Sun Java System Message Queue 3.7 UR1 Installation Guide

Sun Microsystems, Inc.
4150 Network Circle
Santa Clara, CA 95054
U.S.A.

Part No: 819–4466–10
April 2007

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 U.S. 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.

This distribution may include materials developed by third parties.

Parts of the product may be derived from Berkeley BSD systems, licensed from the University of California. UNIX is a registered trademark in the U.S. and other countries, exclusively licensed through X/Open Company, Ltd.

Sun, Sun Microsystems, the Sun logo, the Solaris logo, the Java Coffee Cup logo, docs.sun.com, Java, and Solaris 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 an architecture developed by Sun Microsystems, Inc.

The OPEN LOOK and Sun$^{\text{TM}}$ Graphical User Interface was developed by Sun Microsystems, Inc. for its users and licensees. Sun acknowledges the pioneering efforts of Xerox in researching and developing the concept of visual or graphical user interfaces for the computer industry. Sun holds a non-exclusive license from Xerox to the Xerox Graphical User Interface, which license also covers Sun's licensees who implement OPEN LOOK GUIs and otherwise comply with Sun's written license agreements.

Products covered by and information contained in this publication are controlled by U.S. Export Control laws and may be subject to the export or import laws in other countries. Nuclear, missile, chemical or 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.

DOCUMENTATION IS PROVIDED “AS IS” AND ALL EXPRESS OR IMPLIED CONDITIONS, REPRESENTATIONS AND WARRANTIES, INCLUDING ANY IMPLIED WARRANTY OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT, ARE DISCLAIMED, EXCEPT TO THE EXTENT THAT SUCH DISCLAIMERS ARE HELD TO BE LEGALLY INVALID.

Sun Microsystems, Inc. détient les droits de propriété intellectuelle 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 plusieurs brevets américains ou des applications de brevet en attente aux Etats-Unis et dans d'autres pays.

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

Certaines composants de ce produit peuvent être dérivées du logiciel Berkeley BSD, licenciés par l'Université de Californie. UNIX est une marque déposée aux Etats-Unis et dans d'autres pays; elle est licenciée exclusivement par X/Open Company, Ltd.

Sun, Sun Microsystems, le logo Sun, le logo Solaris, le logo Java Coffee Cup, docs.sun.com, Java et Solaris sont des marques de fabrique ou des marques déposées de Sun Microsystems, Inc. aux Etats-Unis et dans d'autres pays. Toutes les marques 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.

L'interface d'utilisation graphique OPEN LOOK et Sun a été développée par Sun Microsystems, Inc. pour ses utilisateurs et licenciés. Sun reconnaît les efforts de pionniers de Xerox pour la recherche et le développement du concept des interfaces d'utilisation visuelle ou graphique pour l'industrie de l'informatique. Sun détient une licence non exclusive de Xerox sur l'interface d'utilisation graphique Xerox, cette licence couvrant également les licenciés de Sun qui mettent en place l'interface d'utilisation graphique OPEN LOOK et qui, en outre, se conforment aux licences écrites de Sun.

Les produits qui font l'objet de cette publication et les informations qu'il contient sont régis par la legislation américaine en matière de contrôle des exportations et peuvent être soumis au droit d'autres pays dans le domaine des exportations et importations. Les utilisations finales, ou utilisateurs finaux, pour des armes nucléaires, des missiles, des armes chimiques ou biologiques ou pour le nucléaire maritime, directement ou indirectement, sont strictement interdites. Les exportations ou réexportations vers des pays sous embargo des Etats-Unis, ou vers des entités figurant sur les listes d'exclusion d'exportation américaines, y compris, mais de manière non exclusive, la liste de personnes qui font 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 legislation américaine en matière de contrôle des exportations et la liste de ressortissants spécifiquement designés, sont rigoureusement interdites.

LA DOCUMENTATION EST FOURNIE "EN L'ETAT" ET TOUTES AUTRES CONDITIONS, DECLARATIONS ET GARANTIES EXPRESSES OU TACITES SONT FORMELLEMENT EXCLUES, DANS LA MESURE AUTORISEE PAR LA LOI APPLICABLE, Y COMPRIS NOTAMMENT TOUTE GARANTIE IMPLICITE RELATIVE A LA QUALITE MARCHANDE, A L'APTITUDE A UNE UTILISATION PARTICULIERE OU A L'ABSENCE DE CONTREFACON.

070213@16599

Contents

Preface

This *Sun Java System Message Queue 3.7 UR1 Installation Guide* explains how to install Sun Java System Message Queue 3.7 UR1.

Who Should Use This Book

This guide is intended for Message Queue developers and administrators. It contains fundamental information that users need to install the Message Queue product. Java Enterprise System users may also need to refer to this book for reference information about installed images.

Before You Read This Book

Before reading this book, you should read the *Message Queue Release Notes*, which describe new features and enhancements, known issues and limitations, and other information related to the current Message Queue release.

How This Book Is Organized

All readers should read the Introduction and then the appropriate platform-specific chapter. Table P–1 briefly describes the contents of each chapter.

TABLE P–1 Book Contents

Chapter	Description
Chapter 1	Describes Message Queue product editions, software modules, and installed directory structure, as well as compatibility issues for those upgrading from a previous Message Queue release.
Chapter 2	Describes the hardware and software requirements, defaults, and installation and deinstallation procedures for the Solaris platform.

TABLE P–1 Book Contents (Continued)

Chapter	Description
Chapter 3	Describes the hardware and software requirements, defaults, and installation and deinstallation procedures for the Linux platform.
Chapter 4	Describes the hardware and software requirements, defaults, and installation and deinstallation procedures for the Windows platform.

Related Documentation

The documents that comprise the Message Queue documentation set are listed in the following table in the order in which you would normally use them.

TABLE P–2 Message Queue Documentation Set

Document	Audience	Description
Sun Java System Message Queue 3.7 UR1 Installation Guide	Developers and administrators	Explains how to install Message Queue software on Solaris, Linux, and Windows platforms.
Sun Java System Message Queue 3.7 UR1 Release Notes	Developers and administrators	Includes descriptions of new features, limitations, and known bugs, as well as technical notes.
Sun Java System Message Queue 3.7 UR1 Technical Overview	Developers and administrators	Describes Message Queue concepts, features, and components.
Sun Java System Message Queue 3.7 UR1 Developer's Guide for Java Clients	Developers	Provides a quick-start tutorial and programming information for developers of Java client programs using the Message Queue implementation of the JMS or SOAP/JAXM APIs.
Sun Java System Message Queue 3.7 UR1 Administration Guide	Administrators, also recommended for developers	Provides background and information needed to perform administration tasks using Message Queue administration tools.
Sun Java System Message Queue 3.7 UR1 Developer's Guide for C Clients	Developers	Provides programming and reference documentation for developers of C client programs using the Message Queue C implementation of the JMS API (C-API).

Online Help

Message Queue 3.7 UR1 includes command-line utilities for performing Message Queue message service administration tasks.

Message Queue 3.7 UR1 also includes a graphical user interface (GUI) administration tool, the Administration Console (`imqadmin`). Context-sensitive help is included in the Administration Console; see "Administration Console Online Help" in *Sun Java System Message Queue 3.7 UR1 Administration Guide*.

JavaDoc

JMS and Message Queue API documentation in JavaDoc format is provided at the following location:

Platform	Location
Solaris	`/usr/share/javadoc/imq/index.html`
Linux	`/opt/sun/mq/javadoc/index.html`
Windows	`IMQ_HOME/javadoc/index.html`

This documentation can be viewed in any HTML browser. It includes standard JMS API documentation, as well as Message Queue-specific APIs for Message Queue administered objects, which are of value to developers of messaging applications.

Example Client Applications

Message Queue 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/sun/mq/examples`
Windows	`IMQ_HOME/demo/`

Example C Client Programs

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/sun/mq/examples/C/`
Windows	`IMQ_HOME/demo/C/`

The Java Message Service (JMS) Specification

The JMS specification can be found at the following location:

`(http://java.sun.com/products/jms/docs.html)`

The specification includes sample client code.

Directory Variable Conventions

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

Note – The information in Table P–3 applies only to the standalone installation of Message Queue. When Message Queue is installed and run as part of an Application Server installation, the values of the directory variables are set differently: `IMQ_HOME` is set to *appServer_install_dir*`/imq` (where *appServer_install_dir* is the Application Server installation directory), and `IMQ_VARHOME` is set to *appServer_domainName_dir*`/imq` (where *appServer_domainName_dir* is the domain directory for the domain starting the Message Queue broker).

TABLE P–3 Directory Variable Conventions

Variable	Description
`IMQ_HOME`	Used in Message Queue documentation to refer to the Message Queue base directory (root installation directory). ■ On Solaris and Linux, there is no root Message Queue installation directory. Therefore `IMQ_HOME` is not used in Message Queue documentation to refer to file locations in Solaris and Linux. ■ On Windows, the root Message Queue installation directory is set to the directory in which you unzip the Message Queue bundle.

