Installation Guide

Sun™ ONE Message Queue

Version 3.5

817-3725-10
November 2003

Copyright © 2003 Sun Microsystems, Inc., 4150 Network Circle, Santa Clara, California 95054, U.S.A. All rights reserved.

Sun Microsystems, Inc. has intellectual property rights relating to technology embodied in the product that is described in this document. In particular, and without limitation, these intellectual property rights may include one or more of the U.S. patents listed at http://www.sun.com/patents and one or more additional patents or pending patent applications in the U.S. and in other countries.

U.S. Government Rights - Commercial software. Government users are subject to the Sun Microsystems, Inc. standard license agreement and applicable provisions of the FAR and its supplements. Use is subject to license terms. This distribution may include materials developed by third parties.

Sun, Sun Microsystems, the Sun logo, Java, Solaris, Sun[tm] ONE, JDK, Java Naming and Directory Interface, Javadoc, JavaMail, JavaHelp, the Java Coffee Cup logo and the Sun[tm] ONE logo are trademarks or registered trademarks of Sun Microsystems, Inc. in the U.S. and other countries.

All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. in the U.S. and other countries. Products bearing SPARC trademarks are based upon architecture developed by Sun Microsystems, Inc.

UNIX is a registered trademark in the U.S. and other countries, exclusively licensed through X/Open Company, Ltd.

This product is covered and controlled by U.S. Export Control laws and may be subject to the export or import laws in other countries. Nuclear, missile, chemical biological weapons or nuclear maritime end uses or end users, whether direct or indirect, are strictly prohibited. Export or reexport to countries subject to U.S. embargo or to entities identified on U.S. export exclusion lists, including, but not limited to, the denied persons and specially designated nationals lists is strictly prohibited.

Copyright © 2003 Sun Microsystems, Inc., 4150 Network Circle, Santa Clara, California 95054, Etats-Unis. Tous droits réservés.

Sun Microsystems, Inc. détient les droits de propriété intellectuels relatifs à la technologie incorporée dans le produit qui est décrit dans ce document. En particulier, et ce sans limitation, ces droits de propriété intellectuelle peuvent inclure un ou plus des brevets américains listés à l'adresse http://www.sun.com/patents et un ou les brevets supplémentaires ou les applications de brevet en attente aux Etats - Unis et dans les autres pays.

L'utilisation est soumise aux termes de la Licence.

Cette distribution peut comprendre des composants développés par des tierces parties.

Sun, Sun Microsystems, le logo Sun, Java, Solaris, Sun[tm] ONE, JDK, Java Naming and Directory Interface, Javadoc, JavaMail, JavaHelp, le logo Java Coffee Cup et le logo Sun[tm] ONE sont des marques de fabrique ou des marques déposées de Sun Microsystems, Inc. aux Etats-Unis et dans d'autres pays.

Toutes les mar ques SPARC sont utilisées sous licence et sont des marques de fabrique ou des marques déposées de SPARC International, Inc. aux Etats-Unis et dans d'autres pays. Les produits portant les marques SPARC sont basés sur une architecture développée par Sun Microsystems, Inc.

UNIX est une marque déposée aux Etats-Unis et dans d'autres pays et licenciée exlusivement par X/Open Company, Ltd.

Ce produit est soumis à la législation américaine en matière de contrôle des exportations et peut être soumis à la règlementation en vigueur dans d'autres pays dans le domaine des exportations et importations. Les utilisations, ou utilisateurs finaux, pour des armes nucléaires,des missiles, des armes biologiques et chimiques ou du nucléaire maritime, directement ou indirectement, sont strictement interdites. Les exportations ou réexportations vers les pays sous embargo américain, ou vers des entités figurant sur les listes d'exclusion d'exportation américaines, y compris, mais de manière non exhaustive, la liste de personnes qui fo nt objet d'un ordre de ne pas participer, d'une façon directe ou indirecte, aux exportations des produits ou des services qui sont régis par la législation américaine en matière de contrôle des exportations et la liste de ressortissants spécifiquement désignés, sont rigoureusement interdites.

Contents

List of Tables 7

List of Procedures 9

Preface 11
Audience for This Guide 11
Organization of This Guide 11
Conventions 12
Text Conventions 12
Directory Variable Conventions 13
Other Documentation Resources 15
The MQ Documentation Set 15
Example Client Applications 16
Example Java Client Applications 16
Example C Client Applications 16

Chapter 1 Introduction 17
Product Editions 17
Platform Edition 18
Enterprise Edition 18
Supported Platforms and Products 19
MQ Software Modules 21
Installing from Web and CD-ROM 21
Upgrade Issues from MQ 3.0.x Versions 21
Platform Issues 22
Solaris 22
Linux 22
Windows 22

Compatibility Issues 23
Broker Compatibility 23
Administered Object Compatibility 25
Administration Tool Compatibility 26
Client Compatibility 26
Where To Go Next 27

Chapter 2 Solaris Installation 29
Hardware and Software Requirements 29
Installing MQ on Solaris 30
Installed Directory Structure 31
Installing from the Web 33
To Install MQ on Solaris from the Web 33
Installing from CD-ROM 39
To Install MQ on Solaris from CD-ROM 39
Configuring MQ for Automatic Startup 41
Configuring the Java Runtime for MQ 42
Upgrading from Platform to Enterprise Edition 42
To Upgrade to Enterprise Edition on Solaris 43
Where To Go Next 44
Uninstalling MQ on Solaris 45
To Remove MQ on Solaris 45

Chapter 3 Linux Installation 47
Hardware and Software Requirements 47
Installing MQ on Linux 48
Finding and Removing Earlier MQ Versions 48
Migrating Files from Non-Standard Locations 49
Finding and Removing MQ RPM's (Version 3.0.1 SP1 or Later Only) 49
To Find and Remove Earlier RPM-Installed Versions of MQ 49
Finding and Removing an MQ Tar-Based Installation 50
To Find and Remove Earlier Tar-Based Installed MQ 50
Installed Directory Structure 50
Installing from the Web 52
To Install MQ on Linux from the Web 52
Installing from CD-ROM 54
To Install MQ on Linux from CD-ROM 54
Configuring the Java Runtime for MQ 54
Upgrading from Platform to Enterprise Edition 55
To Upgrade to Enterprise Edition on Linux 55

Where To Go Next 55
Uninstalling MQ on Linux 56
To Remove MQ on Linux 56

Chapter 4 Windows Installation 57
Hardware and Software Requirements 57
Installing MQ on Windows 58
Installed Directory Structure 59
Installing From CD-ROM or the Web 60
To Install MQ on Windows 60
Installation Defaults 62
Troubleshooting Installation Problems 62
To Troubleshoot Installation Problems on Windows 62
Upgrading from Platform to Enterprise Edition 63
To Upgrade to Enterprise Edition on Windows 63
Where To Go Next 64
Uninstalling MQ on Windows 64
To Remove MQ on Windows 64

List of Tables

Table 1 Book Contents 11
Table 2 Document Conventions 12
Table 3 MQ Directory Variables 13
Table 4 MQ Documentation Set 15
Table 1-1 MQ 3.5 Basic Product Support Matrix 19
Table 1-2 MQ 3.5 Optional Product Support Matrix 20
Table 1-3 MQ Software Modules 21
Table 1-4 Compatibility of MQ 3.5 with MQ 3.0 Data 25
Table 2-1 Hardware and Software Requirements for Solaris 29
Table 2-2 Installed Directory Structure: Solaris 31
Table 2-3 MQ 3.5 Solaris Distribution Files 33
Table 2-4 Packages in Solaris Bundle 34
Table 2-5 Packages Required for Various Scenarios 35
Table 2-6 Broker Startup Configuration Properties 41
Table 2-7 Upgrade Packages in Solaris Enterprise Bundle 43
Table 3-1 Hardware and Software Requirements for Linux 47
Table 3-2 Installed Directory Structure: Linux 51
Table 4-1 Hardware and Software Requirements for Windows 57
Table 4-2 Installed Directory Structure: Windows 59
Table 4-3 Windows Installation Defaults 62

List of Procedures

To Install MQ on Solaris from the Web 33
To Install MQ on Solaris from CD-ROM 39
To Upgrade to Enterprise Edition on Solaris 43
To Remove MQ on Solaris 45
To Find and Remove Earlier RPM-Installed Versions of MQ 49
To Find and Remove Earlier Tar-Based Installed MQ 50
To Install MQ on Linux from the Web 52
To Install MQ on Linux from CD-ROM 54
To Upgrade to Enterprise Edition on Linux 55
To Remove MQ on Linux 56
To Install MQ on Windows 60
To Troubleshoot Installation Problems on Windows 62
To Upgrade to Enterprise Edition on Windows 63
To Remove MQ on Windows 64

Preface

This book explains how to install Sun™ Open Net Environment (Sun ONE) Message Queue (MQ) 3.5, and this preface contains the following sections:

- Audience for This Guide
- Organization of This Guide
- Conventions
- Other Documentation Resources

Audience for This Guide

The audience for this guide is Sun ONE Message Queue developers and administrators.

Organization of This Guide

All users should read the introduction and then the appropriate platform-specific chapter. The following table briefly describes the contents of each chapter:

Table 1 Book Contents

Chapter	Description
Chapter 1, “Introduction”	Explains the MQ editions, supported platforms and products, the MQ software modules, and the installed directory structure.
Chapter 2, “Solaris Installation”	Explains the hardware and software requirements, defaults, and installation procedure for Solaris, as well as how to uninstall the product on this platform.

Table 1 Book Contents *(Continued)*

Chapter	Description
Chapter 3, "Linux Installation"	Explains the hardware and software requirements, defaults, and installation procedure for Linux, as well as how to uninstall the product on this platform.
Chapter 4, "Windows Installation"	Explains the hardware and software requirements, defaults, and installation procedure for Windows, as well as how to uninstall the product on this platform.

