



MQSeries® for Sun Solaris

Quick Beginnings

Version 5.1



MQSeries® for Sun Solaris

Quick Beginnings

Version 5.1

Note

Before using this document, read the general information under Appendix D, "Notices" on page 111.

Third edition (June 1999)

This edition applies to Version 5.1 of *MQSeries for Sun Solaris* and to all subsequent releases and modifications until otherwise indicated in new editions or technical newsletters.

Order publications through your IBM® representative or the IBM branch office serving your locality. Publications are not stocked at the address below.

IBM welcomes your comments. A form for readers' comments may be provided at the back of this publication. If the form has been removed, you may send your comments to the following address:

IBM United Kingdom Laboratories,
Information Development,
Mail Point 095,
Hursley Park,
Winchester,
Hampshire,
England,
SO21 2JN

When you send information to IBM, you grant IBM a nonexclusive right to use or distribute the information in any way it believes appropriate without incurring any obligation to you.

© **Copyright International Business Machines Corporation 1995,1999. All rights reserved.**

US Government Users Restricted Rights – Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

Contents

Figures	v
Tables	v
Welcome to MQSeries for Sun Solaris	vii
Road Map	vii
Conventions	viii
What's New	ix
Changes for this Edition	ix
Changes for the Second Edition	x
Chapter 1. About MQSeries for Sun Solaris	1
Message Queuing	1
Messages and Queues	2
MQSeries Objects	3
Clients and Servers	5
Instrumentation Events	6
Transactional Support	7
Chapter 2. Planning to Install the MQSeries for Sun Solaris Server	9
Hardware Requirements	9
Software Requirements	10
Connectivity	10
Compilers Supported for Solaris Applications	11
Options	11
Delivery	12
Installation	12
MQSeries for Sun Solaris Components	12
README file	14
Migrating from an Earlier Version of MQSeries for Sun Solaris	15
Chapter 3. Installing the MQSeries for Sun Solaris Server	17
Preparing for Installation	17
Installing the MQSeries for Sun Solaris Server	19
Kernel Configuration	20
Installing the Server and Client on the Same Machine	21
Translated Messages	22
Translated Books	22
Verifying the Installation of MQSeries for Sun Solaris	22
User Exits	28
Setting the Queue Manager CCSID on MQSeries for Sun Solaris	29

Contents

Chapter 4. Planning to Install MQSeries Clients	31
Applications on Version 5.1 clients	31
Migrating from MQSeries Version 2 Products to Version 5.1	31
AIX Client: Hardware and Software Required	32
Components for AIX	33
DOS Client: Hardware and Software Required	34
Components for DOS	35
HP-UX Client: Hardware and Software Required	35
Components for HP-UX	37
OS/2 Warp Client: Hardware and Software Required	37
Components for OS/2 Warp	38
Sun Solaris Client: Hardware and Software Required	39
Components for Sun Solaris	40
Windows 3.1 Client: Hardware and Software Required	40
Components for Windows 3.1	42
Windows 95 and Windows 98 Client: Hardware and Software Required	42
Components for Windows 95 and Windows 98	43
Windows NT Client: Hardware and Software Required	44
Components for Windows NT	45
Chapter 5. Installing MQSeries Clients	47
Installing on AIX	47
Installing on DOS	54
Installing on HP-UX	55
Installing on OS/2 Warp	56
Installing on Sun Solaris	65
Installing on Windows 95 or Windows 98	67
Installing on Windows 3.1	71
Installing on Windows NT	73
Configuring and Verifying a Client	79
Chapter 6. Using the MQSeries Command Sets	81
Introducing Command Sets	81
Working with Queue Managers	84
Working with MQSeries Objects	87
Chapter 7. Using the MQSeries World Wide Web Interface	93
Overview of MQSeries Internet Gateway	93
Obtaining More Information	94
Obtaining MQSeries Internet Gateway	94
Chapter 8. Obtaining Additional Information	95
The MQSeries for Sun Solaris Library	95
Viewing Online Information	97
Printing and Ordering Books	99
Available from the Internet	100
Appendix A. Sample MQI Programs and MQSC Files	101

Tables

Appendix B. Applying maintenance to MQSeries for Sun Solaris	105
Space Requirements	106
Appendix C. Support for Different Code Sets on MQSeries for Sun Solaris	107
Migration to Euro Support	110
Appendix D. Notices	111
Trademarks	113
Index	115

Figures

1. Kernel parameter values - example setting on a Solaris 2.6 or Solaris 7 system	20
---	----

Tables

1. Getting Started Road Map	vii
2. Summary of Changes June 1999	ix
3. Summary of Changes March 1998	x
4. MQSeries publications – file names	99
5. MQSC command files	101
6. Sample programs - source files	101
7. Samples for transaction processing with CICS and Encina	103
8. Samples for transaction processing with Tuxedo	103
9. Sample programs - databases	104
10. Miscellaneous files	104
11. Locales and CCSIDs	107

Tables

Welcome to MQSeries for Sun Solaris

This book describes MQSeries for Sun Solaris and explains how to plan for, install, and use the product.