TABLE P–3 Directory Variable Conventions *(Continued)*

Variable	Description
IMQ_VARHOME	The /var directory in which Message Queue 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 Java System Application Server, Evaluation Edition, IMQ_VARHOME defaults to the IMQ_HOME/var directory. ■ On Linux, IMQ_VARHOME defaults to the /var/opt/sun/mq directory. ■ On Windows, IMQ_VARHOME defaults to the IMQ_HOME/var directory.
IMQ_JAVAHOME	An environment variable that points to the location of the Java runtime (JRE) required by Message Queue executables. ■ On Solaris, Message Queue looks for the latest JRE, but a user can optionally set the value of IMQ_JAVAHOME to the location of the preferred JRE. ■ On Linux, Message Queue looks for the latest JRE, but a user can optionally set the value of IMQ_JAVAHOME to the location of the preferredJRE. ■ On Windows, IMQ_JAVAHOME is set to point to an existing Java runtime if a supported version is found on the system. If a supported version is not found, one will be installed.

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 (/).

Typographic Conventions

The following table describes the typographic conventions that are used in this book.

TABLE P–4 Typographic Conventions

Typeface	Meaning	Example
AaBbCc123	The names of commands, files, and directories, and onscreen computer output	Edit your .login file. Use ls -a to list all files. machine_name% you have mail.
AaBbCc123	What you type, contrasted with onscreen computer output	machine_name% **su** Password:

TABLE P–4 Typographic Conventions *(Continued)*

Typeface	Meaning	Example
aabbcc123	Placeholder: replace with a real name or value	The command to remove a file is `rm` *filename*.
AaBbCc123	Book titles, new terms, and terms to be emphasized	Read Chapter 6 in the *User's Guide*. A *cache* is a copy that is stored locally. Do *not* save the file. **Note:** Some emphasized items appear bold online.

Shell Prompts in Command Examples

The following table shows the default UNIX® system prompt and superuser prompt for the C shell, Bourne shell, Korn shell, and Windows operating system.

TABLE P–5 Shell Prompts

Shell	Prompt
C shell	`machine_name%`
C shell for superuser	`machine_name#`
Bourne shell and Korn shell	`$`
Bourne shell and Korn shell for superuser	`#`
Windows	C:\

Symbol Conventions

The following table explains symbols that might be used in this book.

TABLE P–6 Symbol Conventions

Symbol	Description	Example	Meaning
`[ ]`	Contains optional arguments and command options.	`ls [-l]`	The `-l` option is not required.

TABLE P–6 Symbol Conventions *(Continued)*

Symbol	Description	Example	Meaning
`{ | }`	Contains a set of choices for a required command option.	`-d {y|n}`	The `-d` option requires that you use either the `y` argument or the `n` argument.
`${ }`	Indicates a variable reference.	`${com.sun.javaRoot}`	References the value of the `com.sun.javaRoot` variable.
-	Joins simultaneous multiple keystrokes.	Control-A	Press the Control key while you press the A key.
+	Joins consecutive multiple keystrokes.	Ctrl+A+N	Press the Control key, release it, and then press the subsequent keys.
→	Indicates menu item selection in a graphical user interface.	File → New → Templates	From the File menu, choose New. From the New submenu, choose Templates.

Documentation, Support, and Training

The Sun web site provides information about the following additional resources:

- Documentation (`http://www.sun.com/documentation/`)
- Support (`http://www.sun.com/support/`)
- Training (`http://www.sun.com/training/`)

Searching Sun Product Documentation

Besides searching Sun product documentation from the docs.sun.com web site, you can use a search engine by typing the following syntax in the search field:

search-term `site:docs.sun.com`

For example, to search for "broker," type the following:

```
broker site:docs.sun.com
```

To include other Sun web sites in your search (for example, java.sun.com, www.sun.com, and developers.sun.com), use "`sun.com`" in place of "`docs.sun.com`" in the search field.

Third-Party Web Site References

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 or alleged to be caused by or in connection with use of or reliance on any such content, goods, or services that are available on or through such sites or resources.

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 full document title and part number. The part number is a 7-digit or 9-digit number that can be found on the book's title page or in the document's URL. For example, the part number of this book is 819-4466.

CHAPTER 1

Introduction

This chapter provides an overall introduction to installing the Sun Java System Message Queue 3.7 UR1 product. It covers the following topics:

- "Product Editions" on page 13
- "Supported Platforms and Components" on page 14
- "Message Queue Software Modules" on page 16
- "Installing from the Web" on page 16
- "Migration Issues" on page 16

Product Editions

Prior to the release of Version 3.7 UR1, the Sun Java System Message Queue product was available in two editions, Platform and Enterprise, each containing different features and corresponding to a different licensed capacity. Version 3.7 UR1 combines the functionality of both editions.

- If you have a previous version of Message Queue Enterprise Edition installed, you can upgrade to Version 3.7 UR1 with no loss of functionality.
- If you have a previous version of Message Queue Platform Edition installed, you can upgrade to Version 3.7 UR1 and enjoy access to a richer set of features, including:
  - Support for multiple-broker message services
  - HTTP/HTTPS connections
  - Secure connection services
  - Scalable connection capability
  - Queue delivery to more than three consumers
  - C client support

Supported Platforms and Components

Message Queue 3.7 UR1 is supported on Solaris, Linux, and Windows operating systems and platforms. The following chapters describe the hardware and software requirements for each of these platforms and provide platform-specific installation instructions.

In addition to platform-specific requirements, Message Queue 3.7 UR1 also depends on certain basic components that you must install in order to be able to develop and run Message Queue clients. Table 1–1 describes these components. Other versions or vendor implementations can also be used, but they are untested by Sun Microsystems and therefore not supported.

TABLE 1–1 Required Support Components

Component	Supports	Supported Versions
Java Runtime Environment (JRE) (Sun Microsystems versions only)	Message Queue broker and administration tools	Java Runtime Environment 1.5.0_09 Java 2 Platform, Standard Edition, 5.0 (1.5.0_09): ■ Solaris 9 (SPARC and X86) ■ Solaris 10 (SPARC and X86) ■ Linux Red Hat Advanced Server 3.0, Update 6 ■ Linux Red Hat Advanced Server 4.0, Update 2 ■ Windows XP Professional SP2 (design-time and client support only) ■ Windows 2000 Server and Advanced Server with Service Pack 4 ■ Windows Server 2003 Enterprise Edition
Java Software Development Kit (JDK), Standard Edition (Sun Microsystems production versions only)	Java client development and deployment	JDK 1.4.2_10 Java 2 Platform, Standard Edition, 5.0 (1.5.0_09): Same platforms as for JRE, above

Table 1–2 shows additional components that you can install to provide further support for a Message Queue client. You may not need all of the components listed: for example, if you are not writing a C Message Queue client, you will not need the C compiler, C++ runtime library, NSPR, or NSS. .

TABLE 1–2 Optional Support Components

Component	Supports	Supported Versions
LDAP (Lightweight Directory Access Protocol) Directory Server	Message Queue user repository and administered objects	Sun Java System Directory Server, Version 6.0
Web Server	HTTP/HTTPS	Sun Java System Web Server, Enterprise Edition Version 7.0
Application Server	HTTP/HTTPS	Sun Java System Application Server, Enterprise Edition Version 8.2
Database	Plugged-in persistence	Apache Derby, Version 10.1.1 Oracle 9i, Version 9.2 The Pointbase database is no longer supported
JNDI (Java Naming and Directory Interface)	Administered object support and LDAP user repository	JNDI Version 1.2.1 LDAP Service Provider, Version 1.2.2 File System Service Provider, Version 1.2 Beta 3 (administered object support only; supported for development and testing, but not for deployment in a production environment)
C Compiler and compatible C++ runtime library	Message Queue C clients	**Solaris:** Sun Studio, Version 10 or later, C++ compiler with standard mode and C compiler **Linux:** gcc/g++, Version 3.2.3 **Windows:** Microsoft Windows Visual C++, Version 6.0 SP3
NSPR (Netscape Portable Runtime)	Message Queue C clients	Version bundled with Java Enterprise System 5 Bundled as a shared package in the download bundle
NSS (Network Security Service)	Message Queue C clients	Version bundled with Java Enterprise System 5 Bundled as a shared package in the download bundle

Message Queue Software Modules

Table 1–3 shows the full set of software modules included with the Message Queue 3.7 UR1 product.

TABLE 1–3 Message Queue Software Modules

Module	Contents
Broker	Server-side software for routing and delivering messages. Requires the Java runtime module.
Administration tools	Command-line utilities and GUI tools for administering a Message Queue messaging system. Requires the client runtime and Java runtime modules.
Java client runtime	`.jar` files needed to write and compile Java clients using the Message Queue Java application programming interface (API).
C client runtime	Libraries and header files needed to write and compile C clients using the Message Queue C application programming interface (API).
Documentation	API documentation needed by Java client application developers in JavaDoc format.
Example applications	Example client applications.
JDK runtime	JDKRuntime Environment (Windows only).
License	Software needed to enable the licensed capacity of a Message Queue messaging system.

Installing from the Web

You can download Message Queue 3.7 UR1 from the product Web site at

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

See the subsequent chapters for platform-specific installation instructions and descriptions of the installed directory structures.

Migration Issues

This section covers issues you need to be aware of when migrating to Message Queue 3.7 UR1 from earlier versions of Message Queue. These issues fall into two general categories: platform issues and compatibility issues.

Note – Migration from Message Queue versions earlier than 3.0.1 is not supported.

Platform Issues

This section describes issues specific to the Solaris, Linux, and Windows platforms.

Solaris

On the Solaris platform, you can install Message Queue 3.7 UR1 on top of Versions 3.0.*x*, 3.5, and 3.6, and your existing instance data (configuration properties, file-based persistent data store, log files, flat-file user repository, and access control properties file) will be used by Version 3.7 UR1 (see "Compatibility Issues" on page 18).