Conventions

This section provides information about the conventions used in this document.

Text Conventions

Table 2 Document Conventions

Format	Description
italics	Italicized text represents a placeholder. Substitute an appropriate clause or value where you see italic text. Italicized text is also used to designate a document title, for emphasis, or for a word or phrase being introduced.
`monospace`	Monospace text represents example code, commands that you enter on the command line, directory, file, or path names, error message text, class names, method names (including all elements in the signature), package names, reserved words, and URLs.
`[]`	Square brackets to indicate optional values in a command line syntax statement.
ALL CAPS	Text in all capitals represents file system types (GIF, TXT, HTML and so forth), environment variables (IMQ_HOME), or acronyms (MQ, JSP).
Key+Key	Simultaneous keystrokes are joined with a plus sign: Ctrl+A means press both keys simultaneously.
Key-Key	Consecutive keystrokes are joined with a hyphen: Esc-S means press the Esc key, release it, then press the S key.

Directory Variable Conventions

MQ makes use of three directory variables; how they are set varies from platform to platform. Table 3 describes these variables and summarizes how they are used on the Solaris, Windows, and Linux platforms.

Table 3 MQ Directory Variables

Variable	Description
IMQ_HOME	This is generally used in MQ documentation to refer to the MQ base directory (root installation directory): • On Solaris, there is no root MQ installation directory. Therefore, IMQ_HOME is not used in MQ documentation to refer to file locations on Solaris. • On Solaris, for Sun ONE Application Server, Evaluation Edition, the root MQ installation directory is: *Application Server base directory*/imq. • On Windows, the root MQ installation directory is set by the MQ installer (by default, as C:\Program Files\Sun\MessageQueue3). • On Windows, for Sun ONE Application Server, the root MQ installation directory is: *Application Server base directory*/imq. • On Linux, there is no root MQ installation directory. Therefore, IMQ_HOME is not used in MQ documentation to refer to file locations on Linux.
IMQ_VARHOME	This is the /var directory in which MQ temporary or dynamically-created configuration and data files are stored. It can be set as an environment variable to point to any directory. • On Solaris, IMQ_VARHOME defaults to the /var/imq directory. • On Solaris, for Sun ONE Application Server, Evaluation Edition, IMQ_VARHOME defaults to the IMQ_HOME/var directory. • On Windows IMQ_VARHOME defaults to the IMQ_HOME\var directory. • On Windows, for Sun ONE Application Server, IMQ_VARHOME defaults to the IMQ_HOME\var directory. • On Linux, IMQ_VARHOME defaults to the /var/opt/imq directory

Table 3 MQ Directory Variables *(Continued)*

Variable	Description
IMQ_JAVAHOME	This is an environment variable that points to the location of the Java runtime (JRE) required by MQ executables: • On Solaris, IMQ_JAVAHOME defaults to the /usr/j2se/jre directory, but a user can optionally set the value to wherever the required JRE resides. • On Windows, IMQ_JAVAHOME defaults to IMQ_HOME\jre, but a user can optionally set the value to wherever the required JRE resides. • On Linux, MQ first looks for the java runtime in the /usr/java/j2sdk*Version* directory, and then looks in the /usr/java/j2re*Version* directory, but a user can optionally set the value of IMQ_JAVAHOME to wherever the required JRE resides.

In this guide, IMQ_HOME, IMQ_VARHOME, and IMQ_JAVAHOME are shown *without* platform-specific environment variable notation or syntax (for example, $IMQ_HOME on UNIX). Path names generally use UNIX directory separator notation (/).

Other Documentation Resources

In addition to this guide, MQ provides additional documentation resources.

The MQ Documentation Set

The documents that comprise the MQ documentation set are listed in Table 4 in the order in which you would normally use them.

Table 4 MQ Documentation Set

Document	Audience	Description
MQ *Installation Guide*	Developers and administrators	Explains how to install MQ software on Solaris, Linux, and Windows platforms.
MQ *Release Notes*	Developers and administrators	Includes descriptions of new features, limitations, and known bugs, as well as technical notes.
MQ *Java Client Developer's Guide*	Developers	Provides a quick-start tutorial and programming information for developers of Java client programs using the MQ implementation of the JMS or SOAP/JAXM APIs.
MQ *Administrator's Guide*	Administrators, also recommended for developers	Provides background and information needed to perform administration tasks using MQ administration tools.
MQ *C Client Developer's Guide*	Developers	Provides programming and reference documentation for developers of C client programs using the MQ C implementation of the JMS API. (C-API).

Example Client Applications

MQ provides a number of example client applications to assist developers.

Example Java Client Applications

Example Java client applications are located in the following directories, depending on platform. See the `README` file located in these directories and in each of their subdirectories.

Platform	Location
Solaris	`/usr/demo/imq/`
Linux	`/opt/imq/demo/`
Windows	`IMQ_HOME\demo\`

Example C Client Applications

Example C client applications are located in the following directories, depending on platform. See the `README` file located in these directories.

Platform	Location
Solaris	`/opt/SUNWimq/demo/C/`
Linux	`/opt/imq/demo/C/`
Windows	`IMQ_HOME\demo\C\`

Chapter 1

Introduction

This chapter provides an overall introduction to installing the MQ product. The topics covered are:

- Product Editions
- Supported Platforms and Products
- MQ Software Modules
- Installing from Web and CD-ROM
- Upgrade Issues from MQ 3.0.x Versions
- Upgrade Issues from MQ 3.0.x Versions
- Where To Go Next

Product Editions

The Sun ONE Message Queue product is available in two editions: Platform and Enterprise—each containing different features and corresponding to a different licensed capacity, as described below. (To upgrade MQ from one edition to another, see "Upgrade Issues from MQ 3.0.x Versions" on page 21 and the relevant instructions for each platform in this manual.)

Platform Edition

This edition can be downloaded free from the Sun website and is also bundled with the Sun ONE Application Server product. The Platform Edition places no limit on the number of JMS client connections supported by each MQ message server. It comes with two licenses, as described below:

- **a basic license.** This license provides basic JMS support (it's a fully compliant JMS provider), but does *not* include such enterprise features as load balancing (multi-broker message service), HTTP/HTTPS connections, secure connection services, scalable connection capability, queue delivery to more than three consumers, and C client support. The license has an unlimited duration, and can therefore be used in less demanding production environments.
- **a 90-day trial enterprise license.** This license includes all enterprise features (such as support for multi-broker message services, HTTP/HTTPS connections, secure connection services, scalable connection capability, queue delivery to more than three consumers, and C- client support) *not* included in the basic license. However, the license has a limited 90-day duration enforced by the software, making it suitable for evaluating the enterprise features available in the Enterprise Edition of the product (see "Enterprise Edition").

NOTE The 90-day trial license can be enabled by starting the MQ message service—an MQ broker instance—with the `-license` command line option (see the MQ *Administrator's Guide*) and passing "`try`" as the license to use:

```
imqbrokerd -license try
```

You must use this option each time you start the broker instance, otherwise it defaults back to the basic Platform Edition license.

Enterprise Edition

This edition is for deploying and running messaging applications in a production environment. It includes support for multi-broker message services, HTTP/HTTPS connections, secure connection services, scalable connection capability, client connection failover, queue delivery to more than three consumers, and C client support. You can also use the Enterprise Edition for developing, debugging, and load testing messaging applications and components. The Enterprise Edition has an unlimited duration license that places no limit on the number of brokers in a multi-broker message service. Licenses are purchased based on the number of CPU's.

Supported Platforms and Products

MQ 3.5 is supported on Solaris, Linux, and Windows operating systems and platforms. It also depends upon other technologies, as indicated in the following table. Other versions or vendor implementations can also be used but they are untested by Sun Microsystems and therefore not supported.

Table 1-1 lists and describes the basic components that you must install in order to be able to develop and run MQ clients.

Table 1-1 MQ 3.5 Basic Product Support Matrix

Platform/Product	Used For	Supported Platform/Product Version[1]
Java Runtime Environment (JRE) (Sun Microsystems versions only)	MQ broker (message server) and MQ administration tools	JRE 1.4.2_02: • Solaris 8 (SPARC), Solaris 9 (SPARC and X86) Platform-bundled JRE might need to be upgraded. • Linux Red Hat Advanced Server 2.1 Update 2 JRE 1.4.2 is bundled on MQ CD distribution, but not yet with Linux platform. • Windows XP Professional, 2000 Professional SP3, 2000 Advanced Server SP3, .NET Server 2003 Enterprise Edition JRE 1.4.2 is bundled and installed with MQ.
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 1.4.0, 1.4.1, and 1.4.2)	JDK 1.4.0, 1.4.1, and 1.4.2_02: • Solaris (same platforms as JRE) • Linux (same platforms as JRE) • Windows (same platforms as JRE) Version 1.3.1[2]: • (SPARC only) Solaris 8 and Solaris 9 • Windows XP professional, 2000 Professional SP3, 2000 Advanced Server SP3 Version 1.2.2: Not supported, but should work (in case you cannot upgrade to a later version)

1. Check the MQ *Release Notes* for any updates to supported versions
2. Download this JDK from: http://java.sun.com/j2se/1.4

Table 1-2 lists and describes the components that you can install in order to provide additional support for an MQ 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 1-2 MQ 3.5 Optional Product Support Matrix

Product	Used For	Supported Product Version
LDAP Directory Server	MQ user repository and administered object support	Sun ONE Directory Server Version 5.2.1
Web Server	HTTP and HTTPS support	Sun ONE Web Server, Enterprise Edition 6.1
Application Server	HTTP and HTTPS support	Sun ONE Application Server 7, Update 2 on Solaris 9 (Note: MQ is the JMS provider for Application Server 7)
Database	Plugged-in persistence support	PointBase, Version 4.5 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	MQ 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)	MQ C client support	• Solaris (SPARC) 4.1.2 • Solaris (X86): 4.1.3 • Linux: 4.1.5 • Windows: 4.1.5
NSS (Network Security Service)	MQ C client support	• Solaris (SPARC): 3.3.2 • Solaris (X86): 3.3.3 • Linux: 3.3.5 • Windows: 3.3.5

MQ Software Modules

The following table identifies the full set of software modules included with the MQ product (see Table 1-3 for their installed location).