Road Map

Use Table 1 to find the information you need to get started with *MQSeries for Sun Solaris*.

<i>Table 1. Getting Started Road Map</i>	
If you want to...	Refer to...
Read about MQSeries for Sun Solaris	Chapter 1, "About MQSeries for Sun Solaris" on page 1
Learn about system requirements for MQSeries for Sun Solaris	Chapter 2, "Planning to Install the MQSeries for Sun Solaris Server" on page 9
Install MQSeries for Sun Solaris	Chapter 3, "Installing the MQSeries for Sun Solaris Server" on page 17
Learn about system requirements for clients	Chapter 4, "Planning to Install MQSeries Clients" on page 31
Install an MQSeries client	Chapter 5, "Installing MQSeries Clients" on page 47
Start using command sets	Chapter 6, "Using the MQSeries Command Sets" on page 81
Start using the Web Interface	Chapter 7, "Using the MQSeries World Wide Web Interface" on page 93
View or print online documentation	Chapter 8, "Obtaining Additional Information" on page 95
Contact IBM	<i>Readers' Comment Form</i>

Conventions

Conventions

Knowing the conventions used in this book will help you use it more efficiently.

- **Boldface type** indicates the name of an item you need to select or the name of a command.
- *Italics type* indicates new terms, book titles, or variable information that must be replaced by an actual value.
- Monospace type indicates an example (such as a fictitious path or file name) or text that is displayed on the screen.

What's New

Changes made since the first edition, are summarized here.

Changes for this Edition

Changes to this edition are marked by vertical bars in the left margin. Table 2 describes the major changes made in this edition.

Table 2. Summary of Changes June 1999

Page	Change
5	Namelists are now supported
15	Migration from an earlier version has now been documented
31	The MQSeries client planning information contains major changes
47	The MQSeries client installation procedure contains major changes
86	The endmqm command has a new flag added
110	The euro character is now supported

What's New

Changes for the Second Edition

Table 3 describes the changes made in the second edition (GC33-1870-01)

<i>Table 3. Summary of Changes March 1998</i>	
page	Change
14	Addition of a section on creating system default objects
29	Addition of a section on setting the queue manager coded character set identifier (CCSID)
106	Addition of a section on restoring a previous service level of MQSeries

Chapter 1. About MQSeries for Sun Solaris

This chapter introduces IBM MQSeries and describes its relationship with other products. It contains basic explanations of the following topics:

- "Message Queuing"
- "Messages and Queues" on page 2
- "MQSeries Objects" on page 3
- "Clients and Servers" on page 5
- "Instrumentation Events" on page 6
- "Transactional Support" on page 7

For more detailed explanations of these topics see Chapter 1, "Introduction to IBM MQSeries" in the *MQSeries Planning Guide*.

Message Queuing

MQSeries enables applications to use message queuing to participate in message-driven processing. Applications can communicate across different platforms by using the appropriate message queuing software products. The applications are shielded from the mechanics of the underlying communications.

MQI – a Common Application Programming Interface

All MQSeries products implement a common application programming interface (message queue interface or MQI), regardless of the platform on which the applications are run. The calls made by the applications and the messages they exchange are common. This makes it much easier to write and maintain applications than it is when using traditional methods. It also makes it easier to port applications from one platform to another.

The MQI is described in detail in the *MQSeries Application Programming Reference* book.

Messages and Queues

Time-Independent Applications

With message queuing, the exchange of messages between the sending and receiving programs is time independent. This means that the sending and receiving applications are decoupled so that the sender can continue processing without having to wait for the receiver to acknowledge the receipt of the message. In fact, the target application does not even have to be running when the message is sent. It can retrieve the message after it is started.

Message-Driven Processing

On arriving on a queue, messages can automatically start an application using a mechanism known as *triggering*. If necessary, the applications can be stopped when the message or messages have been processed.

Messages and Queues

Messages and queues are the basic components of a message queuing system.

What is a Message?

A *message* is a string of bytes that has meaning to the applications that use it. Messages are used for transferring information from one application to another (or to different parts of the same application). The applications can be running on the same platform, or on different platforms.

MQSeries messages have two parts; the *application data* and a *message descriptor*. The content and structure of the application data is defined by the application programs that use the data. The message descriptor identifies the message and contains other control information, such as the type of message and the priority assigned to the message by the sending application.