If you depended on the `.jar` files listed below to be in their Version 3.0.*x* locations, please note that they have been moved to the `/usr/share/lib` directory:

```
activation.jar
commons-logging.jar
fscontext.jar
imq.jar
imqxm.jar
jaxm-api.jar
jms.jar
mail.jar
saaj-api.jar
saaj-impl.jar
```

In Message Queue 3.0, these files were in the `/usr/share/lib/imq` directory; in Message Queue 3.0.1, they were in the `/usr/share/lib` directory with symbolic links in the `/usr/share/lib/imq` directory. In Message Queue 3.7 UR1, there are no symbolic links files.

Linux

On the Linux platform, you should *not* try to install Message Queue 3.7 UR1 on top of earlier versions, but should instead uninstall any earlier versions of Message Queue before installing Version 3.7 UR1. This is because the installed directory structure has been changed as of Message Queue 3.6, and this complicates the migration of instance data (configuration properties, file-based persistent data store, log files, flat-file user repository, and access control properties file) from Versions 3.0.*x* and 3.5 to Version 3.7 UR1. A utility program, `mqmigrate`, is provided to ease the migration of these data files from versions earlier than 3.6 (see "Finding and Removing Earlier Message Queue Versions" on page 40). (If you are upgrading from Message Queue 3.6, you do not need to use the `mqmigrate` utility, since the files are already in the correct locations.) After migrating the instance data files if necessary, you should uninstall any Message Queue 3.0.*x*, 3.5, and 3.6 Red Hat Package Manager (RPM) packages and then install the 3.7 UR1 RPM packages using `mqinstall`.

Scripts that use hard-coded paths to the Linux installations of Message Queue Versions 3.0.*x* or 3.5 will need to be changed to the new directory location for Message Queue 3.7 UR1. If you need symbolic links that point the old locations to the new locations, you must install the `sun-mq-compat` RPM package; this package may not be supported in future releases.

Windows

The upgrade procedure for Message Queue 3.7 UR1on the Windows platform varies depending on the previously installed version of Message Queue, as described in the following sections.

Message Queue 3.5 and 3.6

Before installingMessage Queue 3.7 UR1, you should uninstall any existing installations of Message Queue 3.5 or 3.6. If you are installing Version 3.7 UR1 in a different location than Versions 3.5 or 3.6, you must first move any instance data (configuration properties, file-based persistent data store, log files, flat-file user repository, and access control properties file) that you want to preserve from its old location, corresponding to the old value of the directory variable `IMQ_VARHOME/instances/`, to the new location, corresponding to the new value of `IMQ_VARHOME/instances`.

Message Queue 3.0.*x*

If you are upgrading from Message Queue 3.0.*x* to 3.7 UR1, it is likely that you will have to manually migrate any previous instance data (configuration properties, file-based persistent data store, log files, flat-file user repository, and access control properties file) from its Message Queue 3.0.*x* location to the new Message Queue 3.7 UR1 location, specified as follows:

`IMQ_VARHOME\instances\`*instanceName*

Compatibility Issues

Message Queue 3.7 UR1 is generally compatible with Message Queue 3.0.*x*, 3.5, and 3.6. However, changes have been made in broker properties, administered objects, persistence schemas, file locations, and administration tools that can affect migration from the earlier versions to Version 3.7 UR1.

The Message Queue 3.7 UR1 install operation does not remove or overwrite the Message Queue 3.0.*x*, 3.5, or 3.6 `IMQ_VARHOME` directory. This directory contains configuration and security-related files (see “Broker Compatibility” on page 19). Most of this data is compatible with Message Queue 3.7 UR1, and can be preserved using the instructions in the following sections.

Compatibility issues that you may need to address when migrating from Message Queue 3.0.*x*, 3.5, or 3.6 to Message Queue 3.7 UR1 include the following:

- “Broker Compatibility” on page 19
- “Administered Object Compatibility” on page 21

- "Administration Tool Compatibility" on page 21
- "Client Compatibility" on page 20

Broker Compatibility

A Message Queue 3.7 UR1 message broker will interoperate with one from Versions 3.0.*x*, 3.5, or 3.6. However, some changes have been made in broker properties and the persistent store schema. The degree of compatibility depends on the earlier Message Queue version from which you are upgrading:

- Message Queue 3.6 data is generally compatible with Message Queue 3.7 UR1 and requires no migration.
- Message Queue 3.5 data is generally compatible with Message Queue 3.7 UR1 and can be used after migration to Message Queue 3.7 UR1.
- Some Message Queue 3.0.x data is compatible with Message Queue 3.7 UR1, as described below, and can be used after migration to Message Queue 3.7 UR1.

When upgrading from Message Queue 3.0.*x*, 3.5, or 3.6 to Message Queue 3.7 UR1, you can choose either of two approaches to handling the older versions' instance configuration (`config.properties`) files:

- Use the older `config.properties` files directly.
- Copy the files to another location and consult the property settings they contain when you configure Message Queue 3.7 UR1 brokers.

Any persistent Message Queue 3.0.*x*, 3.5, or 3.6 data (messages, destinations, and durable subscriptions) is automatically converted to Message Queue 3.7 UR1 data when starting up a Message Queue 3.7 UR1 broker for the first time. For example, existing destinations will be converted to Message Queue 3.7 UR1 destinations, preserving existing attributes and using default values for new attributes.

For Message Queue 3.0.*x*, the automatic migration of persistent data leaves the original data intact in its original location. The migrated data, however,can no longer be used by a Message Queue 3.0.*x* broker. If this is a concern, save the original data to a secure location after migration. Otherwise, you can delete this data in any of the following ways:

- Use the Broker utility's `-upgrade-store-nobackup` option when starting up the Message Queue 3.7 UR1 broker for the first time:

  ```
  imqbrokerd -upgrade-store-nobackup
  ```

- Delete the old persistent data store manually:
  - For a file-based data store, delete the old persistent store, located at

    …/instances/*instanceName*/filestore/

  - For a JDBC-based data store, delete the old tables using the command

    ```
    imqdbmgr delete oldtbl
    ```

- Note: Applies to Message Queue 3.0.*x* only. You can continue to use the Message Queue 3.0.*x* user repository and access control properties files after installing Message Queue 3.7 UR1, however these files have been made instance-specific and are now placed in the `.../instances/`*instanceName*`/etc` directories. The Message Queue 3.7 UR1 installer does not overwrite the Message Queue 3.0.*x* files. When you first start up a Message Queue 3.7 UR1 broker, copies of the Message Queue 3.0.*x* files are placed in the appropriate Message Queue 3.7 UR1 location (see Appendix A, "Platform-Specific Locations of Message Queue Data," in *Sun Java System Message Queue 3.7 UR1 Administration Guide*). If Message Queue 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 Message Queue 3.0.*x*, 3.5, or 3.6 brokers and Message Queue 3.7 UR1 brokers in a cluster, the master broker must be a Message Queue 3.0.*x*, 3.5, or 3.6 broker (whichever is older), and the cluster will run as a Message Queue 3.0.*x*, 3.5, or 3.6 cluster.

TABLE 1–4 Compatibility of Message Queue 3.0.*x* Data with Message Queue 3.7 UR1

Data Category	Message Queue 3.0.*x* Location	Message Queue 3.7 UR1 Compatibility
Broker properties	`IMQ_VARHOME/instances/`*instanceName*`/props/config.properties`	New properties added Some property names changed Old property names still recognized
Persistent data store (messages, destinations, durable subscriptions)	`IMQ_VARHOME/instances/`*instanceName*`/filestore/` or JDBC-accessible data store	Converted to Message Queue 3.7 UR1 format at `IMQ_VARHOME/`*instanceName*`/fs370` when Message Queue 3.7 UR1 broker started for the first time
Flat-file user repository	**Solaris:** `/etc/imq/passwd` **Linux:** `IMQ_HOME/etc/passwd` **Windows:** `IMQ_HOME\etc\passwd`	Compatible Automatically copied to `/instances/`*instName*`/etc/passwd` when Message Queue 3.7 UR1 broker started for the first time
Access control properties file	**Solaris:** `/etc/imq/accesscontrol.properties` **Linux:** `IMQ_HOME/etc/accesscontrol.properties` **Windows:** `IMQ_HOME\etc\accesscontrol.properties`	Compatible Automatically copied to `/instances/`*instName*`/etc/accesscontrol.properties` when Message Queue 3.7 UR1 broker started for the first time

Client Compatibility

Message Queue 3.7 UR1 message brokers and Java clients (applications and components) are two-way compatible with those from Message Queue 3.0.*x*, 3.5, or 3.6: that is, a Message Queue

3.7 UR1 broker will support a Message Queue 3.0.*x*, 3.5, or 3.6 Java client, and a Message Queue 3.7 UR1 Java client can connect to a Message Queue 3.0*x*, 3.5, or 3.6 broker. Expanded Message Queue 3.7 UR1 capabilities are unavailable in such cases, however.

Note – C client programs cannot connect to a Message Queue 3.0.*x* broker; they are supported only by Message Queue 3.7 UR1 brokers or by Message Queue 3.5 and 3.6 brokers running with a trial or enterprise license.

Administered Object Compatibility

Message Queue 3.7 UR1 administered objects have been enhanced with new attributes, and some attributes have been renamed from earlier versions. Although you can continue to use object stores and administered objects created in Message Queue 3.0.*x*, 3.5, or 3.6, it is best to upgrade your administered objects after installing Message Queue 3.7 UR1.