Table 1-3 MQ Software Modules

Module	Contents
Broker	The server-side software used to route and deliver messages. This module requires the Java runtime module.
Administration tools	The command-line utilities and GUI tools used to administer an MQ messaging system. This module requires the client runtime and Java runtime modules.
Java client runtime	Libraries and header files needed to write and compile Java clients using the MQ Java API.
C client runtime	Libraries and header files needed to write and compile C clients using the MQ C API (Enterprise Edition only).
Documentation	API documentation needed by Java client application developers in JavaDoc format.
Example applications	Example client applications.
Java runtime	Java Runtime Environment (Windows only).
License	The software needed to enable the licensed capacity of an MQ messaging system.

Installing from Web and CD-ROM

You have the option of either downloading MQ 3.5 from the product website:

```
http://www.sun.com/software/message_queue/
```

or installing it from CD-ROM. For detailed instructions, see the platform-specific instructions in subsequent chapters. The installed directory structure is described for each platform in the chapter describing that platform.

Upgrade Issues from MQ 3.0.x Versions

This section covers issues you need to be aware of when upgrading to MQ 3.5 from MQ 3.0.x versions. These issues fall into two general categories: platform issues and compatibility issues.

Platform Issues

This section describes issues and precautions to take when upgrading to MQ 3.5 on the three supported platforms: Solaris, Linux, and Windows.

Solaris

On the Solaris platform, you can install MQ 3.5 on top of MQ 3.0.x versions, and your previous instance data (configuration properties, flat-file persistent store, log files, flat-file user repository, and/or access control properties file) will be used by MQ 3.5 (see "Compatibility Issues" on page 23).

If you depended on jar files being in their 3.0.x locations, please note that these have been moved to the `/usr/share/lib` directory. This applies to the following `.jar` files: `jms.jar`, `imq.jar`, `imqxm.jar`, `activation.jar`, `saaj-api.jar`, `saaj-impl.jar`, `mail.jar`, `commons-logging.jar`, `jaxm-api.jar`, `fscontext.jar`.

- In MQ 3.0, these files were in the `/usr/share/lib/imq directory`
- In MQ 3.0.1, these files were in the `/usr/share/lib directory`, with sym links in the `/usr/share/lib/imq` directory

In MQ 3.5 there are no sym links files.

Linux

On the Linux platform, you should un-install any MQ 3.0 versions before installing MQ 3.5, and *not* try to install MQ 3.5 on top of them. This is due to the fact that the installed directory structure has been changed for MQ 3.5, and this complicates the migration of instance data (configuration properties, flat-file persistent store, log files, flat-file user repository, and/or access control properties file) from MQ 3.0 to MQ 3.5. A new utility, `mqmigrate`, is provided to ease the migration of this data (see "Finding and Removing Earlier MQ Versions" on page 48).

Users need to uninstall the 3.0.1 RPM with `-e` (erase) and then to install the 3.5 RPM using the `-i` (install) option as described in this guide.

Windows

On the Windows platform, if you are upgrading from a Message Queue 3.0 version product to Message Queue 3.5, it is likely that you will have to manually migrate any previous instance data (configuration properties, flat-file persistent store, log files, flat-file user repository, and/or access control properties file) from the Message Queue 3.0 location to the Message Queue 3.5 location.

This is because the default Message Queue installation directory (represented by the IMQ_HOME directory variable) has been changed on the Windows platform. The same issue arises if you installed Message Queue 3.0 in a non-default location and are installing Message Queue 3.5 in a different location.

Instance data is stored in the following directory:

IMQ_VARHOME\instances*instanceName*

where IMQ_VARHOME defaults to IMQ_HOME\var

When you install Message Queue 3.5 in a location different from Message Queue 3.0, you have to move any instance data you wish to preserve from the old location, corresponding to the old value of IMQ_VARHOME, to the new location, corresponding to the new value of IMQ_VARHOME.

Compatibility Issues

MQ 3.5 is generally compatible with MQ 3.0 (and subsequent versions 3.0.1, 3.0.1 SP1, and 3.0.1 SP2). However, changes have been made in broker properties, administered objects, persistence schema, file locations, and administration tools that can impact an upgrade from MQ 3.0 versions to MQ 3.5.

The MQ 3.5 install operation does not remove or over-write the MQ 3.0.x IMQ_VARHOME directory. This directory contains configuration and security-related files (see Table 1-4 on page 25). Most of this data is compatible with MQ 3.5, and can be preserved using the instructions in the following sections.

The issues that you might need to address when upgrading from MQ 3.0 to MQ 3.5 include the following:

- Broker Compatibility
- Administered Object Compatibility
- Administration Tool Compatibility
- Client Compatibility

Broker Compatibility

An MQ 3.5 broker will inter-operate with an MQ 3.0 broker, however changes have been made in broker properties and the persistent store schema. Some MQ 3.0 data is compatible with MQ 3.5, as shown in Table 1-4 on page 25, and can be used after upgrading to MQ 3.5.

When upgrading from MQ 3.0.x to MQ 3.5, you should consider the following:

- You can use MQ 3.0.x `config.properties` files, or you can copy them to another location and consult the property settings they contain when you configure MQ 3.5 brokers.
- Any persistent MQ 3.0.x data—messages, destinations, durable subscriptions—is automatically converted to MQ 3.5 data when starting up an MQ 3.5 broker for the first time. For example, any existing MQ 3.0.x destinations will be converted to MQ 3.5 destinations, preserving existing attributes and using default values of new attributes.

  The automatic migration of persistent data leaves the MQ 3.0.x data intact, unless you specify the following option when starting up the MQ 3.5 broker for the first time: `imqbrokerd -upgrade-store-nobackup`.

  If you do *not* use this option, you should delete the old persistent store manually:

  - For a built-in (flat file) data store, delete the old persistent store, located at

    …`/instances/`*instanceName*`/filestore/`

  - For a plugged-in (JDBC-compliant) data store, delete the old tables using the following command:

    ```
    imqdbmgr delete oldtbl
    ```

  NOTE When you migrate persistent data from MQ 3.0.x to MQ 3.5, the migrated data can no longer be used by an MQ 3.0.x broker. If this is a concern, save the old persistent data to a secure location instead of deleting it as directed above.

- You can continue to use the MQ 3.0.x user repository and access control properties files after installing MQ 3.5, however these files have been made instance-specific and are now placed in the …`/instances/`*instanceName*`/etc` directories. The MQ 3.5 installer does not overwrite the MQ 3.0.x files. When you first start up an MQ 3.5 broker, copies of the MQ 3.0.x files are placed in the appropriate MQ 3.5 location (see the MQ *Administrator's Guide*, Appendix A). If MQ 3.0.x user repository and access control properties files are not found at the old location, new files will be created in the …`/instances/`*instanceName*`/etc` directory.
- If you mix MQ 3.0.x brokers and MQ 3.5 brokers in a cluster, the master broker must be a MQ 3.0.x broker, and the cluster will run as an MQ 3.0.x. cluster.

Table 1-4 Compatibility of MQ 3.5 with MQ 3.0 Data

MQ 3.0.x Data Category	Location of MQ 3.0.x Data	Compatibility with MQ 3.5
Broker properties	`IMQ_VARHOME/instances/`*instanceName*`/props/config.properties`	New properties have been added, and some property names have changed. Old property names are still recognized.
Persistent store: messages, destinations, durable subscriptions	`IMQ_VARHOME/instances/`*instanceName*`/filestore/` or JDBC-accessible data store	Converted to MQ 3.5 format when MQ 3.5 broker is started for the first time. The persistent store is at `IMQ_VARHOME/`*instanceName*`/fs350`
Security: flat-file user repositories	`/etc/imq/passwd` (Solaris) `IMQ_HOME/etc/passwd` (other platforms)	Compatible. File automatically copied to following location at first broker startup: `…/instances/`*instanceName*`/etc/passwd`
Security: access control file	`/etc/imq/accesscontrol.properties` (Solaris) `IMQ_HOME/etc/accesscontrol.properties` (other platforms)	Compatible. File automatically copied to following location at first broker startup: `…/instances/`*instanceName*`/etc/accesscontrol.properties`

Administered Object Compatibility

MQ 3.5 administered objects have been enhanced with new attributes and some MQ 3.0.x attributes have been renamed. Therefore, when upgrading from MQ 3.0 to MQ 3.5, you should consider the following:

- You can use the same object store and administered objects that you created in MQ 3.0.x; however, it is best to upgrade your administered objects after installing MQ 3.5. The Administration Console (`imqadmin`) and the ObjectManager command line utility (`imqobjmgr`), when performing an update operation, will convert MQ 3.0.x administered objects into MQ 3.5 administered objects.
- The MQ 3.5 client runtime will look up and instantiate MQ 3.0.x administered objects and convert them for use by MQ 3.5 clients. However, this will *not* convert MQ 3.0.x administered objects residing in the object store from which the lookup was made.

- Existing MQ 3.0 clients (applications and/or components)—that is, clients that directly instantiate administered objects—are compatible with MQ 3.5. However, if they are to use the *new* administered object attributes (see Chapter 4 and Appendix A of the MQ *Java Client Developer's Guide* for information on administered object attributes), they will need to be rewritten. (Re-compiling MQ 3.0 clients with MQ 3.5 will show which MQ 3.0 attributes have been renamed in MQ 3.5. The old names will still work.)
- Scripts that start Java clients and which set administered object attribute values using command line options are compatible with MQ 3.5. However, if they are to use the *new* administered object attributes (see Chapter 4 and Appendix A of the MQ *Java Client Developer's Guide* for information on administered object attributes), they will need to be rewritten.

