queue Data," in the *Message Queue Administration Guide*.

Changed Linux RPM Package Names The names of Message Queue Linux RPM package files have changed.

Table 10 shows the new names of Message Queue 3 2005Q1 Linux RPM packages.

Table 10 RPM Packages in Linux Bundle

RPM	Description	Notes
`sun-mq-config`	`/etc` files	
`sun-mq-var`	`/var` files	
`sun-mq`	`/opt` files	Depends on `sun-javahelp`, `sun-mq-jmsclient`, `sun-mq-config`, and `sun-mq-var`
`sun-mq-ent`	Enterprise Edition license file.	This RPM is only shipped with Message Queue, Enterprise Edition, available with Java Enterprise System.
`sun-mq-jaxm`	Message Queue Java API for XML Messaging (JAXM): API	Depends on `sun-saaj`
`sun-mq-jmsclient`	JMS: API and runtime	No dependencies
`sun-mq-xmlclient`	XML Client	Depends on `sun-jmsclient` and `sun-saaj`
`sun-mq-capi`	C-API	Depends on `sun-nspr`, `sun-nss`
`sun-mq-compat`	Symlinks to 3.5 locations.	Depends on `sun-mq` This RPM is not installed by default. **Note:** This package is only needed if your existing clients depend on Message Queue 3.5 file locations.
`sun-mq-sup`	Needed to support JDK 1.3.	Depends on `sun-mq` This RPM is not installed by default. **Note:** This package is only shipped with Message Queue, Platform Edition.
`sun-mq-[locale]`	L10N files	
`sun-javahelp`[1]	JavaHelp: API and runtime	Supports JavaHelp viewer for Admin Console help
`sun-nss`[1]	Network Security Services (NSS) libraries.	Needed to support C language clients
`sun-nspr`[1]	Netscape Portable Runtime (NSPR) libraries	Needed to support C clients
`sun-saaj`[1]	SOAP with Attachments API for Java: API and runtime	Required to support Java clients using SOAP/JAXM API

Table 10 RPM Packages in Linux Bundle *(Continued)*

RPM	Description	Notes
`sun-jaxp`[1]	Message Queue Java API for XML Processing (JAXP): API and runtime	Required to support Java clients using SOAP/JAXM API
`sun-javamail`[1]	JavaMail: API and runtime	Required to support Java clients using SOAP/JAXM API
`sun-jaf`[1]	JavaBeans Activation Framework: API and runtime	Required to support Java clients using SOAP/JAXM API

[1] This RPM package is shared by a number of Sun Java System products. It is supplied with Message Queue, Platform Edition in the distribution, but not supplied with Message Queue, Enterprise Edition.

Symlinks Package for 3.5 Clients The locations of individual files installed as part of Message Queue on Linux have changed. This can break applications that depend on the former location of certain Message Queue files. For instance, if your clients use scripts to locate jar files that were installed with Message Queue 3.5, these clients will no longer locate these files.

To remedy this situation, the Linux package `sun-mq-compat` is included with this release. This package installs symlinks from Message Queue 3.5 file locations to the installed Message Queue 3 2005Q1 (3.6) file locations. This package may not be delivered in future releases. You should fix any scripts that point to former Message Queue file locations.

The `sun-mq-compat` package is in the following locations.

- Message Queue, Platform Edition distribution:
  When you unzip the distribution, an `rpms` directory is created.
- Message Queue, Enterprise Edition (delivered with the Java Enterprise System distribution):
  `Linux_x86/Product/message_queue/Packages`

➤ To install the `sun-mq-compat` package

1. Become Root.
2. From the rpm package directory, use the following command:

```
rpm -ivh --nodeps sun-mq-compat-3.6-<ReleaseNo>.i386.rpm
```

Issues Related to the Next Major Release of Message Queue

The next major release of Message Queue may introduce changes that make your clients incompatible with that release. This information is provided now to allow you to prepare for these changes.