When performing an update operation, the Administration Console (`imqadmin`) and the ObjectManager command line utility (`imqobjmgr`) will automatically convert administered objects from Versions 3.0.*x*, 3.5, and 3.6 into Message Queue 3.7 UR1 form, using default values for the new attributes. The Message Queue 3.7 UR1 client runtime will also look up and instantiate administered objects from those earlier versions and convert them for use by Message Queue 3.7 UR1 clients, but this will *not* convert the objects permanently in the object store in which they reside.

Existing Message Queue clients that directly instantiate administered objects are compatible with Message Queue 3.7 UR1. For attributes that have been renamed in Message Queue 3.7 UR1, the old names will still work. (Recompiling the client with Message Queue 3.7 UR1 will show which attributes have been renamed.) However, clients will need to be rewritten if they are to use any of the new administered object attributes. Similarly, scripts that start Java clients and set administered object attribute values using command line options are compatible with Message Queue 3.7 UR1,but must be rewritten in order to use the new attributes.

Administration Tool Compatibility

Because of the addition of new commands and new administrative capabilities, the Message Queue 3.7 UR1 administration tools (Administration Console and command line utilities) will work only with Message Queue 3.7 UR1 brokers. All Message Queue 3.0.*x*, 3.5, and 3.6 commands and command options remain supported, however.

CHAPTER 2

Solaris Installation

This chapter covers the following topics as they apply to a Solaris installation of Message Queue 3.7 UR1:

Hardware and Software Requirements

In order to install Message Queue 3.7 UR1, your Solaris system (SPARC or x86) should satisfy the minimum hardware and software requirements shown in Table 2–1.

TABLE 2–1 Hardware and Software Requirements (Solaris)

Component	Requirements
CPU	Sun UltraSPARC Intel Pentium 2 (or compatible)
RAM	256 MB

TABLE 2–1 Hardware and Software Requirements (Solaris) *(Continued)*

Component	Requirements
Disk Space	**SPARC platform:** ■ Compressed installation file: Approximately 40 MB ■ Temporary working directory (for extracting installation files): Additional 34 MB ■ Installed product: Approximately 34 MB **x86 platform:** ■ Compressed installation file: Approximately 26 MB ■ Temporary working directory (for extracting installation files): Additional 31 MB ■ Installed product: Approximately 31 MB **Note –** The installed product may require more space if the broker stores persistent messages locally.
Operating System	Solaris 9 Solaris 10 **Note –** To ensure proper operation of Message Queue 3.7 UR1, you should install all required Solaris Patches for Java 2 Platform, Standard Edition 5.0. For the latest information about required and recommended patches and to download the patches, see `http://java.sun.com/j2se/1.5.0/download.jsp`
Java 2 Platform, Standard Edition (J2SE)	See "Supported Platforms and Components" on page 14 for supported versions of the Java Runtime Environment (JRE) and Java Software Development Kit (JDK). **Note –** The Message Queue 3.7 UR1 software distribution includes the required JRE version at the time of release.

Installing Message Queue

The Message Queue 3.7 UR1 product can be downloaded from the Sun Java System Web site. Message Queue also depends on components that you must install in order to develop and run Message Queue clients. See Table 1–1 and Table 1–2 for more information.

Finding Earlier Message Queue Versions

Because Message Queue is installed with other products (such as Solaris 9, Solaris 10, and Sun Java System Application Server), you should check whether Message Queue has already been installed on your system. To do so, enter the following command:

```
imqbrokerd -version
```

If Message Queue is already installed, the version number is displayed.

If you find that a previous installation already exists on your system, see "Upgrading From Previous Versions" on page 25for information on upgrading to Message Queue 3.7 UR1.

Upgrading From Previous Versions

To upgrade from Message Queue Versions 3.0.*x*, 3.5, and 3.6, you need not uninstall those versions; they will be overwritten.

Note – If you want to upgrade to Message Queue 3.7 UR1, you must purchase Message Queue 3.7 UR1 and use the Sun Java Enterprise System installer to upgrade your version of Message Queue. See the *Sun Java Enterprise System Installation Guide* for more information.

Installed Directory Structure

Table 2–2 shows the installed directory structure for a full installation of Message Queue 3.7 UR1 on the Solaris platform. (The directory structure may vary if you perform a partial installation.)

Note – File locations for Message Queue bundled with Sun Java System Application Server may differ from those shown in the table.

TABLE 2–2 Installed Directory Structure (Solaris)

Directory	Contents
`COPYRIGHT` (not installed)	Copyright text file
`DISTRIBUTIONREADME`	List of redistributable files
`ENTITLEMENT`	Evidence of purchase
`LICENSE` (not installed)	License text file
`README` (not installed)	`README` text file
`THIRDPARTYLICENSECREADME`	NSS/NSPR license

TABLE 2–2 Installed Directory Structure (Solaris) *(Continued)*

Directory	Contents
`/usr/bin`	Executable files for the following administration tools: - Administration Console (`imqadmin`) - Broker utility (`imqbrokerd`) - Command utility (`imqcmd`) - Object Manager utility (`imqobjmgr`) - Database Manager utility (`imqdbmgr`) - User Manager utility (`imqusermgr`) - Key Tool utility (`imqkeytool`)
`/usr/share/lib`	Support files for Message Queue Java client runtime: - `.jar` files for building and running Java Message Service (JMS) client applications - `.rar` files for JMS Resource Adapter
`/usr/share/lib/imq`	Support files for Message Queue tools and processes: - `.jar` files used by Message Queue system - `.war` files for HTTP servlet deployment
`/usr/share/lib/imq/props`	Broker's default configuration files
`/usr/share/lib/imq/ext`	Files needed for JDBC-based persistence
`/usr/share/lib/imq/help`	Message Queue help files
`/usr/share/lib/imq/images`	Admin GUI image files
`/usr/share/javadoc/imq`	Message Queue and JMS API documentation in JavaDoc format
`/usr/demo/imq`	Example Java client applications
`/opt/SUNWimq/demo/C`	Example C client applications
`/opt/SUNWimq/include`	Header files to support C client applications
`/opt/SUNWimq/lib`	Libraries to support C client applications **Note –** The versions of Netscape Portable Runtime (NSPR) and Network Security Service (NSS) needed to support the C API are the same as those for Java Enterprise System 5.
`/var/imq`	Message Queue working storage
`/var/imq/instances`	Configuration properties, file-based persistent data stores, log files, flat-file user repositories, and access control properties files for individual broker instances
`/etc/imq`	License files, instance template files, and `rc` script configuration files for automatic startup

Working with Solaris 10 Zones

A *zone* is a Solaris Container technology that provides separate environments on a machine and logically isolates applications from one another. Zones allow you to create virtual operating system environments within an instance of the Solaris operating system. Running applications in different zones allows you to run different instances or different versions of the same application on the same machine while, at the same time, permitting centralized administration and efficient sharing of resources.

This section provides a brief description of zones and describes their use with Message Queue 3.7 UR1.

Zones Basics

A zone environment includes a global zone and one or more non-global zones. When Solaris 10 is first installed on a system there is only one global zone. An administrator can create other non-global zones as children of the global zone. Each zone appears as an independent system running Solaris. Each zone has its own IP address, own system configuration, own instances of running applications, and its own area on the file system.

The global zone contains resources that can be shared among non global zones; this allows the centralization of certain administrative functions. For example, packages installed in the global zone are available (propagated) to all existing non-global zones. This enables you to centralize life-cycle management like installation, upgrade, and uninstallation. At the same time, the isolation provided by non-global zones results in greater security and allows you to have differently configured instances or different versions of the same application running on the same machine.

Non-global zones are either whole root zones or sparse root zones: which of these you choose as an environment for an application depends on how you want to balance administrative control with resource optimization.

- *Whole root zones* contain a read/write copy of the file system on the global zone. Packages installed in the global zone are automatically copied (with their registry information) to the whole root zones. This maximizes administrative control, at the expense of resources.
- *Sparse root zones* contain a read/write copy of a portion of the file system on the global zone; other file systems are mounted as read-only file systems. Packages installed in the global zone are available to sparse root zones by means of read-only file systems and through the automatic synchronization of registry information. Sparse root zones optimize resource sharing at the cost of centralized administration.

Java Enterprise System Zones Limitations

The components that make up the Java Enterprise System depend on some shared components; this creates some limitations in working with zones. In a zones environment, shared components are governed by the following rules.

- All shared components within a zone must be of the same JES version. This requirement has three consequences.
  - If you want to install different versions of shared components, each version must reside in a separate zone.
  - Within a zone, if a shared component is upgraded or a later version is installed then all shared components must be upgraded.
  - When you install shared components in the global zone, you must take care that shared components in non global zones are upgraded if necessary.
- Shared components cannot be installed in sparse root zones because of the read/only file system in sparse root zones. Instead, they must be installed in the global zone. Those product components that depend on shared components must first be installed in the global zone and then propagated into non-global zones.

These requirements affect the installation of Message Queue because it is a component product of Java Enterprise System and, as such, is limited in its use of zones.

Note – The Message Queue product is installed into the `/usr` directory and must therefore be installed or upgraded in the global zone first.