Administration Tool Compatibility

Because of the addition of new commands and new administrative capabilities, the MQ 3.5 administration tools (the Administration Console and command line utilities) only work with MQ 3.5 brokers. However, all MQ 3.0 commands and command options remain supported.

Client Compatibility

When upgrading from MQ 3.0.x to MQ 3.5, you should consider the following, regarding Java clients:

- An MQ 3.5 broker will support an MQ 3.0.x client (but without additional MQ 3.5 capabilities).
- An MQ 3.5 Java client can connect to an MQ 3.0.x broker (but without additional MQ 3.5 capabilities).
- Java clients built on JDK 1.2, 1.3, or 1.4 can inter-operate with a broker running JRE 1.4. However, clients that use a secure (SSL-based) connection to a broker will require additional JSSE and JNDI libraries if they are not built on JDK 1.4 (which includes these libraries). These libraries are provided on Solaris in the `SUNWiqsup` package, which is not installed by default; they are provided on Windows as part of a "custom" install; on linux they are installed as part of the default MQ install.
- C clients programmed cannot connect to an MQ 3.0.x broker; they are supported only by MQ 3.5 brokers running with a trial or enterprise license.

Where To Go Next

When you are ready to install MQ on a specific platform, see the appropriate chapter for your platform (Solaris, Linux, or Windows). Each chapter contains hardware and software requirements, installation procedures, and other relevant instructions, such as how to upgrade editions and how to proceed after installation.

Chapter 2

Solaris Installation

This chapter explains the following topics as they apply to a Solaris installation:

- Hardware and Software Requirements
- Installing MQ on Solaris
- Configuring MQ for Automatic Startup
- Configuring the Java Runtime for MQ
- Upgrading from Platform to Enterprise Edition
- Where To Go Next
- Uninstalling MQ on Solaris

Hardware and Software Requirements

At a minimum, your Solaris™ system (SPARC™ or X86) should satisfy the requirements indicated in the following table.

Table 2-1 Hardware and Software Requirements for Solaris

Component	Requirements
Operating system	Solaris 8, Solaris 9, (SPARC platforms) Solaris 9 (X86 platform) **Note:** To ensure proper operation of MQ, you should install all required Solaris Patches for Java 2. For the latest information about the patches and to download the recommended and required patches, see: `http://java.sun.com/j2se/1.4.2/download.html`

Table 2-1 Hardware and Software Requirements for Solaris *(Continued)*

Component	Requirements
CPU	Sun UltraSPARC™
	Intel Pentium 2 (or compatible)
RAM	256 Mbytes
Disk space	The compressed installation file is approximately 9 Mbytes.
	The temporary working directory used for extracting the installation files requires an additional 9 Mbytes.
	The installed product requires approximately 9 Mbytes of hard drive space. MQ, however, may need more space if the broker stores persistent messages locally.
Java 2 Standard Edition (J2SE)	See Table 1-1 on page 19 for the supported versions of the Java Runtime Environment (JRE) and Java Software Development Kit (JDK) that are supported on Solaris.
	The MQ software distribution CD includes the required JRE version at the time of release.

Installing MQ on Solaris

The MQ product can be downloaded from the Sun ONE website or installed from the product CD-ROM. See the appropriate section below for details.

NOTE If you are upgrading from MQ 3.0 versions, you need not uninstall those versions—they will be over-written. However, if you are upgrading from MQ 2.0, it is recommended that you first uninstall MQ 2.0 software, as described in the MQ 2.0 *Installation Guide*, before installing MQ 3.5. Also check the MQ 3.0.1 *Release Notes* for information about migrating any existing MQ 2.0 data you wish to preserve.

NOTE Because MQ is installed with other products (such as Solaris 9, Sun ONE Application Server, and possibly others), you might wish to check whether MQ has already been installed on your system. To do so, enter the following command:

```
pkginfo | grep SUNWiq
```

If MQ packages are already installed, you can check the version by entering:

```
pkgparam -v packageName | grep PRODVERS
```

where *packageName* is any of the MQ packages. The package version is the value of `SUNW_PRODVERS`.

Installed Directory Structure

The install image below reflects a full (all packages) Solaris installation. This image might vary if you perform a partial installation.

NOTE File locations for MQ bundled with Sun ONE Application Server, may differ from locations specified in Table 2-2.

Table 2-2 Installed Directory Structure: Solaris

Solaris	Contents
`COPYRIGHT` (not installed)	Copyright text file
`LICENSE` (not installed)	License text file
`LICENSEREADME/THIRDPARTY-LICENSEREADME` (not installed)	License for third party software distributed by MQ: DOM4J, Apache Commons Logging
`README` (not installed)	README text file

1. See Table 1-2 on page 20 for the versions of NSPR and NSS needed to support the C-API.

Table 2-2 Installed Directory Structure: Solaris *(Continued)*

Solaris	Contents
`/usr/bin/`	Contains the executables for the broker (`imqbrokerd`) and the following MQ administration tools: • Administration Console (`imqadmin`) • Command (`imqcmd`) • Object Manager (`imqobjmgr`) • User Manager (`imqusermgr`) • Database Manager (`imqdbmgr`) • Key Tool (`imqkeytool`)
`/usr/share/lib/`	Contains files that support the MQ Java client runtime: `/*.jar` jar files used to build and run Java JMS client applications `/*.war` war files used for HTTP servlet deployment `/*.rar` rar files used for MQ resource adapter
`/usr/share/lib/imq/`	Contains files used to support MQ tools and processes: `/ext` location for placing files needed for plug-in persistence capability `/props` subdirectory contains the broker's default configuration files `/help` subdirectory contains MQ help files `/images`
`/etc/imq/`	Contains license files, instance template files, and rc script configuration files that can be used for automatic startup
`/var/imq/`	Working storage directory for MQ. `/instances` subdirectory which will contain configuration files, log files, access control files, flat-file user repository, and file-based persistent data stores for each broker instance
`/usr/share/javadoc/imq/`	Contains the MQ and JMS API documentation distributed as Javadoc (HTML)
`/usr/demo/imq/`	Source code for and instructions on how to run Java client example applications:
`/opt/SUNWimq/include/`	Header files needed to support MQ C clients (Enterprise Edition only)
`/opt/SUNWimq/lib/`	Libraries needed to support MQ C clients (Enterprise Edition only)[1]

1. See Table 1-2 on page 20 for the versions of NSPR and NSS needed to support the C-API.

Table 2-2 Installed Directory Structure: Solaris *(Continued)*

Solaris	Contents
`/opt/SUNWimq/demo/C/`	Examples for writing C clients (Enterprise Edition only)

1. See Table 1-2 on page 20 for the versions of NSPR and NSS needed to support the C-API.

Installing from the Web

The following instructions explain how to download and install the MQ product on Solaris from the Sun ONE website.

➤ **To Install MQ on Solaris from the Web**

1. Read the product license. Installation and use of the product is subject to acceptance of the license agreement.
2. Download the MQ product distribution file from the website into an empty, temporary working directory, *temp_directory*.

   The zipped distribution file name depends on the MQ edition and hardware platform:

Table 2-3 MQ 3.5 Solaris Distribution Files

Edition	SPARC	X86
Platform	`imq3_5-plt-solsparc.zip`	`imq3_5-plt-soli386.zip`
Enterprise	`imq3_5-ent-solsparc.zip`	`imq3_5-ent-soli386.zip`

3. Change directories:

   `cd` *temp_directory*

4. Unzip the distribution file.

   `unzip imq3_5-`*edition*`-`*platform*`.zip`

   where *edition* is `plt` or `ent`, depending on the edition, and *platform* is `solsparc` or `soli386`, depending on the platform.

The `unzip` command creates a `mq3_5-`*edition* directory which contains the distribution files: `LICENSE`, `README`, `LICENSEREADME\THIRDPARTYLICENSEREADME`, and `COPYRIGHT` files; install and uninstall scripts; a `patches` directory; and a `pkgs` directory that contains the MQ packages.

Table 2-4 describes the MQ packages, and Table 2-5 on page 35 provides a guide to the packages you need for different use scenarios. In addition, if any of these files already exist on your system, you need to check whether the patch revision number is greater than that provided by MQ. If it is, you should do a custom install.

Table 2-4 Packages in Solaris Bundle

#	Package	Description	Note
1	`SUNWaclg`[1]	Apache Commons Logging Framework: API and runtime	Required to support Java clients using SOAP/JAXM API.
2	`SUNWiqcdv`	C header files and demo files	Required for developing C client programs (Enterprise Edition only).
3	`SUNWiqcrt`	C client shared libraries	Required for running C client programs (Enterprise Edition only).
4	`SUNWiqdoc`	MQ Java client API javadoc and example applications	Needed only for Java client development.
5	`SUNWiqfs`	MQ JNDI File System Service Provider	Required only for client development and administration tools that use the JNDI FIle System Service Provider. The JNDI Service Provider is not supported for deployment.
6	`SUNWiqjx`	MQ Java API for XML Messaging (JAXM): API and runtime	Required to support Java clients using SOAP/JAXM API.
7	`SUNWiqlpl` or `SUNWiqlen`	MQ license files for Platform or Enterprise Edition message server	Depends on MQ edition.
8	`SUNWiqr`	MQ message server root package	Needed for MQ executables.
9	`SUNWiqsup`	JNDI and JSSE jar files	Includes the JNDI and JSSE jar files needed for client development and deployment with JDK 1.2 and 1.3.