- This is the last feature release of Sun Java System Message Queue that will be backwards compatible with Sun One Message Queue 3.0.1. In particular, future releases of Sun Java System Message Queue will not support the following:
  - Connection of 3.0.1 clients to brokers of the latest version
  - Upgrading a 3.0.1 persistent store to the latest version
  - Clustering of 3.0.1 brokers with brokers of the latest version
  - Use of 3.0.1 property files, user stores, access control lists, etc. with brokers of the latest version.
- This is the last release of Sun Java System Message Queue that will support being integrated as the “System JMS Messaging Provider” for Sun Java System Application Server 7.X. Future releases of Sun Java System Message Queue will only support Sun Java System Application Server 8.0 and above.
- This is the last release of Sun Java System Message Queue that will include the SOAP runtime and support the Message Queue SOAP Administered Objects.
- Future releases will only support SOAP in conjunction with a Java 2 Standard Edition Platform version that supports SOAP.
- Sun Java System Message Queue client support for all releases of Java 2 Standard Edition 1.3 will be dropped. Java 2 Standard Edition 1.4 will continue to be supported.
- The locations of individual files installed as part of Sun Java System Message Queue might change. This could break existing applications that depend on the current location of certain Message Queue files.
- Sun Java System Message Queue clients that use a version of Message Queue older than the next major version might not have access to the new features offered in that version of the product.

Documentation Updates for Message Queue 3 2005Q1 (3.6)

The following Message Queue documents were updated from Version 3.5 of the product:

Installation Guide

The *Message Queue Installation Guide* was updated to reflect branding changes and platform-specific information. This document now contains installation information relevant to Message Queue, Platform Edition.

Message Queue, Enterprise Edition installation information moved to the *Sun Java System Installation Guide*.

For information on upgrade and migration to Message Queue 3 2005Q1 (3.6), Enterprise Edition on Solaris and Linux, see the *Sun Java System Upgrade and Migration Guide*.

For information on upgrade and migration to Message Queue 3 2005Q1 (3.6), Enterprise Edition on Windows, see the Introduction in the *Message Queue Installation Guide*.

Technical Overview

The *Message Queue Technical Overview* is a new document that describes the Message Queue product, its features, architecture, technology, and terminology. This new book contains overview information previously contained in other books and is meant to be used by Message Queue users--both administrators and developers--and prospective users, as well.

Administration Guide

The *Message Queue Administration Guide* has been updated to reflect branding changes and new features. This document has also been reorganized for better usability by Message Queue administrators. Overview information previously contained in this document has been moved to the *Message Queue Technical Overview*.

Java Client Developer's Guide

The *Java Client Developer's Guide* has been updated to reflect branding changes and new features. The document has also been renamed to *Message Queue Developer's Guide for Java Clients*.

The *Message Queue Developer's Guide for Java Clients* has been reorganized for better usability by Message Queue Java Client Developers. Overview information previously contained in this document has been moved to the *Message Queue Technical Overview*.

C Client Developer's Guide

The *C Client Developer's Guide* was updated to reflect branding changes and new features. The document has also been renamed to *Message Queue Developer's Guide for C Clients*.

The *Message Queue Developer's Guide for C Clients* has been reorganized for better usability by Message Queue C Client Developers. Overview information previously contained in this document has been moved to the *Message Queue Technical Overview*.

Known Issues and Limitations

This section contains a list of the known issues with Message Queue 3 2005Q1 (3.6). The following product areas are covered:

- General Issues
- Installation Issues
- Administration/Configuration Issues
- Broker Issues

For a list of current bugs, their status, and workarounds, Java Developer Connection™ members should see the Bug Parade page on the Java Developer Connection web site. Please check that page before you report a new bug. Although all Message Queue bugs are not listed, the page is a good starting place if you want to know whether a problem has been reported.

The relevant page is:

`http://developer.java.sun.com/developer/bugParade`

NOTE Java Developer Connection membership is free but requires registration. Details on how to become a Java Developer Connection member are provided on Sun's "For Developers" web page.

To report a new bug or submit a feature request, send mail to `imq-feedback@sun.com`.

General Issues

This section covers general issues in Message Queue 3 2005Q1 (3.6). Some of these were introduced with previous Message Queue versions. This section groups issues according to whether they apply to both Enterprise and Platform Editions of Message Queue or to the Enterprise Edition only.

Both Enterprise and Platform Editions

- In Message Queue 3 2005Q1, the example broker configuration for using an ldap server as a user repository is provided in the comment area in the `config.properties` file, and the ldap user repository example in the `default.properties` file has been commented out.

  If you previously relied on any property value in the example ldap user repository properties setup in the `default.properties` file, your JMS application client will receive a security exception when attempting to create a JMS connection. This will happen after you upgrade to Message Queue 3 2005Q1.