Message Queue Cases

When Message Queue is installed in the global zone, it is set to propagate into all of the non-global zones. After installing Message Queue in the global zone, you will have the same version of Message Queue installed in all zones: if you log into any zone and run the command `pkginfo -l SUNWiqu`, you will see it installed, and it will be the same version as in the global zone. You can then run independent instances of the Message Queue broker in each zone since they do not share the instance and configuration data kept in `/var` and `/etc`. Most other Java Enterprise System components are not propagated if they are installed in the global zone.

Because Message Queue is propagated into non-global zones, the global instance is forever linked to the installations in the non-global zones. Therefore, any time you uninstall or upgrade Message Queue in the global zone it will impact instances running in the non-global zones. The following example shows how this might cause unintended results.

1. You install Message Queue 3.7 UR1 in the global zone. This results in the Message Queue 3.7 UR1 packages also being installed into all non-global zones.
2. You uninstall Message Queue 3.7 UR1 in a whole root zone. Then, you install Message Queue 3.6 in the whole root zone.

   You now have different versions of Message Queue running in different zones, which is a set up you might find useful.
3. You uninstall Message Queue 3.7 UR1 in the global zone. This will uninstall Message Queue from all other zones - including the Message Queue 3.6 instance in the whole root zone.

Always be aware of the cascading effect of installing or uninstalling Message Queue in the global zone.

The following two use-cases explain how you install different instances and different versions of Message Queue in different zones.

Note – If you want to install Message Queue in a whole root zone on Solaris 10, Solaris 10U1, or Solaris 10U2, you must upgrade Lockhart in the global zone first. See the workaround for bug 645030 for additional information.

▼ To Install the Same Version of Message Queue in Different Zones

1. **Install the desired version of Message Queue in the global zone.**

   These versions will be propagated into any existing non-global zone. If you create additional non-global zones, Message Queue will also be propagated into these zones. Please note that you can install different instances in whole root zones as well as sparse root zones, but using sparse root zones allows you to make more efficient use of disk space and other resources.

2. **If you want Message Queue to be propagated into any other non-global zones, create these zones now.**

3. **Run an instance of Message Queue in each non-global zone.**

▼ To Install Different Versions of Message Queue in Different Zones

1. **Uninstall Message Queue from the global zone.**

2. **Create whole root zones and configure each zone not to share the /usr directory by using the following directive when you create the zone**

   ```
   remove inherit-pkg-dir dir=/usr
   ```

3. **Install different versions of Message Queue in each whole root zone.**

Note – Remember that installing or uninstalling Message Queue in the global zone will affect all instances (and versions) of Message Queue running in whole root zones.

Message Queue Solaris Packages

The following table describes the Message Queue Solaris packages, and Table 2–4 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 Message Queue. If it is, you should do a custom install.

TABLE 2–3 Packages in Solaris Bundle

#	Package	Description	Note
1	SUNWiqcdv	C header files and demo files	Required for developing C client programs.
2	SUNWiqcrt	C client shared libraries	Required for running C client programs.
3	SUNWiqdoc	Message Queue Java client API JavaDoc and example applications	Needed for Java client development.
4	SUNWiqfs	Message Queue JNDI File System Service Provider	Required for client development and administration tools that use the JNDI FIle System Service Provider. The JNDI Service Provider is not supported for deployment.
5	SUNWiqjx	Message Queue Java API for XML Messaging (JAXM): API and runtime	Required to support Java clients using SOAP/JAXM API.
6	SUNWiqlen	Legacy package containing Enterprise license file.	No longer used.
7	SUNWiqlpl	Legacy package containing try license	No longer used.
8	SUNWiqr	Message Queue message server root package	Needed for Message Queue executables.
9	SUNWiqu	Message Queue broker and administration tools.	
10	SUNWiquc	Message Queue Java API for JMS messaging and Java client runtime.	Required to support Java clients using JMS API.
11	SUNWiqum	Message Queue JMS/SOAP Message Transformer: API and runtime	Required to perform conversions between SOAP messages and JMS messages.
12	SUNWjaf[1]	JavaBeans Activation Framework: API and runtime	Required to support Java clients using SOAP/JAXM API.
13	SUNWjaxp[1]	Java API for XML processing	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.

[1] This package is shared with a number of Sun Java System products. This package is not installed by default. You must manually install any shared packages after installing Message Queue packages.

TABLE 2–3 Packages in Solaris Bundle *(Continued)*

#	Package	Description	Note
15	SUNWjhdev[1]	JavaHelp Development Utilities	Required to upgrade the current JavaHelp runtime package (SUNWjhrt).
16	SUNWjhdoc[1]	JavaHelp Documentation	Required to upgrade the current JavaHelp runtime package (SUNWjhrt).
17	SUNWjhdem[1]	JavaHelp Demos	Required to upgrade the current JavaHelp runtime package (SUNWjhrt).
18	SUNWjmail[1]	JavaMail: API and runtime	Required to support Java clients using SOAP/JAXM API.
19	SUNWpr[1]	Netscape Portable Runtime (NSPR) libraries	Needed to support C clients.
20	SUNWprx[1]	Netscape Portable Runtime (NSPR) libraries	Needed to support C clients. (64-bit) (SPARC only).
21	SUNWtls[1]	Network Security Services (NSS) libraries	Needed to support C clients.
22	SUNWtlsx[1]	Network Security Services (NSS) libraries	Needed to support C clients. (64-bit SPARC only).
23	SUNWtlsu[1]	Network Security Services Utilities	Programs Needed to support SSL for C clients.
24	SUNWtlsux[1]	Network Security Services Utilities Programs	Needed to support SSL for C clients. (64-bit SPARC only).
25	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 with a number of Sun Java System products. This package is not installed by default. You must manually install any shared packages after installing Message Queue packages.

The following table provides a guide to the packages you need for different use scenarios.

TABLE 2–4 Packages Required for Various Scenarios

Scenario	Packages Needed	Notes

Message Queue message server and administration tools	`SUNWiqr`
	`SUNWiqu`
	`SUNWiqlpl`
	`SUNWiquc`
	`SUNWjhrt`
	`SUNWiqfs` (optional)
		Required for a Message Queue broker to run on a host.

Developing and/or deploying Java clients using the JMS API	`SUNWiquc` `SUNWiqdoc` (optional)	Can be installed on a system without a Message Queue broker.

Developing and/or deploying Java clients using the SOAP/JAXM API	`SUNWjaf`
	`SUNWjmail`
	`SUNWiqjx`
	`SUNWxsrt`
	`SUNWjaxp`
	`SUNWiqdoc` (optional)
		Can be installed on a system without a Message Queue broker. 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 Message Queue broker. The Message Queue Message Transformer API depends on both the JMS and SOAP APIs.

Developing and/or deploying C clients	`SUNWiqcrt`
	`SUNWiqcdv`
	`SUNWpr`
	`SUNWprx`
	`SUNWtls`
	`SUNWtlsx`
	`SUNWtlsu`
	`SUNWtlsux` (for SSL)
		The `SUNWprx`, `SUNWtlsx`, and `SUNWtlsux` packages on the Solaris SPARC platform are legacy packages and are no longer used. The `SUNWtlsu` and `SUNWtlsux` packages are used to create and manage NSS certificate database files by a C client when using SSL.

Installation Procedure

The following instructions explain how to download and install the Message Queue product on Solaris from the Sun Java System Web site.

▼ To Install Message Queue on Solaris

1 **Read the product license. Installation and use of the product are subject to acceptance of the license agreement.**

2 **Download the Message Queue product distribution file from the Web site into an empty, temporary working directory,** *temp_directory***.**

The zipped distribution file name depends on the Message Queue hardware platform:

Edition	SPARC	x86
Platform	`mq3_7-ent-solsparc.zip`	`mq3_7-ent-soli386.zip`

3 **Change to a temporary directory.**

`cd` *temp_directory*

4 **Unzip the distribution file.**

`unzip mq3_7-ent-`*platform*`.zip`

where *platform* is either `solsparc` or `soli386`, depending on the platform.

The `unzip` command creates an `mq3_7-ent` directory which contains the distribution files: `LICENSE`, `ENTITLEMENT`, `DISTRIBUTIONREADME`, `THIRDPARTYLICENSECREADME`, `README`, and `COPYRIGHT` files; install and uninstall scripts; and a `pkgs` directory that contains the Message Queue packages, as well as shared Solaris packages that have been updated for use with Solaris 9 (`SunOS5.9`).

Table 2–3 describes the Message Queue packages, and Table 2–4 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 Message Queue. If it is, you should do a custom install.

5 **Change to the directory containing the Message Queue distribution files.**

```
cd mq3_7-ent
```

6 **Become root.**

```
su root
```

When prompted, type your root password.

7 **Run the mqinstall script.**

This script will overwrite all of the Message Queue-specific packages listed in table Table 2–3

```
./mqinstall
```

The script lists the distribution packages, if any, that are already installed, and then lists the packages about to be installed. Please note that the install script will not install any shared packages (packages that do not begin with `SUNWiq` and that might already be installed on your system). You must install shared packages manually, as described in step 9.

8 **Enter `y` (yes) if you want to install all the packages. If you want to install the packages manually, enter `n` (no).**

If you run the `mqinstall` script, it creates a log file in the `/var/sadm/install/logs/` directory.