1. This package is shared by a number of Sun ONE products.

Table 2-4 Packages in Solaris Bundle *(Continued)*

#	Package	Description	Note
10	SUNWiqu	MQ message server and administration tools	
11	SUNWiquc	MQ Java API for JMS messaging and Java client runtime.	Required to support Java clients using JMS API.
12	SUNWiqum	MQ JMS/SOAP Message Transformer API and runtime	Required to perform conversions between SOAP messages and JMS messages.
13	SUNWjaf[1]	JavaBeans Activation Framework: API and runtime	Required to support Java clients using SOAP/JAXM API.
14	SUNWjhrt[1]	JavaHelp API and runtime	Required if installing on Solaris 8. (Solaris 9 and above already have this package installed.) Will only install if a JVM 1.4 or greater has first been installed.
15	SUNWjmail[1]	JavaMail: API and runtime	Required to support Java clients using SOAP/JAXM API.
16	SUNWpr[1]	Netscape Portable Runtime (NSPR) libraries	Needed to support C clients (Enterprise Edition only).
17	SUNWtls[1]	Network Security Services (NSS) libraries.	Needed to support C language clients (Enterprise Edition only).
18	SUNWxsrt[1]	SOAP with Attachments API for Java: API and runtime	Required to support Java clients using SOAP/JAXM API.

1. This package is shared by a number of Sun ONE products.

Table 2-5 provides a guide to the packages you need for different use scenarios:

Table 2-5 Packages Required for Various Scenarios

Scenario	Packages Needed	Notes
MQ message server and administration tools	SUNWiqr SUNWiqu SUNWiqlpl or SUNWiqlen SUNWiquc SUNWjhrt (optional) SUNWiqfs (optional)	Required for a MQ message server to run on a host.

Table 2-5 Packages Required for Various Scenarios *(Continued)*

Scenario	Packages Needed	Notes
Developing and/or deploying Java clients using the JMS API	SUNWiquc SUNWiqdoc (optional) SUNWiqsup (JDK 1.2 and 1.3 only)	Can be installed on a system without a MQ message server.
Developing and/or deploying Java clients using the SOAP/JAXM API	SUNWaclg SUNWjaf SUNWjmail SUNWiqjx SUNWxsrt SUNWiqdoc (optional)	Can be installed on a system without a MQ message server. Note: SOAP clients require JDK1.4
Developing and/or deploying Java clients using the JMS/SOAP Message Transformer	SUNWiqum Plus all packages needed to support Java clients using both the JMS and SOAP/JAXM API	Can be installed on a system without a MQ message server. The MQ Message Transformer API depends on both the JMS and SOAP APIs.
Developing and/or deploying C clients	SUNWiqcrt SUNWiqcdv SUNWpr SUNWtls	Enterprise Edition only.

5. Change to the directory containing the MQ distribution files.

   `cd mq3_5-`*`edition`*

6. Become root:

   `su root`

   When prompted, type your root password.

7. Check your system for patches to any of the non-MQ-specific packages listed in Table 2-4 on page 34 (packages which do not begin with `SUNWiq` and which might already be installed on your system). Enter the following command to check for patches:

   `showrev -a`

Output indicating incompatible patches looks like this:

```
Patch: 114045-05 Obsoletes: Requires: Incompatibles:
       Packages: SUNWpr, SUNWtls
```

If patches are shown to packages listed in Table 2-4 on page 34, you might not want to install those particular MQ packages, and you should pay close attention to Step 8a.

If you mistakenly downgrade your NSS or NSPR packages back to the unpatched versions, you can upgrade to the patched versions by installing the patches that are included in the MQ solaris bundle:

a. Remove the old patches:

```
#patchrm 114045-XX (sparc)
#patchrm 114049-XX (sparc)
                        //use actual patch # shown in showrev -a
```

or

```
#patchrm 114046-XX (i386)
#patchrm 114050-05 (i386)
```

b. Install the patches back

```
#cd mq3_5-<edition>/patches
```

```
#patchadd 114045-xx (sparc)
```

or

```
#patchadd 114046-xx (i386)
```

8. If you intend to install *all* the packages listed in Table 2-4 on page 34, you will need to run the `mqinstall` script. Please note that this script will overwrite all the listed packages.

a. run the `mqinstall` script.

```
./mqinstall
```

The script lists the distribution packages, if any, that are already installed, and then lists the packages about to be installed.

NOTE If any of the patches found in Step 7 are packages that are already installed, then you will probably not want to install the corresponding MQ packages. In that case, enter n (no) and continue with Step 9.

NOTE The `mqinstall` utility does not install package #9 (`SUNWiqsup`) in Table 2-4 on page 34. If you are doing client development using JDK 1.2 or 1.3, you will need to install the `SUNWiqsup` package as described in Step 9.

b. Enter `y` (yes) if you want to install all the packages, and skip to Step 10. Otherwise, enter `n` (no) and continue with Step 9.

If you run the script, it creates a log file in the following directory:

```
/var/sadm/install/logs/
```

9. If you do not want to install all of the packages listed in Table 2-4 on page 34 (you want only a subset, or you don't want to over-write later versions of packages),

a. change to the `pkgs` directory

```
cd pkgs
```

b. run the `pkgadd` command to install the packages you want:

```
pkgadd -d ./ -a admin.conf
```

The `pkgadd` utility lists the names of all packages in the directory available for installation (see Table 2-4). When prompted, indicate the packages you want to install. (The `-a admin.conf` option permits an over-write of any packages that are already installed on your system.)

The `pkgadd` utility installs the packages you specified, perhaps asking for additional information, and eventually returns to the original prompt, displaying the list of packages available for installation.

c. Type `q` to quit.

10. Exit the root shell.

11. Back up the zip distribution file from your temporary working directory.

This is your logical media. It will be needed to uninstall or reinstall MQ. Treat this file as you would any other installation media and place a copy in a safe location.

12. Clean up all remaining files in your temporary working directory. For your convenience, you should save the `uninstall` script.

NOTE The instance data for any pre-existing broker instance (including the default broker instance, named `imqbroker`) is owned by whoever created that instance. Therefore, once installation is complete, be sure to run any MQ broker instance as the owner with privileges to the `/var/imq/instances/`*instanceName* directory. (You can do this using the `-name` *instanceName* option.) As an alternative, you can log in as `root` and change the privileges on the directory. See the MQ *Administrator's Guide* for more information.

Installing from CD-ROM

The following instructions explain how to install the MQ product on Solaris from CD-ROM.

NOTE A compressed installation file (zip) is provided on the CD if you do not want to install MQ directly from the packages on the CD.

➤ **To Install MQ on Solaris from CD-ROM**

1. Log in as root or change to superuser.

   For example, type the following at a command prompt:

   ```
   su root
   ```

   Then type your superuser password.

2. Insert the MQ CD into your CD-ROM drive.

   If the Volume Manager™ software is running on your machine, the CD-ROM is automatically mounted to the `/cdrom/messagequeue3_5` directory.

   If the Volume Manager is not running on your machine:

   - Create a directory called `/cdrom/messagequeue3_5` by typing:

     ```
     mkdir -p /cdrom/messagequeue3_5
     ```

   - Mount the CD-ROM manually:

     `mount -rF hsfs` *cdrom-device* `/cdrom/messagequeue3_5`

   An example of *cdrom-device* is `/dev/dsk/c0t0d0s0`.

NOTE Volume Manager is a tool provided on Solaris that allows you to perform administrative tasks, such as mounting CD-ROMs, more easily. Volume Manager mounts a CD-ROM as `/cdrom/`*name_of_media*, where *name_of_media* is determined from the CD-ROM itself.

Open and read (using your preferred text editor) the `LICENSE` file located in the `solaris-sparc/` or `solaris-x86/` directory of the CD.

- If you choose NOT to accept the license agreement, discontinue installation and contact the place where you purchased the product to determine the return policy.
- If you choose to accept the agreement, continue with the installation steps below.

3. Change to the directory on the CD containing the installation packages. For example, type:

   ```
   cd /cdrom/messagequeue3_5/solaris-sparc
   ```

   or

   ```
   cd /cdrom/messagequeue3_5/solaris-x86
   ```

   The contents of this directory vary depending on the product edition.

4. Continue with Step 6 on page 36 under "Installing from the Web.".

Configuring MQ for Automatic Startup

If you wish to set the broker (the MQ message server) for automatic startup, you need to become root and edit the following configuration file:

`/etc/imq/imqbrokerd.conf`

The startup properties you can set in this configuration file are shown in Table 2-6:

Table 2-6 Broker Startup Configuration Properties

Property Name	Description
AUTOSTART	Specifies (YES/NO) if the broker is automatically started at boot time. Default: NO
ARGS	Specifies command line options and arguments to pass to the broker startup command. See the MQ *Administrator's Guide* for a listing and description of `imqbrokerd` command line options. (For example `-name` *instanceName*)
RESTART	Specifies (YES/NO) if the broker is automatically re-started if it abnormally exits. Default: YES

To check that startup changes are correct (without booting the system), you can, as root, explicitly run the MQ initialization script (`S52imq`) in "debug" mode:

```
env DEBUG=1 /etc/rc3.d/S52imq start
```

Configuring the Java Runtime for MQ

At startup time, a broker (the MQ message server) checks to make sure it has access to the required Java runtime version (JDK/JRE 1.4). There are a number of ways you can configure or set the JRE used by the broker. These are shown in the following list, in order of precedence:

1. Pass in the JDK or JRE using either the `imqbrokerd -javahome` or `-jrehome` command line options, respectively (if both are passed in, the last one on the command line will take precedence).
2. Set the JDK or JRE in the `IMQ_JAVAHOME` environment variable.
3. Let the broker use the installed JDK.

   This is the JDK located in `/usr/j2se`

To figure out why a broker is picking up a specific JDK/JRE, you can start the broker with the following command:

```
imqbrokerd -verbose
```

Upgrading from Platform to Enterprise Edition

MQ comes in two editions, as explained in "Product Editions" on page 17. Upgrading from the Platform to Enterprise Edition must be done at the same version level; that is, you can upgrade from Platform Edition 3.5 to Enterprise Edition 3.5, but you cannot upgrade from Platform Edition 3.0.1 to Enterprise Edition 3.5

To upgrade from the Platform Edition to the Enterprise Edition, you need to first purchase the Enterprise Edition, and then install, as described below, the Enterprise Edition license contained in the distribution, plus the C-API packages, if you wish to write C client programs. Installation of these packages does not overwrite the MQ 3.5 modules already installed and does not modify the configuration of your MQ messaging system.