What is a Queue?

A *queue* is a data structure that stores messages. The messages may be put on the queue by applications or by a queue manager as part of its normal operation.

Queues exist independently of the applications that use them. A queue can exist in main storage (if it is temporary), on disk or similar auxiliary storage (if it must be kept in

MQSeries Objects

case of recovery), or in both places (if it is currently being used, and must also be kept for recovery). Each queue belongs to a *queue manager*, which is responsible for maintaining it. The queue manager puts the messages it receives onto the appropriate queue.

Queues can exist either in your local system, in which case they are called *local queues*, or at another queue manager, in which case they are called *remote queues*.

Applications send and receive messages using MQI calls. For example, one application can put a message on a queue using the MQPUT call, and another application can retrieve the message from the same queue using the MQGET call.

MQSeries Objects

An MQSeries object is a recoverable resource managed by MQSeries. Many of the tasks described in this chapter involve manipulating the following types of MQSeries object:

- Queue managers
- Queues
- Process definitions
- Channels
- Namelists

For system administrators, commands are available to manipulate objects. Default objects are created for you when you create a queue manager.

Each object has a *name* associated with it and can be referenced in MQSeries commands and MQI calls by that name. Names must be unique within each of the object types. For example, you can have a queue and a process with the same name, but you cannot have two queues with the same name.

Queue Managers

A queue manager provides queuing services to applications, and manages the queues that belong to it. It ensures that:

- Object attributes are changed according to the commands received.
- Special events such as trigger events or instrumentation events are generated when the appropriate conditions are met.
- Messages are put on the correct queue, as requested by the application making the MQPUT call. The application is informed if this cannot be done, and an appropriate reason code is given.

MQSeries Objects

Each queue belongs to a single queue manager and is said to be a *local queue* to that queue manager. The queue manager to which an application is connected is said to be the local queue manager for that application. For the application, the queues that belong to its local queue manager are local queues. A *remote queue* is simply a queue that belongs to another queue manager. A *remote queue manager* is any queue manager other than the local queue manager. A remote queue manager may exist on a remote machine across the network or it may exist on the same machine as the local queue manager. MQSeries supports multiple queue managers on the same machine.

Queues

A queue is an MQSeries object that can store messages. Each queue has *queue attributes* that determine what happens when applications reference the queue in MQI calls. The attributes indicate:

- Whether applications can retrieve messages from the queue (get enabled)
- Whether applications can put messages onto the queue (put enabled)
- Whether access to the queue is exclusive to one application or shared between applications
- The maximum number of messages that can be stored on the queue at the same time (maximum queue depth)
- The maximum size of messages that can be put on the queue (maximum message size)

Process Definitions

A *process definition object* defines an application that is to be started in response to a *trigger event* on an MQSeries queue manager.

A trigger event is a logical combination of conditions that is detected by a queue manager. For example, a trigger event may be generated when the number of messages on a queue reaches a predefined level. This event causes the queue manager to put a trigger message on a specified initiation queue. This trigger message is retrieved by a *trigger monitor*, a special application that monitors an initiation queue. The trigger monitor then starts up the application program that was specified in the trigger message.

If a queue manager is to use triggering, at least one initiation queue must be defined for that queue manager.

See Chapter 14, "Starting MQSeries applications using triggers" in the *MQSeries Application Programming Guide* for more information about triggering.

Channels

A channel provides a communication path. There are two types of channel: message channels and MQI channels.

A *message channel* provides a communication path between two queue managers on the same, or different, platforms. The message channel is used for the transmission of messages from one queue manager to another, and shields the application programs from the complexities of the underlying networking protocols.

A message channel can transmit messages in one direction only. If two-way communication is required between two queue managers, two message channels are required.

An *MQI channel* connects an MQSeries client to a queue manager on a server machine. It is for the transfer of MQI calls (for example, MQPUT) and responses only and is bidirectional. A channel definition exists for each end of the link. On some platforms, some types of MQI channel can be defined automatically.

For more information on channels and how to use them, see “Distributed queuing components” in the *MQSeries Intercommunication* book.

Namelist

A *namelist* is an MQSeries object that contains a list of other MQSeries objects. Typically, namelists are used by applications such as trigger monitors, where they are used to identify a group of queues. The advantage of using a namelist is that it is maintained independently of applications; that is, it can be updated without stopping any of the applications that use it. Also, if one application fails, the namelist is not affected and other applications can continue using it.

Clients and Servers

MQSeries supports client/server configurations for MQSeries applications.

An *MQSeries client* is a part of the MQSeries product that is installed on a machine to accept MQI calls from applications and pass them to an *MQI server* machine. There they are processed by a queue manager. Typically, the client and server reside on different machines, but they can also exist on the same machine.