9 **Check your system for patches to any of the shared packages listed in table Table 2–3 (packages that do not begin with `SUNWiq` and that might already be installed on your system). Look in the following directory to see what versions will be installed:**

./pkgs/Solaris_*versNo*/*pkg_name*/pkginfo

Then, find the current installed version of the shared component using a command like the following:

% pkgparam -v *pkg_name* | grep VERSION

or

% pkgparam -v *pkg_name* | grep SUNW_PRODVERS

- If there is no package already installed, then you can go ahead and install the new one.
- If an existing package is older than the new one, remove the older one using the `pkgrm` command. For example,

  ```
  % pkgrm SUNWpr
  ```

  Then, add the new packages as shown in Step 10.
- If the package version is the same, don't add or remove anything.

10 **If you want to install a subset of the packages listed in table Table 2–3, if you want to install shared packages, or if you do not want to overwrite later versions of packages, do the following:**

a. **Change to the `pkgs` directory**

```
cd pkgs
```

b. **Run the pkgadd command to install the packages you want. Note the special directions below for upgrading shared components.**

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

The pkgadd utility lists the names of all packages in the directory available for installation (see Table 2–3). When prompted, indicate the packages you want to install, by entering the number of the package. To select multiple packages, enter a list of numbers separated by a comma.

(The-a admin.conf option permits an overwrite of any packages that are already installed on your system.) The pkgadd utility installs the packages you specify, possibly asking for additional information, and eventually returns to the original prompt, displaying the list of packages available for installation. Please note that running the pkgadd command from the ./pkgs directory with the default selection (all), will install all the Message Queue packages as well, which are already installed by the mqinstall script

---

Note – To upgrade shared components (indicated by a package name that does not start with SUNWiq), you must also run the command pkgadd -d from the ./pkgs/Solaris_9 directory or the ./pkgs/Solaris_10 directory, depending on the operating system where you are installing the product. (This directory contains operating-system-specific shared components only). For example, the following command adds the shared components that are needed in Solaris 10.

```
cd pkgs/Solaris_10
pkgadd -d ./ -a ../admin.conf
```

---

c. **Type q to quit.**

11 **Exit the root shell.**

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

This is your logical media. You will need it to uninstall or reinstall Message Queue. Treat this file as you would any other installation media and place a copy in a safe location.

Note – The instance data for any pre-existing broker instance (including the default broker instance, named imqbroker) is owned by the user that created that instance. Therefore, once installation is complete, be sure to run any Message Queue broker instance as the owner with privileges to the /var/imq/instances/*instanceName* directory. You become the owner by logging in as that user.

Checking Your Installation

To check that the expected version of Message Queue is running on your system, navigate to the Message Queue directory and enter the command: imqbrokerd -version. The output to this command specifies the version of the JDK and Message Queue that are installed on your system.

Configuring Message Queue for Automatic Startup

If you want to set the broker (the Message Queue message server) for automatic startup, you need to become root and to edit the configuration file `/etc/imq/imqbrokerd.conf`. The startup properties you can set in this configuration file are shown in the following table.

TABLE 2–5 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 "Command Utility" in *Sun Java System Message Queue 3.7 UR1 Administration Guide*for a listing and description of `imqbrokerd` command line options. (For example `-name` *instanceName*.)
`RESTART`	Specifies (*YES/NO*) if the broker is automatically restarted if it abnormally exits. Default: *YES*

To check that startup changes are correct (without booting the system), you can, as root, explicitly run the Message Queue initialization script in debug mode by using the following command:

```
env DEBUG=1 /etc/init.d/imq start
```

Configuring the Java Runtime

At startup, the Message Queue broker checks to make sure it has access to the required Java runtime version (JDK/JRE 1.5). 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 you specify both, 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. The broker tries to locate the latest JDK installed on the system.

To check which version the broker is using, you can start the broker with the following command: `imqbrokerd -verbose`

Uninstalling Message Queue

The following instructions explain how to uninstall Message Queue.

▼ To Remove Message Queue on Solaris

1. **Stop any running client applications.**

2. **Stop any running brokers. You will be prompted for user name (admin user) and password.**

   ```
   imqcmd shutdown bkr [-b hostName:port]
   ```

3. **If you want to delete dynamic data, the Message Queue flat-file user repository, and the Message Queue access control properties file associated with each broker instance, remove this data using the following command.**

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

4. **Find the zipped distribution file used to install Message Queue and place it in a temporary directory,** *temp_directory***.**

5. **Change directories to** *temp_directory***:**

   ```
   cd temp_directory
   ```

6. **Unzip the distribution file.**

   ```
   unzip mq3_7-ent-platform.zip
   ```

   where *platform* is `solsparc` or `soli386`, depending on the platform.

7. **Change to the directory containing the Message Queue packages.**

   ```
   cd mq3_7-ent
   ```

8. **Become root.**

   ```
   su root
   ```

   When prompted, type your root password.

9. **Run the uninstall script.**

   ```
   ./mquninstall
   ```

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

10 **If you want to uninstall all the listed packages, enter y (yes), and skip to Step 15. Otherwise, continue with Step 11.**

11 **If you do not want to uninstall all of the Message Queue packages, then enter n (no), and use the `pkgrm` command, as described in Step 12, to uninstall the specific packages you want to remove.**

12 **Remove the Message Queue packages, using the following command:**

`pkgrm` *packageName*

where *packageName* specifies a Message Queue package. To remove multiple packages, separate the names by a space.

Because other products might be using Message Queue packages, be careful about removing them. The `pkgrm` command will warn you of any dependencies on a package before removing it.

13 **When prompted, confirm your removal request by typing y (yes).**

14 **Type q to quit.**

15 **Exit the root shell.**

Where To Go Next

Read the `README` and *Message Queue Release Notes* files.

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

For an overview of Sun Java System Message Queue concepts, see the *Message Queue Technical Overview*.

For a brief introduction to writing and compiling a client application, see the *Message Queue Developer's Guide for Java Clients* or the *Message Queue Developer's Guide for C Clients*.

For details on configuring brokers and managing a Message Queue messaging system, see the *Message Queue Administration Guide*.

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

CHAPTER 3

Linux Installation

This chapter covers the following topics as they apply to a Linux installation of Message Queue 3.7 UR1:

- "Hardware and Software Requirements" on page 39
- "Installing Message Queue" on page 40
- "Configuring the Java Runtime" on page 48
- "Locating RPM Information" on page 49
- "Uninstalling Message Queue on Linux" on page 49
- "Where To Go Next" on page 50

Hardware and Software Requirements

In order to install Message Queue 3.7 UR1, your Linux system should satisfy the minimum hardware and software requirements shown in Table 3–1.

TABLE 3–1 Hardware and Software Requirements (Linux)

Component	Requirements
CPU	Intel Pentium 2 (or compatible)
RAM	256 MB
Disk space	■ Compressed installation (`.zip`) file: Approximately 18 MB ■ Installed product: Approximately 18 MB **Note –** The installed product may require more space if the broker stores persistent messages locally.
Operating system	Red Hat Advanced Server 3.0, Update 6 Red Hat Advanced Server 4.0, Update 2

TABLE 3–1 Hardware and Software Requirements (Linux) *(Continued)*

Component	Requirements
Java 2 Platform, Standard Edition (J2SE)	See "Supported Platforms and Components" on page 14 for supported versions of the Java Runtime Environment (JRE) and Java Software Development Kit (JDK). **Note –** The Message Queue 3.7 UR1 software distribution includes the required JRE version at the time of release.

Installing Message Queue

The Message Queue 3.7 UR1 product can be downloaded from the Sun Java System Web site. Message Queue also depends on components that you must install in order to develop and run Message Queue clients. See Table 1–1 and Table 1–2 for more information.

Finding and Removing Earlier Message Queue Versions

Because Message Queue is installed with other products (such as Sun Java System Application Server), you should check whether an earlier version has already been installed on your system and, if so, uninstall it before installing Message Queue 3.7 UR1.

Migrating Files from an Earlier Installation

If Message Queue Version 3.0.*x* or 3.5 is already installed on your system and you want to preserve any of its instance data files (flat-file user repository, access control properties file, or password file), you will need to use the `mqmigrate` utility to migrate those files before uninstalling the earlier version of Message Queue. If you do not want to preserve your old data files, there is no need to run the `mqmigrate` utility.

Before you run the `mqmigrate` utility, you must be root.

```
su root
```

When prompted, enter your root password.

The `mqmigrate` utility assumes that the data files to be migrated are located in the standard directories shown in Table 3–2. In Message Queue 3.0.*x* installations, the base directory *baseDir* is assumed by default to be `/opt`, but the files may be installed in another, nonstandard base directory instead. The `mqmigrate` command provides an option, `-basedir`, for designating an alternate base directory from which to migrate the files:

```
./mqmigrate [-basedir baseDir]
```

For example, if the earlier installation's data files reside in a directory named, `/export`, you would migrate them with the command

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

TABLE 3–2 Message Queue Standard Data Directories

Message Queue Version	Standard Data Directories
3.0.*x*	*baseDir*`/imq/var` *baseDir*`/imq/etc`
3.5	`/var/opt/imq` `/etc/opt/imq`
3.6	`/var/opt/sun/mq` `/etc/opt/sun/mq`
3.7 UR1	`/var/opt/sun/mq` `/etc/opt/sun/mq`

Removing an Earlier Installation

Depending on the version, Message Queue might have been installed using either `.tar` files or the Red Hat Package Manager (RPM). You must therefore check for both of these installation methods. It is recommended that you check first for RPM installations and then for `.tar` file installations.