To upgrade to the Enterprise Edition, you need only the following packages:

Table 2-7 Upgrade Packages in Solaris Enterprise Bundle

#	Package	Description	Note
7	SUNWiqlen	MQ license file for Enterprise Edition message server	
3	SUNWiqcrt	C-API shared libraries	Required for running C client programs (Enterprise Edition only)
2	SUNWiqcdv	C-API header files and demo files	Required for developing C client programs (Enterprise Edition only)
16	SUNWpr[1]	Netscape Portable Runtime (NSPR) libraries.	Needed to support C client programs (Enterprise Edition only)
17	SUNWtls[1]	Network Security Services (NSS) libraries.	Needed to support C client programs (Enterprise Edition only)

1. This is a shared package.

➤ To Upgrade to Enterprise Edition on Solaris

1. Stop any running brokers.

   `imqcmd shutdown bkr -u` *name* `-p` *password* `[-b` *hostName:port*`]`

2. Follow the installation procedure in "Installing from the Web" on page 33, Step 1 through Step 6 or in "Installing from CD-ROM" on page 39, Step 1 through Step 3.

3. Add the Enterprise Edition packages:

   a. Change to the `pkgs` directory

      ```
      cd pkgs
      ```

   b. Run the `pkgadd` command to install the packages in Table 2-7 that you want; you might want only a subset, or you might not want to over-write later versions of packages. For example:

      ```
      pkgadd -d . SUNWiqlen
      ```

   c. Type **q** to quit.

4. Verify that the Enterprise Edition license is available by running the following command:

   ```
   imqbrokerd -license
   ```

Where To Go Next

Read the `README` and MQ *Release Notes* files.

- The `README` includes information on where to find documentation, news and updates, and how to send feedback.
- The MQ *Release Notes* contain information on code and documentation changes, open bugs, and important technical notes. This document is available on the Sun ONE website.

For an overview of Message Queue concepts, a brief introduction to writing and compiling a client application, see the MQ *Java Client Developer's Guide*.

For details on configuring brokers and managing an MQ messaging system, see the MQ *Administrator's Guide*.

For class and member information used when writing a client application, browse the API documentation in the `/usr/share/javadoc/imq` directory.

To uninstall the product, see the following section.

Uninstalling MQ on Solaris

The following instructions explain how to uninstall MQ.

➤ **To Remove MQ on Solaris**

1. Stop any running client applications.
2. Stop any running brokers.

   `imqcmd shutdown bkr -u` *name* `-p` *password* `[-b` *hostName:port*`]`
3. If you want to delete dynamic data, the MQ flat-file user repository, and the MQ access control file associated with each broker instance, remove this data using the following command.

   `imqbrokerd -name` *instanceName* `-remove instance`
4. Get the uninstall script from the distribution CD or follow Step 5 through Step 7. If you already have the uninstall script you can proceed to Step 8.
5. Find the zipped distribution file used to install MQ (see Step 11 on page 38) and place it in a temporary directory, *temp_directory*.
6. Change directories to *temp_directory*:

   `cd` *temp_directory*
7. Unzip the distribution file.

   `unzip imq3_5-`*edition*`-`*platform*`.zip`

   where *edition* is `plt` or `ent`, depending on the edition, and *platform* is `solsparc` or `soli386`, depending on the platform.
8. Change to the directory containing the MQ packages.

   `cd mq3_5-`*edition*
9. Become root:

   `su root`

   When prompted, type your root password.
10. Run the un-install script.

    `./mquninstall`

    The installation script lists MQ packages that are not shared, if any, that are currently installed. (It does not list shared MQ packages installed with MQ, and which might be in use by other programs.)

11. If you want to un-install *all* the listed packages, enter `y` (yes), and skip to Step 16. Otherwise, continue with Step 12.
12. If you do not want to un-install all of the MQ packages, then enter `n` (no), and use the `pkgrm` command, as described in Step 13, to un-install the specific packages you want to remove.
13. Remove the MQ packages, using the following command:

    `pkgrm` *packageName*

    where *packageName* is any of the MQ packages. To remove multiple packages, separate the package names by a space.

    Because other products might be using MQ packages, be careful about removing them. The `pkgrm` command will warn you of any dependencies on a package before removing it.
14. When prompted, confirm your removal request by typing **`y`**.
15. Type **`q`** to quit.
16. Exit the root shell.

Chapter 3

Linux Installation

This chapter explains the following topics as they apply to a Linux installation:

- Hardware and Software Requirements
- Installing MQ on Linux
- Configuring the Java Runtime for MQ
- Upgrading from Platform to Enterprise Edition
- Where To Go Next
- Uninstalling MQ on Linux

Hardware and Software Requirements

At a minimum, your Linux system should satisfy the minimum requirements indicated in the following table.

Table 3-1 Hardware and Software Requirements for Linux

Component	Requirements
Operating system	Red Hat Advanced Server 2.1 Update 2
CPU	Intel Pentium 2 (or compatible)
RAM	256 Mbytes
Disk space	The zip file containing the product is approximately 9 Mbytes. The installed product requires approximately 8 Mbytes of hard drive space. MQ, however, may need more space if the broker stores persistent messages locally.

Table 3-1 Hardware and Software Requirements for Linux *(Continued)*

Component	Requirements
Java 2 Standard Edition (J2SE)	See Table 1-1 on page 19 for the supported versions of the Java Runtime Environment (JRE) and Java Software Development Kit (JDK) that are supported on Linux. The MQ software distribution CD includes the required JRE version at the time of release.

Installing MQ on Linux

The Sun ONE Message Queue product can be downloaded from the Sun ONE website or installed from the product CD-ROM. See the appropriate section below for instructions.

NOTE If you are upgrading from MQ 3.0 versions, you are required to first uninstall MQ 3.0 version software, as described in the following section.

Finding and Removing Earlier MQ Versions

Because MQ is installed with other products (such as Solaris 9, Sun ONE Application Server, and possibly others), you should check whether MQ has already been installed on your system, and uninstall it before installing MQ 3.5.

If you find that an earlier version of MQ is already on your system, you need to decide if you want to preserve any security-related data (flat-file user repository, access control file, or passfile). If, so, you will need to use the `mqmigrate` utility, as explained in subsequent paragraphs, before uninstalling the earlier version of MQ. If you do not want to preserve your old data files, you do not have to run the `mqmigrate` utility.

Note that RPM's are now relocatable for all files in the `/opt` directory. You can use `--prefix=dir` with the `-i` option if you want to change the default `/opt` location.

Depending on the version, MQ might have been installed using tar files or the Red Hat Package Manager (RPM). To check for installed versions, therefore, you need to check for both. It is recommended that you check first for RPM installations and then for tar file installations.

Migrating Files from Non-Standard Locations

The `mqmigrate` utility has a new option that allows you to migrate data that has been installed in a non-standard location. The syntax for the new option is as follows:

```
-basedir <baseDirectory>
```

Where *baseDirectory* specifies the non-standard directory where the files were installed. For example, if the old data was untar'ed in the `/export` directory, you should migrate the old data using the command

```
./mqmigrate -basedir /export
```

Finding and Removing MQ RPM's (Version 3.0.1 SP1 or Later Only)

➤ **To Find and Remove Earlier RPM-Installed Versions of MQ**

1. Enter the following command:

   ```
   rpm -qa | grep imq
   ```

   If found, the version numbers of any RPM's are imbedded in the RPM name. If none are found, proceed to "Finding and Removing an MQ Tar-Based Installation."

2. If you find MQ RPM's of a version earlier than 3.5, remove them as follows:

   a. Run the `mqmigrate` utility (in the MQ 3.5 distribution) after completing Step 5 on page 53):

      ```
      ./mqmigrate
      ```

      The `mqmigrate` utility moves existing broker instance data (broker configuration files and persistent data) and security-related files, to new MQ 3.5 locations.

   b. Remove the existing MQ installed software.

      `rpm -e` *RPMName*

Finding and Removing an MQ Tar-Based Installation

➤ **To Find and Remove Earlier Tar-Based Installed MQ**

1. See if the default MQ installation directory (`/opt/imq/bin`) exists on your system.

   If found, proceed to Step 2.

   If not found, MQ might have been installed in a non-default location. If you cannot remember the installation directory, search for the MQ `imqbrokerd` executable and note its root install directory. Proceed to Step 2.

   If there is no previous MQ installation, proceed to install MQ by following the directions in "Installing from the Web" on page 52 or "Installing from CD-ROM" on page 54.

2. If you find an earlier MQ installation in the default location (`/opt/imq/bin`), remove it as follows:

   a. Run the `mqmigrate` utility (in the MQ 3.5 distribution) after completing Step 5 on page 53):

      ```
      ./mqmigrate
      ```

      The `mqmigrate` utility moves existing broker instance data (broker configuration files and persistent data) and security-related files, to new MQ 3.5 locations.

   b. Remove the `/opt/imq/` directory and all its contents.

      ```
      rm -rf /opt/imq
      ```

Installed Directory Structure

The install image below reflects a full (all RPM's) Linux installation. This image might vary if you perform a partial installation.

NOTE File locations for MQ bundled with Sun ONE Application Server, may differ from locations specified in Table 3-2.