An *MQI server* is a queue manager that provides queuing services to one or more clients. All the MQSeries objects, for example queues, exist only on the queue

Instrumentation Events

manager machine, that is, on the MQI server machine. A server can support local MQSeries applications as well.

The difference between an MQI server and an ordinary queue manager is that the MQI server can support MQI clients, and each MQI client has a dedicated communications link with the MQI server. For more information about creating channels for clients and servers, see Chapter 8, "Using channels" in the *MQSeries Clients* book. For information about client support in general, see Chapter 1, "Overview" in the *MQSeries Clients* book.

Instrumentation Events

You can use MQSeries instrumentation events to monitor the operation of queue managers.

Instrumentation events cause special messages, called *event messages*, to be generated whenever the queue manager detects a predefined set of conditions. For example, the following conditions give rise to a *Queue Full* event:

- Queue Full events are enabled for a specified queue, and
- An application issues an MQPUT call to put a message on that queue, but the call fails because the queue is full.

Other conditions that can give rise to instrumentation events include:

- A predefined limit for the number of messages on a queue being reached
- A queue not being serviced within a specified time
- A channel instance being started or stopped
- An application attempting to open a queue and specifying a user ID that is not authorized

If you define your event queues as remote queues, you can put all the event queues on a single queue manager (for those nodes that support instrumentation events). You can then use the events generated to monitor a network of queue managers from a single node.

Types of Event

MQSeries events are categorized as follows:

Queue manager events

These events are related to the definitions of resources within queue managers.

For example, if an application attempts to open a queue but the associated user ID is not authorized to perform that operation, a queue manager event is generated.

Transactional Support

Performance events

These events are notifications that a threshold condition has been reached by a resource. For example, a queue depth limit has been reached or, following an MQGET request, a queue has not been serviced within a predefined period of time.

Channel events

These events are reported by channels as a result of conditions detected during their operation. For example, a channel event is generated when a channel instance is stopped.

Transactional Support

An application program can group a set of updates into a *unit of work*. These updates are usually logically related and must all be successful for data integrity to be preserved. If one update succeeded while another failed then data integrity would be lost.

A unit of work *commits* when it completes successfully. At this point all updates made within that unit of work are made permanent or irreversible. If the unit of work fails then all updates are instead *backed out*. *Syncpoint coordination* is the process by which units of work are either committed or backed out with integrity.

A *local* unit of work is one in which the only resources updated are those of the MQSeries queue manager. Here syncpoint coordination is provided by the queue manager itself using a single-phase commit process.

A *global* unit of work is one in which resources belonging to other resource managers, such as XA-compliant databases, are also updated. Here, a two-phase commit procedure must be used and the unit of work may be coordinated by the queue manager itself, or externally by another XA-compliant transaction manager such as TXSeries™ or BEA Tuxedo.

When the queue manager coordinates global units of work itself it becomes possible to integrate database updates within MQSeries units of work. That is, a mixed MQI and SQL application can be written, and commands can be used to commit or roll back the changes to the queues and databases together.

The queue manager achieves this using a two-phase commit protocol. When a unit of work is to be committed, the queue manager first asks each participating database manager whether it is prepared to commit its updates. Only if all of the participants, including the queue manager itself, are prepared to commit, are all of the queue and database updates committed. If any participant cannot prepare its updates, the unit of work is backed out instead.

Transactional Support

Full recovery support is provided if the queue manager loses contact with any of the database managers during the commit protocol. If a database manager becomes unavailable while it is in doubt, that is, it has been called to prepare but has yet to receive a commit or backout decision, the queue manager remembers the outcome of the unit of work until it has been successfully delivered. Similarly, if the queue manager terminates with incomplete commit operations outstanding, these are remembered over queue manager restart.

Chapter 2. Planning to Install the MQSeries for Sun Solaris Server

This chapter is a summary of the requirements running MQSeries for Sun Solaris, the network protocols and the compilers supported, the delivery media, and the various components of the product.

Year 2000 compatibility

MQSeries, when used in accordance with its associated documentation, is capable of correctly processing, providing, and/or receiving date data within and between the twentieth and twenty-first centuries, provided that all products (for example, hardware, software, and firmware) used with this IBM Program properly exchange accurate date data with it.

Customers should contact third party owners or vendors regarding the readiness status of their products.

IBM reserves the right to update the information shown here. For the latest information regarding levels of supported software, refer to:

<http://www.software.ibm.com/ts/mqseries/platforms/supported.html>

For the latest IBM statement regarding Year 2000 readiness, refer to:

<http://www.ibm.com/IBM/year2000/>

Hardware Requirements

- MQSeries Servers:
 - Any Sun SPARC desktop or server system
 - Any Sun UltraSPARC desktop or server system

Disk Storage

The installation requirements depend on which components you install and how much working space you need. This, in turn, depends on the number of queues that you use,

Connectivity

the number and size of the messages on the queues, and whether the messages are persistent or not. You also require archiving capacity on disk, tape, or other media.