  When your JMS client tries to make a connection to the Message Queue 3 2005Q1 broker, you will get a error in the broker log and your JMS client will receive the following exception:

  ```
  SecurityException.
  ```

  ```
  20/Aug/2004:11:16:41 PDT] ERROR [B4064]: Ldap repository ldap property .uidattr not
  defined for authentication type basic:com.sun.messaging.jmq.auth.LoginException:
  [B4064]: Ldap repository ldap property .uidattr not defined for authentication type
  basic
  ```

  Workaround
  Set the broker property `imq.user_repository.ldap.uidattr` following the instructions in the *Message Queue Administration Guide*.

- A bug in RedHat Linux 2.1 (NSS bug# 5078380) can cause the Message Queue C-API `MQCreateConnection` (Message Queue 3 2005Q1 C-API library) or `MQInitializeSSL` (Message Queue 3 2005Q1 or 3.5 C-API library) methods to hang if there is any thread created in the Message Queue C-API application before these two methods get called.

  Workaround
  When using the Message Queue 3 2005Q1 C-API library, there are four options for avoiding this issue:

  - Upgrade to RedHat Linux 3.0
  - Set the `MQ_NSS_5078380_WORKAROUND` environment variable before running your Message Queue C client application and set the Message Queue broker property to `imq.authentication.type=basic`
  - Call `MQInistializeSSL(<`*cert-db-path*`>)` before launching any thread in your Message Queue C client application. You can use the `certutil` utility to create the certificate db files
  - Create a connection to the Message Queue broker by calling `MQCreateConnection` before launching any threads

- Windows platforms set limits to the number of connections to a broker that can be simultaneously started over TCP/IP, in accordance with the maximum value of the backlog size. Backlog is the buffer for connections in the TCP stack—the number of simultaneous TCP connection startups cannot exceed the backlog size. For example, Windows 2000 Professional limits the backlog to 5, and Windows 2000 Server limits the backlog to 200.
- If you are running Windows XP, there is a limit to the number of *inbound* connections. For Windows XP Professional, the maximum number of other computers that are permitted to simultaneously connect over the network is ten. This limit includes all transports and resource sharing protocols combined. For Windows XP Home Edition, the maximum number of other computers that are permitted to simultaneously connect over the network is five. This limitation will affect the number of clients that can connect to the broker running Windows XP.

  Any file, print, named pipe, or mail slot session that does not have any activity is automatically disconnected after the `AutoDisconnect` time has expired; the default for the `AutoDisconnect` time is 15 minutes. When the session is disconnected, one of the ten connections becomes available so that another user can connect to the Windows XP system. Therefore, lowering the `AutoDisconnect` time can help reduce some of the problems with the ten-connection limit or the five-connection limit on a system that is not used heavily for server purposes. For more information, see the following:

  http://support.microsoft.com/default.aspx?scid=kb;EN-US;314882
- You cannot edit a broker's instance configuration file without having started the broker instance at least once. This is because the `config.properties` file does not exist until the broker instance is first started. To configure a broker to use pluggable persistence or to set other configuration properties, run the broker once (with the instance name that should be used to create the broker) to create the `config.properties` file:

Platform	**Location**
Solaris	`/var/imq/instances/`*instanceName*`/props/config.properties`
Linux	`/var/opt/sun/mq/instances/`*instanceName*`/props/config.properties`
Windows	`IMQ_VARHOME\instances\`*instanceName*`\props\config.properties`

Once the `config.properties` file has been created, edit the file to add any configuration property values and then restart the broker.

Enterprise Edition Only

- Only fully-connected broker clusters are supported in this release. This means that every broker in a cluster must communicate directly with every other broker in the cluster. If you are connecting brokers using the `imqbrokerd -cluster` command line argument, be careful to ensure that all brokers in the cluster are included.
- A client connected to a broker that is part of a cluster cannot currently use `QueueBrowser` to browse queues that are located on remote brokers in that cluster. The client can only browse the contents of queues that are located on the broker to which it is directly connected. The client may still send messages to any queue or consume messages from any queue on any broker in the cluster; the limitation only affects browsing.
- If a Master Broker is not used in a broker cluster, persistent information stored by a broker being added to the cluster is not propagated to other brokers in the cluster.
- A connection service using SSL is currently limited to supporting only self-signed server certificates, that is, host-trusted mode.
- When a JMS client using the HTTP transport terminates abruptly (for example, using `Ctrl-C`) the broker takes approximately one minute before the client connection and all the associated resources are released.