Table 3-2 Installed Directory Structure: Linux

Linux	Contents
COPYRIGHT (not installed)	Copyright text file
LICENSE (not installed)	License text file
LICENSEREADME/THIRDPARTY-LICENSEREADME (not installed)	License for third party software distributed by MQ: DOM4J, Apache Commons Logging
README (not installed)	README text file
/opt/imq/bin/	Contains the executables for the broker (imqbrokerd) and the following MQ administration tools: • Administration Console (imqadmin) • Command (imqcmd) • Object Manager (imqobjmgr) • User Manager (imqusermgr) • Database Manager (imqdbmgr) • Key Tool (imqkeytool)
/opt/imq/lib/	Contains files that support the MQ Java client runtime: /*.jar jar files used to build and run Java JMS client applications /*.war war files used for HTTP servlet deployment /*.rar rar files used for MQ resource adapter
/opt/imq/lib/	Contains files used to support MQ tools and processes: /ext location for placing files needed for plug-in persistence capability /props subdirectory contains the broker's default configuration files /help subdirectory contains MQ help files /images
/etc/opt/imq/	Contains license files, instance template files, and rc script configuration files that can be used for automatic startup
/var/opt/imq/	Working storage directory for MQ. /instances subdirectory which will contain configuration files, log files, access control files, flat-file user repository, and file-based persistent data stores for each broker instance

Table 3-2 Installed Directory Structure: Linux *(Continued)*

Linux	Contents
`/opt/imq/javadoc/`	Contains the MQ and JMS API documentation distributed as Javadoc (HTML)
`/opt/imq/demo/`	Source code for and instructions on how to run Java client example applications:
`/opt/imq/include/`	Header files needed to support MQ C clients (Enterprise Edition only)
`/opt/imq/lib/`	Libraries needed to support MQ C clients (Enterprise Edition only)[2]
`/opt/imq/demo/C/`	Examples for writing C clients (Enterprise Edition only)
	The JRE 1.4 files (on Windows only)

Installing from the Web

The following instructions explain how to download and install the MQ product on Linux from the Sun ONE website.

➤ **To Install MQ on Linux from the Web**

1. Accept the license agreement on the MQ product download site.
2. Download the MQ product distribution file from the website into an empty, temporary download directory, *temp_directory*.

   The download file is: `imq3_5-`*edition*`-linx86.zip`

   where *edition* takes one of the following values: `plt` or `ent`, depending on whether you are installing the Platform Edition or Enterprise Edition, respectively.

3. Change to the *temp_directory* and uncompress the distribution file.

   `unzip imq3_5-`*edition*`-linx86.zip`

   The `unzip` command creates a `imq3_5-`*edition* directory which contains the distribution files: `LICENSE`, `README`, `THIRDPARTYLICENSEREADME`, and `COPYRIGHT` files; `mqmigrate` utility; and an `rpms` directory that contains the following RPM's:

   `imq-3_5-`*releaseVersion*`.i386.rpm`
   (included in both Platform and Enterprise Editions)

   `imq-ent-3_5-`*releaseVersion*`.i386.rpm`
   (included in Enterprise Edition only)

4. Change to the directory containing the MQ distribution files.

   `cd mq3_5-`*edition*

5. Log in as root or change to superuser.

   For example, type the following at a command prompt:

   `su root`

   When prompted, type your root password.

6. Install the appropriate RPM's. Note that the Enterprise Edition requires Platform Edition to be installed.

   `rpm -ivh rpms/imq-3_5-`*releaseVersion*`.i386.rpm`
   (Platform Edition)

   `rpm -ivh rpms/imq-ent-3_5-`*releaseVersion*`.i386.rpm`
   (Enterprise Edition)

   The `/opt/imq` directory and its contents are created, and files are placed in other locations as well (see "Upgrade Issues from MQ 3.0.x Versions" on page 21).

7. Back up the `imq3_5-`*edition*`-linx86.zip` file from your temporary working directory.

   This is your logical media. Treat this file as you would any other installation media. Place a copy in a safe location in case you encounter a situation (such as a system failure) that requires reinstallation of the product.

8. Clean up all remaining files in your temporary working directory.

NOTE The instance data for any pre-existing broker instance is owned by whoever created that instance. Therefore, once installation is complete, be sure to run any MQ broker instance (using the `-name` *instanceName* option) as the owner with privileges to the `/var/imq/instances/`*instanceName* directory. This applies to the default broker instance (named `imqbroker`), as well.

Installing from CD-ROM

The following instructions explain how to install the MQ product on Linux from CD-ROM.

➤ **To Install MQ on Linux from CD-ROM**

1. Insert the Sun ONE Message Queue CD into your CD-ROM drive and mount it.

   ```
   mount /mnt/cdrom/linux
   ```

   This command may vary depending on the version of Linux being used. Check the `mount` man page on the system.

2. Change to the directory on the CD containing the MQ distribution:

   ```
   cd /cdrom/linux
   ```

   The contents of this directory vary depending on the MQ product edition.

3. Continue with Step 5 under "Installing from the Web" on page 52.

Configuring the Java Runtime for MQ

At startup time, a broker (the MQ message server) checks to make sure it has access to the required Java runtime version (JDK/JRE 1.4). There are a number of ways you can configure or set the JRE used by the broker. These are shown in the following list, in order of precedence:

1. Pass in the JDK or JRE using either the `imqbrokerd -javahome` or `-jrehome` command line options, respectively (if both are passed in, the last one on the command line will take precedence).

2. Set the JDK or JRE in the `IMQ_JAVAHOME` environment variable.

3. Let the broker use the installed JDK/JRE. The broker will pick up the latest version JDK/JRE installed on the system (greater than 1.4 and less than 2.0).

   The JDK is located in `/usr/java/j2sdk1.`*x.x*, and

   the JRE is located in `/usr/java/j2re1.`*x.x*

To find out why a broker is picking up a specific JDK/JRE, you can start the broker with the following command:

```
imqbrokerd -verbose
```

Upgrading from Platform to Enterprise Edition

MQ comes in two editions, as explained in "Product Editions" on page 17.

To upgrade from the Platform Edition to the Enterprise Edition, you need to first purchase the Enterprise Edition, and then install, as described below, only the Enterprise Edition license contained in the distribution. Installation of the license does not overwrite the MQ 3.5 modules already installed and does not modify the configuration of your MQ messaging system.

➤ **To Upgrade to Enterprise Edition on Linux**

1. Stop any running brokers.

   `imqcmd shutdown bkr -u` *name* `-p` *password* `[-b` *hostName:port*`]`

2. Follow the installation procedure in "Installing from the Web" on page 52, Step 1 through Step 5 or in "Installing from CD-ROM" on page 54, Step 1 through Step 3.

3. Install the MQ Enterprise Edition RPM.

   ```
   rpm -ivh rpms/imq-ent-3_5-01.i386.rpm
   ```

   This installs the Enterprise Edition license and C-API support files into the appropriate MQ directories.

4. Verify that the Enterprise Edition license is available by running

   ```
   imqbrokerd -license
   ```

Where To Go Next

Read the `README` and MQ *Release Notes* files.

- The `README` includes information on where to find documentation, news and updates, and how to send feedback.
- The MQ *Release Notes* contain information on code and documentation changes, open bugs, and important technical notes. This document is available on the Sun ONE website.

For an overview of Sun ONE Message Queue concepts, a brief introduction to writing and compiling a client application, see the MQ *Java Client Developer's Guide*.

For details on configuring brokers and managing an MQ messaging system, see the MQ *Administrator's Guide*.

For class and member information used when writing a client application, browse the API documentation in the `/opt/imq/javadoc` directory.

To uninstall the product, see the following section.

Uninstalling MQ on Linux

The following instructions explain how to uninstall MQ on Linux.

➤ **To Remove MQ on Linux**

1. Stop any running client applications.
2. Stop any running brokers.

   `imqcmd shutdown bkr -u` *name* `-p` *password* `[-b` *hostName:port*`]`
3. Unless you want to retain dynamic data, the MQ flat file user repository, and the MQ access control file associated with each broker instance, remove this data using the following command.

   `imqbrokerd -name` *instanceName* `-remove instance`
4. Become root:

   `su root`
5. Remove the MQ product.

   Issue the following command(s) in the following order:

   `rpm -e imq-ent` (for Enterprise Editions

   `rpm -e imq` (for Platform edition only)

Chapter 4

Windows Installation

This chapter explains the following topics as they apply to a Windows installation:

- Hardware and Software Requirements
- Installing MQ on Windows
- Installation Defaults
- Troubleshooting Installation Problems
- Upgrading from Platform to Enterprise Edition
- Where To Go Next
- Uninstalling MQ on Windows

Hardware and Software Requirements

At a minimum, your Windows system should satisfy the requirements indicated in the following table.

Table 4-1 Hardware and Software Requirements for Windows

Component	Requirements
Operating system	Windows XP Professional, 2000 Professional SP3, 2000 Advanced Server SP3, .NET Server 2003 Enterprise Edition
CPU	Intel Pentium 2
RAM	256 Mbytes

Table 4-1 Hardware and Software Requirements for Windows *(Continued)*

Component	Requirements
Disk space	The self-extracting installation file is approximately 25 Mbytes. The temporary directory used for extracting the installation files requires an additional 45 Mbytes. The installed product requires approximately 45 Mbytes. MQ, however, may need more space if the broker stores persistent messages locally.
Java 2 Standard Edition (J2SE)	See Table 1-1 on page 19 for the supported versions of the Java Runtime Environment (JRE) and Java Software Development Kit (JDK) that are supported on Windows. The MQ installation program will (optionally) install the required JRE version in `IMQ_HOME\jre`.

Installing MQ on Windows

The Sun ONE Message Queue product can be downloaded from the Sun ONE website or installed from the product CD-ROM.

The following instructions explain how to install the MQ product either by downloading from the Sun ONE website or using the CD-ROM.

NOTE If you are upgrading from MQ 3.0.x versions, it is recommended that you first uninstall MQ software as described in the MQ *Installation Guide* of the appropriate version, before installing MQ 3.5.

If you are upgrading from MQ 2.0, it is recommended that you first uninstall MQ 2.0 software, as described in the MQ 2.0 *Installation Guide*, before installing MQ 3.5. Also check the MQ 3.0.1 *Release Notes* for information about migrating any existing MQ 2.0 data you wish to preserve.