These are the approximate storage requirements:

- Server:

A minimum of 25 MB of disk space must be available for the product code and data in the filesystem containing the `/opt` directory.

In addition, if you install the online books in HTML format you require 35 MB of storage for the books in the `/opt` directory.

After installation the books are placed in the `/opt/mqm/html` directory.

Working data for MQSeries for Sun Solaris is stored by default in `/var/mqm`.

Note: For added confidence in the integrity of your data, you are strongly advised to put your logs onto a *different* physical drive from the one that you use for the queues.

Software Requirements

Minimum supported levels are shown. Later compatible levels, if any, will be supported unless otherwise stated.

- Sun Solaris Version 2.6 with patches 105210-13 and 105568-10 or Sun Solaris 7

Note: No DCE or TXSeries products are supported with Sun Solaris 7.

Connectivity

Network protocols supported are SNA LU 6.2 and TCP.

- Any communications hardware supporting SNA LU 6.2 or TCP

For SNA connectivity:

- SunLink SNA Peer-to-Peer Version 9.1
- If token ring is used: SunLink Token Ring Interface/ SBus 3.0.2 (requires patch 102463)
- Sun TRI/P Adaptor V1.0
- TCP is part of the base operating system.

Compilers Supported for Solaris Applications

- Sun WorkShop Compiler C Version 4.2
- Sun WorkShop Compiler C++ Version 4.2
- Micro Focus COBOL compiler Version 4.0 for UNIX

In order to build all MQSeries components, the application developer's kits for CICS®, DCE, and SunLink SNA are also required.

Options

You may use the following options with MQSeries for Sun Solaris.

Transaction Monitors

The following transaction processing monitors (coordination can be through X/Open XA interface) may be used:

- TXSeries for Sun Solaris V4.2
- BEA Tuxedo Version 5.1, 6.1, or 6.4

Databases

- DB2® Universal Database V5.0
- Oracle release 7.3.2.3, or release 8.0.4
- Sybase XA Server V11.1.1 and EBF7589
- Sybase Adaptive Server V11.5
- Sybase Open Client (ctlib and dlib) V11.1.1
- Sybase Embedded SQL/C 11.1.0

DCE

Transarc DCE V1.1 and later. This must be the U.S. Domestic version supporting DES encryption if you want to run the MQSeries-supplied DCE and send, receive or use message exits.

MQSeries for Sun Solaris Components

DCE names and security modules are provided with MQSeries for Sun Solaris V5.1.

Delivery

MQSeries for Sun Solaris V5.1 is supplied on CD-ROM.

Installation

MQSeries for Sun Solaris takes approximately 5 minutes to install, using the Solaris **pkgadd** program.

MQSeries for Sun Solaris Components

When you install MQSeries for Sun Solaris you can choose which components to install. The components are as follows:

Runtime component

Support for external applications. This does **not** enable you to write your own applications.

Base

Support to enable you to create and support your own applications. Requires the runtime component to be installed.

Server

Support for client connections. Requires the runtime component to be installed.

MQSeries Clients

MQSeries Client for Sun Solaris and MQSeries client for Java™:

- The MQSeries Client for Sun Solaris can be installed on the server machine, enabling you to have the MQSeries server and client on the same machine.
- The MQSeries Client for Java allows Java applets running on your machine to communicate with MQSeries. It includes security exits for encryption and authentication of messages sent across the web by the MQSeries Client for Java. These exits consist of some Java classes.

MQSeries for Sun Solaris Components

Notes:

1. Once you have installed and configured your machine to use this version of the MQSeries Client for Java, you may not be able to run applets, that were written using an earlier version of the MQSeries Client for Java, from a browser.

This is because the browser picks up the locally installed version of the MQSeries Client for Java class files from your CLASSPATH statement, and these files are incompatible with earlier releases.

If you want to run your old applets, remove the MQSeries Client for Java library from your CLASSPATH statement and restart your web browser.

2. The Java Client requires a Java 1.1.1 (or later) capable Web Browser. This may not be available on all platforms.

Some providers of Java place limitations on its support or warranty. Read the documentation from the Java provider to understand any limitations on its use.

DCE support

Support for DCE names or security on the server. This should be installed **only** if you are going to use DCE.

Note to users

If you install the DCE option, Java Bindings are **not** supported.

DCE samples

This should be installed **only** if you are going to use DCE.

MQSeries Online Documentation

Online versions of the books for MQSeries for Sun Solaris in HTML format.