  If another instance of the client is started within the one minute period and if it tries to use the same ClientID, durable subscription, or queue, it might receive a "Client ID is already in use" exception. This is not a real problem; it is just the side effect of the termination process described above. If the client is started after a delay of approximately one minute, everything should work fine.

Installation Issues

On Windows, the Add/Remove Programs control panel displays Sun Java Enterprise System, instead of Sun Java System Message Queue.

Workaround:
Select Sun Java Enterprise System to remove Sun Java System Message Queue.

On Windows, the location of NSS and NSPR libraries has changed (#*6271133*).

The location of the NSS and NSPR libraries has changed from previous releases. The *Message Queue Developer's Guide for C Clients* may describe these locations incorrectly. The NSS and NSPR libraries are required to build C clients. The new location of these libraries is in the `share\lib` folder. Example: `C:\Sun\share\lib`

On Windows, a confusing error message appears when attempting to install on a disk without sufficient disk space.

When installing on a disk that does not have sufficient disk space, the installation may fail with a confusing error message. Example: "`Error: -1603 Fatal error during installation.`"

Workaround:
If you receive this error message, attempt the installation on a disk with sufficient space.

On Windows, the `Config_MQ.txt` file contains incorrect configuration information about installer options. (*#6292300*)

The *INSTALLDIR*`\MessageQueue\Config_MQ.txt` file incorrectly mentions two options:

- Quick Configuration
- Config Later

Instead, the options should be:

- Configure Automatically
- Configure Manually

If you select the Configure Automatically option, the Message Queue Broker service will be installed and started automatically by the installer.

If you select the Configure Manually option, the Message Queue Broker service will be installed, but the user has to start the service manually.

There are two ways to start the service manually:

- Click on the shortcut to start the Message Queue Broker.
- Go to the Windows Services Control Manager and start the service.

Administration/Configuration Issues

The imqadmin and imqobjmgr utilities throw an error when the CLASSPATH contains double quotes on Windows machines (*#5060769*)

Workaround
You can ignore this error message; the broker correctly handles notifying consumers of any error. This error does not affect the reliability of the system.

The -javahome option in all solaris/win scripts does not work if the value has a space (#*4683029*)

The -javahome option is used by the Message Queue commands and utilities to specify an alternate Java 2 compatible runtime to use. However, the path to the alternate Java runtime must be located at a path that does not contain spaces.

Examples of paths that have spaces are:

Windows:
```
C:\jdk 1.4
```

Solaris:
```
/work/java 1.4
```

Workaround
Install the Java runtime at a location or path that does not contain spaces.

Message Queue output garbage message to syslog in non-C locale *(#6193884)*

Message Queue will send garbage output to the system log when the broker is run in a non-C locale.

Workaround
Run broker in C locale.

Broker Issues

In a broker cluster, a broker will queue messages to a remote connection which may not be started (#*4951010*)

Workaround
The messages will be received by the consumer once the connection is started. The messages will be redelivered to another consumer if the consumer's connection is closed.

HTTPS createQueueConnection occasionally throws exception on Windows 2000. (#*4953348*)

Workaround
Retry the connection.

When using Ctrl-C to shut down broker, transactions may be cleaned up after store is closed (#*4934446*)

The broker may show errors with the following reason “Store method accessed after the store is closed.” if the broker is shutdown while messages or transactions are processed.

Workaround
You can ignore this error message; the broker correctly handles notifying consumers of any error. This error does not affect the reliability of the system.

Broker becomes inaccessible when persistent store opens too many destinations. (#*4953354*)

Workaround
This condition is caused by the broker reaching the system open-file descriptor limit. On Solaris and Linux use the ulimit command to increase the file descriptor limit.

Unknown Destination:temporary_destination sometimes seen at shutdown (#*5055667*)

When a broker is shutdown while processing messages to a temporary queue, the following error message may show up in the logs:

```
WARNING [B2011]: Storing of JMS message from <connection> failed:
com.sun.messaging.jmq.jmsserver.util.BrokerException: Unknown
Destination:temporary_destination:<destinationid>
```

Workaround
You can ignore this error message. This error does affect the reliability of the broker.

Consumers are orphaned when a destination is destroyed (#*5060787*)

Active consumers are orphaned when a destination is destroyed. Once the consumers have been orphaned, they will no longer receive messages (even if the destination is recreated).

Workaround
There is no workaround for this problem.