If you are installing MQ after a previous uninstall, check that any references to the previous MQ installation have been removed from the system's `PATH` environment variable.

Installed Directory Structure

The install image below reflects a full ("Typical") Windows installation. This image might vary if you perform a partial installation.

NOTE File locations for MQ bundled with Sun ONE Application Server, may differ from locations specified in Table 4-2.

Table 4-2 Installed Directory Structure: Windows

Windows[1]	Contents
`.\COPYRIGHT.TXT`	Copyright text file
`.\LICENSE.TXT`	License text file
`.\LICENSEREADME\THIRDPARTY-LICENSEREADME.TXT`	License for third party software distributed by MQ: DOM4J, Apache Commons Logging
`.\README.TXT`	README text file
`.\bin\` On Windows, the executable files have a .exe filename extension. This directory also includes the utility to install and uninstall the broker as a Windows Service (`imqsvcadmin`) as well other executables (`imqbrokersvc`).	Contains the executables for the broker (`imqbrokerd`) and the following MQ administration tools: • Administration Console (`imqadmin`) • Command (`imqcmd`) • Object Manager (`imqobjmgr`) • User Manager (`imqusermgr`) • Database Manager (`imqdbmgr`) • Key Tool (`imqkeytool`)
`.\lib\`	Contains files that support the MQ Java client runtime: `/*.jar` jar files used to build and run Java JMS client applications `/*.war` war files used for HTTP servlet deployment `/*.rar` rar files used for MQ resource adapter

1. Paths are relative to `IMQ_HOME` (see "Directory Variable Conventions" on page 13).
2. See Table 1-2 on page 20 for the versions of NSPR and NSS needed to support the C-API.

Table 4-2 Installed Directory Structure: Windows *(Continued)*

Windows[1]	Contents
.\lib\	Contains files used to support MQ tools and processes:
	/ext location for placing files needed for plug-in persistence capability
	/props subdirectory contains the broker's default configuration files
	/help subdirectory contains MQ help files
	/images
.\etc\	Contains license files and instance template files.
.\var\	Working storage directory for MQ.
	/instances subdirectory which will contain configuration files, log files, access control files, flat-file user repository, and file-based persistent data stores for each broker instance
.\javadoc\	Contains the MQ and JMS API documentation distributed as Javadoc (HTML)
.\demo\	Source code for and instructions on how to run Java client example applications:
.\include\	Header files needed to support MQ C clients (Enterprise Edition only)
.\lib\	Libraries needed to support MQ C clients (Enterprise Edition only)[2]
.\demo\C\	Examples for writing C clients (Enterprise Edition only)
.\jre\	The JRE 1.4 files (on Windows only)

1. Paths are relative to IMQ_HOME (see "Directory Variable Conventions" on page 13).
2. See Table 1-2 on page 20 for the versions of NSPR and NSS needed to support the C-API.

Installing From CD-ROM or the Web

Follow these steps to install MQ 3.5 on Windows.

➤ **To Install MQ on Windows**

1. Quit any other programs you are running.
2. Download the distribution into a temporary working directory.

   If you are installing from CD-ROM, insert the CD into the CD-ROM drive.

3. In the Windows Explorer, double-click the `imq3_5-`*edition*`-win.exe` file

   where *edition* takes one of the following values: `plt` or `ent`, depending on whether you are installing the Platform Edition or Enterprise Edition, respectively.

   The installation file is extracted and setup is launched. If you are installing from CD-ROM, the setup program is started automatically.

4. Read the license agreement. Installation and use of the product is subject to acceptance of the license agreement.

5. Follow the setup program's instruction to select installation options.

   a. Select the type of installation.

      - **Compact** Includes only the files necessary to run the broker, administration, and client applications. No documentation or example client applications are installed.
      - **Custom** Lets you specify which MQ modules you want to install. Your options include the broker, Java client runtime, administration tools, MQ resource adapter, documentation, example client applications, Java runtime, and C client runtime (Enterprise Edition).
      - **Typical** Installs all files, including broker, client runtime, administration tools, documentation, example client applications, and Java runtime.
      - **Upgrade Platform Edition (Enterprise Only)** Installs only the files necessary to enable Enterprise Edition features. This option does not overwrite the MQ files already installed, and does not modify the configuration of your messaging system.

   b. Select a location for the installation.

   c. Select or create a folder for the Windows Start > Programs menu.

   d. Choose whether or not to install the broker as a Windows service. If you choose to install the broker as a Windows service, the broker will be started automatically at system startup and run in the background. This option requires Windows Administrator Group privileges.

      If you choose not to install the broker as a Windows service now but later change your mind, you can use the `imqsvcadmin` utility to do so. See the MQ *Administrator's Guide* for information on using the `imqsvcadmin` utility.

6. Verify installation by choosing Start > Programs > Sun ONE Message Queue 3.5 >Message Broker to run the broker.

   You can also compile and run example applications located in the `c:\Program Files\Sun\MessageQueue3\demo` directory. See the MQ *Java Client Developer's Guide*.

Installation Defaults

The following table indicates the installation defaults for Windows.

Table 4-3 Windows Installation Defaults

Installation directory	The product is installed, by default, into the following directory: `C:\Program Files\Sun\MessageQueue3`
Environment Variables	**IMQ_HOME**: The installer automatically sets the `IMQ_HOME` environment variable to the installation directory. **IMQ_VARHOME**: The `IMQ_VARHOME` environment variable defaults to `IMQ_HOME\var`. **IMQ_JAVAHOME**: The `IMQ_JAVAHOME` environment variable defaults to `IMQ_HOME\jre`. **PATH**: The installer automatically sets `PATH` to `%PATH%;%IMQ_HOME%\bin`. This enables you to run utilities, such as `imqbrokerd`, `imqcmd`, and `imqobjmgr` without specifying the entire path.

Troubleshooting Installation Problems

If the MQ installation program does not complete successfully on your Windows platform, try the following remedies. These apply regardless of whether you installed the product from CD-ROM or downloaded it from the Web.

➤ **To Troubleshoot Installation Problems on Windows**

1. Use the Windows Task Manager to end the MQ installation program.
2. Clear any temporary directories (for example, `temp` or `tmp`).
3. Start the MQ installation again, being certain to install to the same directory as the initial installation attempt.

If your Windows installation of the MQ product is still not successful, try this more comprehensive procedure:

1. Use the Windows Task Manager to end the MQ installation program.
2. Clear any temporary directories (for example, `temp` or `tmp`).
3. In the Services control panel, temporarily stop any non-Windows network-related services (for example, a Solstice NFS Server service).
4. Delete the Message Queue 3.0 directory and all its contents.
5. In your Systems control panel, remove all references to MQ 3.0 from your environment settings.
6. Restart your Windows operating system.
7. Reinstall MQ in the same directory location as the initial failed installation.
8. Reset any services in the Services (Control Panel) that you stopped in Step 3.

Upgrading from Platform to Enterprise Edition

MQ comes in two editions, as explained in "Product Editions" on page 17.

To upgrade from the Platform Edition to the Enterprise Edition, you need to first purchase the Enterprise Edition, and then install, as described below, the Enterprise Edition license contained in the distribution, plus the C-API packages, if you wish to write C client programs. Installation of this software does not overwrite the MQ 3.5 modules already installed and does not modify the configuration of your MQ messaging system.

You can upgrade from the Platform Edition to the Enterprise Edition by installing the Enterprise Edition.

➤ **To Upgrade to Enterprise Edition on Windows**

1. Stop any running brokers.

   `imqcmd shutdown bkr -u` *name* `-p` *password* `[-b` *hostName:port*`]`

2. Follow the installation procedure explained in "Installing MQ on Windows." Choose the Upgrade Platform Edition option when it appears on your screen.

   This setup does not overwrite the MQ modules installed, and does not modify the configuration of your MQ messaging system.

Where To Go Next

Read the `README` and MQ *Release Notes* files.

- The `README` includes information on where to find documentation, news and updates, and how to send feedback.
- The MQ *Release Notes* contain information on code and documentation changes, open bugs, and important technical notes. This document is available on the Sun ONE website.

For an overview of Sun ONE Message Queue concepts, a brief introduction to writing and compiling a client application, see the MQ *Java Client Developer's Guide*

For details on configuring brokers and managing an MQ messaging system, see the MQ *Administrator's Guide*.

For class and member information used when writing a client application, browse the API documentation in the `IMQ_HOME/javadoc` directory.

To uninstall the product, see the following section.

Uninstalling MQ on Windows

The following instructions explain how to uninstall MQ on Windows.

➤ **To Remove MQ on Windows**

1. Stop any running client applications.
2. Stop any running brokers.

   `imqcmd shutdown bkr -u` *name* `-p` *password* `[-b` *hostName:port*`]`

   If you had installed the broker as a Windows service, you can stop it as follows:

   - From the Windows Start menu, choose Settings, then Control Panel.
   - Double-click the Administrative Tools icon, then the Services icon.
   - In the Services panel, select the MQ Broker entry, then click Stop.
   - Close the Services panel.

3. Unless you want to retain dynamic data, the MQ flat file user repository, and the MQ access control file associated with each broker instance, remove this data using the following command.

   `imqbrokerd -name` *instanceName* `-remove instance`

4. Remove the Sun ONE Message Queue product by choosing Start > Programs > Sun ONE Message Queue 3.5 >Uninstall.

5. After Uninstall completes, remove any leftover files.

   The Windows Uninstall utility does not remove files or directories created as a result of using MQ, such as instance configuration files or a flat-file data store.

6. Manually remove MQ references from your environment.

   - From the Windows Start menu, choose Settings.
   - In the Control Panel, double-click the System icon.
   - On the Environment page, select the `PATH` environment variable and remove the `Message Queue 3.5` reference from the value.
   - Click Set then OK.