Included are:

- MQSeries Documentation
- MQSeries Internet Documentation

PDF versions of the MQSeries books are also on the CD-ROM, but are not listed as installable components.

HTML and PDF versions of some of the MQSeries books are available in the following national languages:

- U.S. English
- Brazilian Portuguese
- French
- German
- Italian
- Japanese
- Korean
- Spanish
- Simplified Chinese
- Traditional Chinese

README

MQSeries Internet Gateway

Provides access to MQSeries applications through HTML and CGI.

Man

Man pages for the following commands:

- Control commands
- Message Queue Interface (MQI)
- MQSeries commands (MQSC)

Samples

Sample application programs.

MQSeries code

The MQSeries code in the national languages listed under the **MQSeries Online Documentation** component.

Note: The “base” product is automatically installed.

Creating the System Default Objects

When you use the **crtmqm** command to create a queue manager with this release of MQSeries, the system default objects are automatically created. The sample MQSC definition file, `amqscoma.tst`, is no longer provided.

If you used `amqscoma.tst` to customize your settings for MQSeries Version 2, and you want to use the same settings with Version 5.1 of the product:

1. Save your copy of `amqscoma.tst`
2. Install MQSeries Version 5.1
3. Load your copy of `amqscoma.tst` and use the file to recreate your default objects

README file

Before starting to install MQSeries for Sun Solaris, review the README file, which you will find in the root directory of the CD-ROM.

Migrating from an Earlier Version of MQSeries for Sun Solaris

If you want to migrate from MQSeries for Sun Solaris V2.2 or MQSeries for Sun Solaris V5.0 to MQSeries for Sun Solaris V5.1, you should follow this procedure:

1. End all queue manager activity. Do this with the **endmqm** command. See “Stopping a Queue Manager” on page 85 for information on how to use the **endmqm** command.
2. Stop all MQSeries activity and remove any shared resources that MQSeries uses. Do this by shutting down the system and restarting it, or use the **ipcs -a** command to display shared memory segments or semaphore sets created by MQSeries, and remove them using the **ipcrm** command. You do not have to recreate your MQSeries objects.
3. Uninstall the old MQSeries for Sun Solaris using the **pkgrm** program. Do not delete the `/var/mqm` directory tree if you want to retain your own MQSeries information, for example your queue manager data.

You are now ready to install MQSeries for Sun Solaris V5.1. See “Installing the MQSeries for Sun Solaris Server” on page 19 for information on installation.

MQSeries Link to Lotus Notes®

MQSeries for Sun Solaris V5.1 does not contain the MQSeries link to Lotus Notes. When you migrate to MQSeries for Sun Solaris V5.1 from a previous version of the product with which the MQSeries link to Lotus Notes was provided, the source of the link to Lotus Notes is deleted from the install directory. However, it is not deleted from the Lotus Notes directory, into which it was originally copied before use. Therefore, if you are an existing user of the MQSeries link to Lotus Notes, you may continue to use this function with MQSeries for Sun Solaris V5.1.

Alternatively, you can download the MQSeries link LotusScript® Extension (MQLSX), which provides equivalent function to the MQSeries link to Lotus Notes, from the MQSeries Web site at:

http://www.software.ibm.com/ts/lotus_connections/platforms/mqlsx.html

Link to Lotus Notes®

Chapter 3. Installing the MQSeries for Sun Solaris Server

This chapter tells you how to install MQSeries for Sun Solaris and how to verify that your installation has been successful.

The MQSeries product is installed into the `/opt/mqm` directory. This *cannot* be changed. However, if you do not have enough space in the `/opt/mqm` file system, follow the procedure given in "Creating Another File System for Product Code" on page 18.

Note: The MQSeries product is contained in the `/mq_solaris` directory of the CD-ROM.

Preparing for Installation

This section guides you through some of the steps you must perform before you install MQSeries for Sun Solaris.

Before Installation

Before you can install MQSeries for Sun Solaris you:

- Must install any patches listed in the README
- Must create a group with the name `mqm`
- Must add **root** to the `mqm` group
- Must create a user ID with the name `mqm`
- Are recommended to create and mount a `/var/mqm` file system, or `/var/mqm`, `/var/mqm/log`, and `/var/mqm/errors` file systems

You should allow a minimum of 30 MB of storage for `/var/mqm`, 2 MB of storage for `/var/mqm/errors`, and 20 MB of storage for `/var/mqm/log` if you are creating separate file systems.

If you are using a single file system, use the sum of these figures as a guide.