New consumer may not receive messages in a broker cluster with autocreated destinations (#*6172794*)

If a remote consumer is added to an autocreated destination at the same time the destination is being destroyed, the consumer may be incorrectly added to the wrong destination. This will cause that consumer to not receive messages published to the destination.

Workaround
Use one of the three following options:

- increase the broker property `imq.autocreate.reaptime`
- use admin created destinations
- close and re-open the consumer

Producer flow control may hang when producers are constantly added to full destination (*#6173336*)

The hang occurs when:

- limit of a destination is set to a low number (<= 2 * max # producers)
- destination is set to `flow_control`
- producers are frequently added to the destination (churning)
  In this case, in a loop, each producer sent a message and then closed.
- these churning producers are being added to a full producer

Workaround
Do the following:

- increase the size of the destination
- decrease the maximum number of consumers
- change code to not churn producers on a connection

Message selection using JMSMessageID doesn't work *(#6196233)*

Message selection using the selector "JMSMessageID = '<message_id>'" does not work.

1. Send a message to a Queue
2. Read the `JMSMessageID` of the sent message - `<message_id>`
3. Create a consumer on the Queue using the selector set to `"JMSMessageID = '<message_id>'"`

   Message will not be received.

Workaround
Change the selector from:

```
JMSMessageID = "ID:message-id-string"
```

to:

```
JMSMessageID IN ('ID:message-id-string', 'message-id-string')
```

Redistributable Files

Sun Java System Message Queue 3 2005Q1 (3.6) contains the following set of files which you may use and freely distribute in binary form:

- `jms.jar`
- `imq.jar`
- `imqxm.jar`
- `fscontext.jar`
- `providerutil.jar`
- `jndi.jar`
- `ldap.jar`
- `ldapbpjar`
- `jaas.jar`
- `jsse.jar`
- `jnet.jar`
- `jcert.jar`
- In addition, you can also redistribute the `LICENSE` and `COPYRIGHT` files.

How to Report Problems and Provide Feedback

If you have problems with Sun Java System Message Queue, contact Sun customer support using one of the following mechanisms:

- Sun Software Support services online at `http://www.sun.com/service/sunone/software`

  This site has links to the Knowledge Base, Online Support Center, and ProductTracker, as well as to maintenance programs and support contact numbers.
- The telephone dispatch number associated with your maintenance contract

So that we can best assist you in resolving problems, please have the following information available when you contact support:

- Description of the problem, including the situation where the problem occurs and its impact on your operation
- Machine type, operating system version, and product version, including any patches and other software that might be affecting the problem
- Detailed steps on the methods you have used to reproduce the problem
- Any error logs or core dumps

Sun Java System Software Forum

There is a Sun Java System Message Queue forum available at the following location:

`http://swforum.sun.com/jive/forum.jspa?forumID=24`

We welcome your participation.

Java Technology Forum

There is a JMS forum in the Java Technology Forums that might be of interest.

`http://forum.java.sun.com`

Sun Welcomes Your Comments

Sun is interested in improving its documentation and welcomes your comments and suggestions.

To share your comments, go to `http://docs.sun.com` and click Send Comments. In the online form, provide the document title and part number. The part number is a seven-digit or nine-digit number that can be found on the title page of the book or at the top of the document. For example, the title of this book is *Sun Java System Message Queue 3 2005Q1 Release Notes*, and the part number is 819-0064-12.

Additional Sun Resources

Useful Sun Java System information can be found at the following Internet locations:

- Documentation for Message Queue
  `http://docs.sun.com/coll/MessageQueue_05q1`
- Sun Java System Documentation
  `http://docs.sun.com/prod/java.sys`
- Sun Java System Professional Services
  `http://www.sun.com/service/sunps/sunone`
- Sun Java System Software Products and Service
  `http://www.sun.com/software`
- Sun Java System Software Support Services
  `http://www.sun.com/service/sunone/software`
- Sun Java System Support and Knowledge Base
  `http://www.sun.com/service/support/software`
- Sun Support and Training Services
  `http://training.sun.com`
- Sun Java System Consulting and Professional Services
  `http://www.sun.com/service/sunps/sunone`
- Sun Java System Developer Information
  `http://developers.sun.com`
- Sun Developer Support Services
  `http://www.sun.com/developers/support`
- Sun Java System Software Training
  `http://www.sun.com/software/training`
- Sun Software Data Sheets
  `http://wwws.sun.com/software`