▼ To Find and Remove Earlier RPM-Installed Versions of Message Queue

1 **Check your system for an existing Message Queue installation.**

Enter the command

```
imqbrokerd -version
```

If an earlier version of Message Queue is already installed, the version number is displayed.

- If no previous Message Queue installation is present, you can proceed to install Message Queue 3.7 UR1, as described in “Installation Procedure” on page 43.
- If you do find an existing Message Queue installation, go on to the next step.

2 **Become root.**

```
su root
```

Enter you root password.

3 **(Optional) Migrate data files from the existing installation.**

If the existing installation is of Message Queue Version 3.5 or earlier and you want to preserve its existing data files, use the command

```
./mqmigrate
```

to migrate the installation's broker configuration files, persistent data, and security-related files to their proper locations for your planned Message Queue 3.7 UR1 installation. (This step is not necessary if the existing installation is of Message Queue 3.6 or later.)

Note – If you are migrating from a Message Queue 3.0.*x* installation located in a base directory other than the default (`/opt`), use the `mqmigrate` command's `-basedir` option to specify the alternate base directory. For example, if the existing installation resides in a base directory named `/export`, use the following command to migrate its data files.

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

If you do not know the correct base directory, you can find it by searching for the Message Queue `imqbrokerd` executable and noting its root installation directory.

---

4 **Remove the existing Message Queue installed software.**

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

- For Message Queue Version 3.5 or earlier:

  ```
  rpm -e imq-ent
  ```

  ```
  rpm -e imq
  ```

- For Message Queue version 3.6 or 3.7:

  List Message Queue installed packages using the following command.

  ```
  rpm -qa | grep mq
  ```

Remove listed packages using the following syntax:

rpm -e *pack1* [[,*pack2*]...]

▼ To Find and Remove Earlier Tar-Based Installed Message Queue

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

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

If there is no previous Message Queue installation, proceed to install Message Queue by following the directions in "Installation Procedure" on page 43.

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

a. **If you want to preserve existing broker instance data, run the `mqmigrate` utility (in the Message Queue 3.7 UR1 distribution) as described in "Migrating Files from an Earlier Installation" on page 40.**

The `mqmigrate` utility moves existing broker instance data (broker configuration files and persistent data) and security-related files, to new Message Queue 3.7 UR1 locations.

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

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

Installation Procedure

The following instructions explain how to download and install the Message Queue product on Linux from the Sun Java System Web site.

▼ To Install Message Queue on Linux

1 **Accept the license agreement on the Message Queue product download site.**

2 **Download the Message Queue product distribution file from the Web site into an empty, temporary download directory, *temp_directory*.**

The download file is: `mq3_7-ent-linx86.zip`

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

```
unzip mq3_7-ent-linx86.zip
```

The `unzip` command creates an `mq3_7-ent` directory, which contains the distribution files: `LICENSE`, `README`, and `COPYRIGHT` files; `mqinstall` utility; `mquninstall` utility; `mqmigrate` utility; and an `rpms` directory.

The following table describes the Message Queue RPM packages in the `rpms` directory. If any of these files already exist on your system, you need to check whether the version number is greater than that provided by Message Queue. If it is, you should do a custom install.

RPM	Description	Notes
`sun-mq-config`	`/etc` files	
`sun-mq-var`	`/var` files	

RPM	Description	Notes

`sun-mq`	`/opt` files for the Message Queue Core RPM	Depends on the files
		`sun-javahelp`
		`sun-mq-jmsclient`
		`sun-mq-config`
		`sun-mq-var`

`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`. Not installed by default. Use if your existing client applications contain scripts that depend upon former file locations (such as jar files).
`sun-mq-[locale]`	L10N files	No dependencies.
`sun-mq-len`	Legacy Enterprise license file.	No longer used.
`sun-javahelp`[1]	JavaHelp: API and runtime	Supports JavaHelp viewer for Admin Console help. This RPM is not installed by default.
`sun-nss`[1]	Network Security Services (NSS) libraries.	Needed to support C language clients. This RPM is not installed by default.
`sun-nspr`[1]	Netscape Portable Runtime (NSPR) libraries	Needed to support C clients. This RPM is not installed by default.
`sun-saaj`[1]	SOAP with Attachments API for Java: API and runtime	Required to support Java clients using SOAP/JAXM API. This RPM is not installed by default.

[1] This package is shared by a number of Sun Java System products. This package is not installed by default. You must manually install any shared packages after installing Message Queue packages.

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. This RPM is not installed by default.
`sun-javamail`[1]	JavaMail: API and runtime	Required to support Java clients using SOAP/JAXM API. This RPM is not installed by default.
`sun-jaf`[1]	JavaBeans Activation Framework: API and runtime	Required to support Java clients using SOAP/JAXM API. This RPM is not installed by default.

[1] This package is shared by a number of Sun Java System products. This package is not installed by default. You must manually install any shared packages after installing Message Queue packages.

4 **Change to the directory containing the Message Queue distribution files.**

```
cd mq3_7-ent
```

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.

Note – Remember to migrate data if necessary. See "Migrating Files from an Earlier Installation" on page 40 for more information.

---

6 **If you intend to install all of the Message-Queue-specific RPM packages (packages which start with `sun-mq`), you will need to run the `mqinstall` script.**

Please note that the install script will overwrite all Message-Queue-specific RPM packages, and will not install any shared RPM packages. You must install shared RPM packages manually, as described in Step 7.

a. **Run the `mqinstall` script.**

```
./mqinstall
```

The script lists the distribution RPM packages including their version numbers, if any, that are already installed, and then lists the RPM packages about to be installed.

Note – If any of the Message Queue RPM packages listed is already installed and its version is newer, then you will probably not want to install the corresponding Message Queue RPM package. In that case, enter `n` (no) and continue with Step 7.

The `mqinstall` script does not install the `sun-mq-compat` RPM package by default and you should only have it installed if you need it. For example, if your existing client applications contain scripts that depend upon former file locations (such as jar files), you will need to install the `sun-mq-compat` RPM package. To install this RPM package, see Step 7.

b. **Enter `y` (yes) if you want to install all the Message-Queue-specific RPM packages. Otherwise, enter `n` (no) and continue with Step 7.**

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

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

The `/opt/sun/mq` directory and its contents are created, and files are placed in other locations as well (see "Migration Issues" on page 16).

7 **If you want to install a subset of the Message QueueRPM packages, if you want to install shared RPM packages, or if you do not want to overwrite later versions of RPM packages, do the following:**

a. **Change to the `rpms` directory.**

```
cd rpms
```

b. **Run the rpm install command to install the RPM packages.**

`rpm -ivh --nodeps` *rpmPkg1*`.rpm` *rpmPkg2*`.rpm` *rpmPkg3*`.rpm`

8 **Back up the `imq3_7-ent-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.

9 **Clean up all remaining files in your temporary working directory.**

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

Installed Directory Structure

Table 3–3 shows the installed directory structure for a full (all RPMs) installation of Message Queue 3.7 UR1 on the Linux platform. (The directory structure may vary if you perform a partial installation.)

Note – File locations for Message Queue bundled with Sun Java System Application Server may differ from those shown in the table.

TABLE 3–3 Installed Directory Structure (Linux)

Directory	Contents
`COPYRIGHT`(not installed)	Copyright text file
`ENTITLEMENT`	Evidence of purchase
`DISTRIBUTIONREADME`	Lists redistributable files
`LICENSE` (not installed)	License text file
`README`(not installed)	README text file
`THIRDPARTYLICENSECREADME`	NSS/NSPR license
`/opt/sun/mq/bin`	Executable files for the following administration tools: ■ Administration Console (`imqadmin`) ■ Broker utility (`imqbrokerd`) ■ Command utility (`imqcmd`) ■ Object Manager utility (`imqobjmgr`) ■ Database Manager utility (`imqdbmgr`) ■ User Manager utility (`imqusermgr`) ■ Key Tool utility (`imqkeytool`)
`/opt/sun/mq/share/lib`	Support files for Message Queue Java client runtime. ■ `.jar` files for building and running Java Message Service (JMS) client applications ■ `.rar` files for JMS Resource Adapter ■ `.war` files for HTTP servlet deployment
`/opt/sun/mq/share/lib/ext`	Files needed for JDBC-based persistence
`/opt/sun/mq/private/share/lib`	Support files for Message Queue tools and processes
`/opt/sun/mq/private/share/lib/props`	Broker's default configuration files
`/opt/sun/mq/private/share/lib/help`	Message Queue help files
`/opt/sun/mq/private/share/lib/images`	Admin GUI image files

TABLE 3–3 Installed Directory Structure (Linux) *(Continued)*

Directory	Contents
/opt/sun/mq/javadoc	Message Queue and JMS API documentation in JavaDoc format
/opt/sun/mq/examples	Example Java client applications
/opt/sun/mq/include	Header files to support C client applications
/var/opt/sun/mq	Message Queue working storage
/var/opt/sun/mq/instances	Configuration properties, file-based persistent data stores, log files, flat-file user repositories, and access control properties files for individual broker instances
/etc/opt/sun/mq	License files, instance template files, and rc script configuration files for automatic startup

Configuring the Java Runtime

At startup time, the Message Queue broker checks to make sure it has access to the required Java runtime version (JDK/JRE 1.5). 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:

▼ To Set the JRE Used by the Broker

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).**

To find out which JDK/JRE the broker is using, start the broker with the following command:

```
imqbrokerd -verbose
```

Locating RPM Information

You can query the RPM database to determine the installed version of RPMs.