Preparing for installation

Notes:

1. The size of the `/var/mqm` file system should be large enough to contain all the messages, on all the queue managers, on this system.
2. If you create separate partitions, the following directories **must** be on a local file system:
 - `/var/mqm`
 - `/var/mqm/log`

You can choose to NFS mount the `/var/mqm/errors` and `/var/mqm/trace` directories to conserve space on your local system.

3. The size of the log file depends upon the log settings that you use. The size recommended is for circular logging using the default settings. For further information on log sizes see “Calculating the size of the log” in the *MQSeries System Administration* book.

After installation, this user ID (`mqm`) owns the directories and files that contain the resources associated with the product. This group and user must be defined for any machine on which the MQSeries software is to be installed, whether the machine is a client or a server machine.

If you want to run any administration commands, for example, **crtmqm** (create queue manager) or **strmqm** (start queue manager), your user ID must be a member of group `mqm`.

For stand-alone machines, you can create the new user and group IDs locally. For machines administered in a network information services (NIS) domain, you can create the user and group IDs on the NIS master server machine.

Creating Another File System for Product Code

If you do not want to have the product code installed in the `/opt/mqm` file system, for example, if that file system is too small to contain the product, you can do one of two things:

1. Create a new file system and mount it as `/opt/mqm`.
2. Create a new directory anywhere on your machine that is large enough to contain the product, and create a symbolic link from `/opt/mqm` to this new directory. For example:

```
mkdir /bigdisk/mqm
ln -s /bigdisk/mqm /opt/mqm
```

Server Installation

Notes:

1. Whichever of these options you pick, you *must* do it before installing the product code.
2. The file system into which the code is installed can be a remote network device, for example NFS, provided that the mount options are defined on that device to allow *setuid* programs – including root access – to be run.

Installing the MQSeries for Sun Solaris Server

This section describes the installation of the MQSeries for Sun Solaris server.

Notes:

1. If you have previously installed MQSeries on your system, you need to remove the product using the **pkgrm** program. See “Migrating from an Earlier Version of MQSeries for Sun Solaris” on page 15 for more information.
2. If the product is present, but not installed correctly, you may need to manually delete the files and directories contained in:

```
/var/mqm  
/opt/mqm
```

Carry out the following procedure:

1. Check to see if Volume Manager is running on your system by typing the following command:

```
/usr/bin/ps -ef | /bin/grep vold
```

If it is running, the CD is mounted on /cdrom/mq_solaris automatically. If it is not running, mount the CD by typing the following commands:

```
mkdir -p /cdrom/mq_solaris  
mount -F hsfs -r /dev/dsk/cntndnsn /cdrom/mq_solaris
```

substituting cntndnsn with the name of your CD-ROM device.

2. Use the Solaris **pkgadd** program, to install the software by carrying out the following procedure:
 - a. Type `pkgadd -d /cdrom/mq_solaris/`.
 - b. You are prompted for a list of components to install. Select those you require. If you want to install the entire MQSeries product, select **all**.
 - c. Press the Enter key.

Kernel configuration

For further information on using **pkgadd** to install software packages, see the Solaris documentation.

Kernel Configuration

MQSeries makes use of semaphores, shared memory, and file descriptors, and it is probable that the **default** kernel configuration is not adequate.

In particular, the default number of semaphores is 60, which is *not* sufficient to support MQSeries.

If you attempt to use MQSeries without increasing `semnmi`, the number of semaphores, the queue manager fails and produces a First Failure Support Technology™ (FFST) file. This file indicates that the system call `semop` received an argument that was not valid. An example of a possible set of actual kernel values on a Solaris 2.6 system is given in Figure 1. However, for Sun Solaris the number of semaphores and semaphore sets has to be such that their control structures occupy less than 25% of the kernel storage.

After installation, you should review the machine's configuration. To do this type the following command:

```
sysdef -i
```

To change the values, add a `set parameter = value` line to the `/etc/system` file. For further information on setting up the system, see the Sun Solaris System Administration documentation.

```
set shmsys:shminfo_shmmax = 4194304
set shmsys:shminfo_shmseg = 1024
set shmsys:shminfo_shmmni = 1024
set semsys:seminfo_semaem = 16384
set semsys:seminfo_semnmi = 1024 (semnmi < semmns)
set semsys:seminfo_semmap = 1026 (semnmi +2)
set semsys:seminfo_semmns = 16384
set semsys:seminfo_semmsl = 100
set semsys:seminfo_semopm = 100
set semsys:seminfo_semmnu = 2048
set semsys:seminfo_semume = 256
set msgsys:msginfo_msgmap = 1026
set msgsys:msginfo_msgmax = 4096
```

Figure 1. Kernel parameter values - example setting on a Solaris 2.6 or Solaris 7 system

Installing Server and Client

Notes:

1. Shared memory usage does not vary with message rate or persistence.
2. Semaphore and swap usage does not vary with message size, message rate or persistence.
3. MQSeries queue managers are independent of each other. Therefore system kernel parameters, for example `shmmni`, `semnmi`, `semnms`, and `semnmu` need to allow for the number of queue managers in the system.