To query the description tag from an installed RPM, use the following command:

```
rpm -q --queryformat ’%{DESCRIPTION}\\n’ sun-mq
```

This output shows a Version 3.7 UR1 `sun-mq` RPM.

```
SUNW_PRODVERS=3.7.0.1
```

Uninstalling Message Queue on Linux

The following instructions explain how to uninstall Message Queue on Linux.

▼ To Remove Message Queue on Linux

1 **Stop any running client applications.**

2 **Stop any running brokers. (You will be prompted for the admin user name and password.)**

```
imqcmd shutdown bkr [-b hostName:port]
```

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

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

4 **Get the uninstall script.**

5 **Find the zipped distribution file used to install Message Queue and place it in a temporary directory,** *temp_directory***.**

6 **Change directories to** *temp_directory***.**

```
cd temp_directory
```

7 **Unzip the distribution file.**

```
unzip mq3_7-ent-linx86.zip
```

8 **Change to the directory containing the Message Queue packages.**

```
cd mq3_7-ent
```

9 **Become root.**

```
su root
```

When prompted, type your root password.

10 **Run the uninstall script.**

```
./mquninstall
```

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

11 **If you want to uninstall all the listed RPM packages, enter `y` (yes) and skip to Step 14. Otherwise, continue with Step 12.**

12 **If you do not want to uninstall all of the Message Queue RPM packages, then enter `n` (no), and use the `rpm -e` command, as described in Step 13, to uninstall the specific RPM packages you want to remove.**

13 **Remove the Message Queue RPM packages manually using the following command:**

```
rpm -e packageName [[packageName]...]
```

Where *packageName* specifies a Message Queue RPM package.

Because other products might be using Message Queue RPM packages, be careful about removing them.

14 **Exit the shell.**

Where To Go Next

Read the `README` and *Message Queue Release Notes* files.

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

For an overview of Sun Java System Message Queue concepts, see the *Message Queue Technical Overview*.

For a brief introduction to writing and compiling a client application, see the *Message Queue Developer's Guide for Java Clients* or the *Message Queue Developer's Guide for C Clients*.

For details on configuring brokers and managing a Message Queue messaging system, see the *Message Queue Administration Guide*.

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

CHAPTER 4

Windows Installation

This chapter explains the following topics as they apply to a Windows installation of Sun Java System Message Queue 3.7 UR1:

- "Hardware and Software Requirements" on page 53
- "Installing Message Queue on Windows" on page 54
- "Troubleshooting Installation Problems" on page 57
- "Uninstalling Message Queue on Windows" on page 57
- "Where To Go Next" on page 58

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 (Windows)

Component	Requirements
Operating system	Windows XP Professional SP2 Windows 2000 all editions, SP4 or higher Wondows Server 2003, Enterprise Edition
CPU	Intel Pentium 3
RAM	256 MB
Disk space	The installation file containing the product is approximately 7 MB. The installed product requires approximately 13 MB. 125 MB are needed if a new Java runtime must be installed. Message Queue, however, may need more space if the broker stores persistent messages locally.

TABLE 4–1 Hardware and Software Requirements (Windows) *(Continued)*

Component	Requirements
Java 2 Platform, Standard Edition (J2SE)	See "Supported Platforms and Components" on page 14 for the supported versions of the Java Runtime Environment (JRE) and Java Software Development Kit (JDK) that are supported on Linux. The Message Queue software distribution includes the required JRE version at the time of release.

Message Queue also depends upon other technologies, as shown in "Supported Platforms and Components" on page 14.

Installing Message Queue on Windows

The Sun Java System Message Queue product can be downloaded from the Sun Java System Web site. Message Queue also depends upon components that you must install to be able to develop and run Message Queue clients. See Table 1–1 and Table 1–2 for more information.

The following instructions explain how to install the Message Queue product.

Upgrading from Previous Versions

To upgrade from prior Message Queue versions, it is recommended that you first uninstall Message Queue software as described in the *Message Queue Installation Guide* of the appropriate version, before installing Sun Java System Message Queue 3.7 UR1.

If you want to preserve data from a previous version of Message Queue when you migrate from an older version, refer to "Migration Issues" on page 16 prior to uninstalling Message Queue.

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

Installed Directory Structure

Table 4–2 shows the installed directory structure for Message Queue 3.7 UR1 on the Windows platform. Paths shown are relative to the Message Queue base directory (root installation directory), represented by the `IMQ_HOME` directory variable.

Note – File locations for Message Queue bundled with Sun Java System Application Server may differ from those shown in the table.

TABLE 4–2 Installed Directory Structure (Windows)

Windows	Contents
`IMQ_HOME\README.txt`	`README` text file
`IMQ_HOME\bin`	Executable files for the following administration tools: ■ Administration Console (`imqadmin`) ■ Broker utility (`imqbrokerd`) ■ Command utility (`imqcmd`) ■ Object Manager utility (`imqobjmgr`) ■ Database Manager utility (`imqdbmgr`) ■ User Manager utility (`imqusermgr`) ■ Service Administrator utility (`imqsvcadmin`) ■ Key Tool utility (`imqkeytool`) All executable files have the filename extension `.exe`. This directory also includes other executables (`imqbrokersvc`).
`IMQ_HOME\lib`	Support files for Message Queue Java client runtime. ■ `.jar` files for building and running Java Message Service (JMS) client applications ■ `.rar` files for JMS Resource Adapter ■ `.war` files for HTTP servlet deployment Support files for Message Queue tools and processes Libraries to support C client applications **Note –** See “Supported Platforms and Components” on page 14 for the versions of Netscape Portable Runtime (NSPR) and Network Security Service (NSS) needed to support the C API.
`IMQ_HOME\lib\props`	Broker's default configuration files
`IMQ_HOME\lib\ext`	Files needed for JDBC-based persistence
`IMQ_HOME\lib\help`	Message Queue help files
`IMQ_HOME\lib\images`	Admin GUI image files
`IMQ_HOME\var`	Message Queue working storage
`IMQ_HOME\var\instances`	Configuration properties, file-based persistent data stores, log files, flat-file user repositories, and access control properties files for individual broker instances

TABLE 4–2 Installed Directory Structure (Windows) *(Continued)*

Windows	Contents
`IMQ_HOME\javadoc`	Message Queue and JMS API documentation in JavaDoc format
`IMQ_HOME\demo`	Example Java client applications
`IMQ_HOME\demo\C`	Example C client applications
`IMQ_HOME\include`	Header files to support C client applications
`IMQ_HOME\etc`	License files and instance template files

Installing Message Queue

Follow these steps to install Message Queue 3.7 UR1 on Windows.

▼ To Install Message Queue on Windows

1 **Unzip the bundle** `mq_37–ent-win.zip`**.**

2 **Edit the file** `$TOP/mq/etc/imqenv.conf` **to set the** `IMQ_DEFAULT_JAVAHOME` **variable to specify a JDK 1.5.0 runtime. For example:**

```
set IMQ_DEFAULT_JAVAHOME=c:\Program Files\Java\jre1.5.0_09
```

3 **Change directories to** `$TOP/mq/bin`**.**

4 **Start the broker using the following command.**

```
./imqbrokerd -tty
```

Running Sample Applications

After installing Message Queue, you can compile and run example applications located in the `c:\Sun\MessageQueue\demo` directory. See the *Message Queue Developer's Guide for Java Clients* or the *Sun Java System Message Queue 3.7 UR1 Developer's Guide for C Clients* for additional instructions.

To run JMS client applications, include the following jar files in your classpath.

```
$TOP/mq/lib/jms.jar
$TOP/mq/lib/imq.jar
```

Troubleshooting Installation Problems

If the Message Queue installation program does not complete successfully on your Windows platform, use the following procedure to correct the problem.

▼ To Troubleshoot Installation Problems on Windows

1. **Clear any temporary directories (for example, `temp` or `tmp`).**

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 MessageQueue directory and all its contents.**

5. **In your Systems control panel, remove all references to MessageQueue from your environment settings.**

   Example: `IMQHOME` and `PATH` environment variables.

6. **Restart your Windows operating system.**

7. **Reinstall Message Queue in the same directory location as the initial failed installation.**

8. **Reset any services in the Services (Control Panel) that you stopped in "Troubleshooting Installation Problems" on page 57.**

Uninstalling Message Queue on Windows

The following instructions explain how to uninstall Message Queue on Windows.

▼ To Remove Message Queue on Windows

1. **Stop any running client applications.**

2. **Stop any running brokers. (You will be prompted for the admin user name and password.)**

   ```
   imqcmd shutdown bkr [-b hostName:port]
   ```

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

   a. **From the Windows Start menu, choose Settings, then Control Panel.**

b. Double-click the Administrative Tools icon, then the Services icon.

c. In the Services panel, select the Message Queue Broker entry, then click Stop.

d. Close the Services panel.

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

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

4 **Delete the folder containing the Message Queue 3.7 UR1 files.**

This is the same folder into which you decompressed the `mq3_7-ent-win.zip` file when installing Message Queue 3.7 UR1.

Where To Go Next

Read the `README` and *Message Queue Release Notes* files.

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

For an overview of Sun Java System Message Queue concepts, see the *Message Queue Technical Overview*.

For a brief introduction to writing and compiling a client application, see the *Message Queue Developer's Guide for Java Clients* or the *Message Queue Developer's Guide for C Clients*.

For details on configuring brokers and managing a Message Queue messaging system, see the *Message Queue Administration 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.