For more details see the relevant SupportPac™, which is available on the MQSeries family home page. See “Available from the Internet” on page 100 for further details.

Sun Solaris has a low default system soft limit for the number of file descriptors. When running a multi-threaded process, you may reach the soft limit for file descriptors. This will give you the MQSeries reason code `MQRC_UNEXPECTED_ERROR` (2195), and an MQSeries FFST file.

To avoid this problem you can increase the system soft limit for the number of file descriptors. To do this:

- Edit the `/etc/system` file and change the value of the system soft limit to match the system hard limit (1024).

Additionally, if you are running MQSeries under the Lotus Domino server, you can reduce the number of active server threads in the Domino HTTP server process. To do this:

- Open the server **Name and address** book, and reduce the **Number active threads** value on the server document to between 50 and 60.

Installing the Server and Client on the Same Machine

To install an MQSeries for Sun Solaris client on the server machine, use the MQSeries Server CD-ROM. Choose the client install option on the server CD-ROM to install the client code on the server machine. Do not use the MQSeries Clients CD-ROM.

You might install components from the MQSeries Clients CD-ROM onto a machine and then later want to install the MQSeries server component on the same machine. If so, you must first remove from the machine any of the components that were installed from the MQSeries Clients CD-ROM. You can then use the MQSeries Server CD-ROM to install the server, client, and any other components that you need. You cannot install the server on a machine that already has other components installed from the MQSeries Clients CD-ROM.

Verifying the Installation

Translated Messages

Messages in U.S. English are always available. If you require another of the languages that is supported by MQSeries for Sun Solaris Version 5.1, you *must* ensure that your NLSPATH environment variable includes the appropriate directory.

For example:

```
export LANG=de
export NLSPATH=/usr/lib/locale/%L/LC_MESSAGES/%N
```

Translated Books

If you choose to install the Online Documentation component, you will get books in the language that was specified when your operating system was installed. However, some books may not be available in languages other than U.S. English and some hypertext links between books may not work. To overcome this you must choose to install a complete set of books in U.S. English as well as those in your national language. See “Viewing Online Books” on page 97 for more information about hypertext linking between translated books.

Verifying the Installation of MQSeries for Sun Solaris

This section describes how to verify that MQSeries for Sun Solaris has been correctly installed and configured. You do this by following the steps outlined in “Verification Procedure” on page 23.

If you want to verify a communications link between multiple MQSeries installations (for example between two servers or between a client and a server), you must ensure that the required communications protocols have been installed (and configured) on **both** machines.

The supported protocols are TCP and SNA.

Note: The following examples assume that you will be using a TCP connection; for information about using other protocols, see Chapter 6, “Channel attributes” in the

Verifying the Installation

MQSeries Intercommunication book. However, you can also verify a *local* installation (which has no communications links with other MQSeries installations) without any communications protocols installed.

Verification Procedure

You can verify an MQSeries installation at three levels:

- A local (standalone) installation, involving no communication links to other MQSeries machines
- A server-to-server installation, involving communication links with other MQSeries servers
- A client/server installation, involving communication links between a server machine and an MQSeries client

Verification of local and server-to-server installations is described in “Verifying a Local Installation,” and in “Verifying a Server-to-Server Installation” on page 25 . For information on verifying a client/server installation, see Chapter 6, “Verifying the installation” in the *MQSeries Clients* book.

Verifying a Local Installation

Follow these steps to install and test a simple configuration of one queue manager and one queue, using sample applications to put a message onto the queue and to read the message from the queue:

1. Install MQSeries for Sun Solaris on the workstation (include the Base Server component as a minimum).
2. Create a default queue manager (in this example called `venus.queue.manager`):
 - At the command prompt in the window type:

```
crtmqm -q venus.queue.manager
```
 - Press Enter.

Messages are displayed telling you that the queue manager has been created, and that the default MQSeries objects have been created.

Note: In prior releases of MQSeries it was necessary to run a script file called **amqscoma.tst** to define the MQSeries default objects. This step is not required in this release of the product.

3. Start the default queue manager:
 - Type `strmqm` and press Enter:
A message tells you when the queue manager has started.

Verifying the Installation

4. Enable MQSC commands by typing the following command and then pressing Enter:

```
runmqsc
```

Note: MQSC has started when the following message is displayed: Starting MQSeries Commands.

MQSC has no command prompt.

5. Define a local queue (in this example, called `ORANGE.QUEUE`):

- Type the following and press Enter:

```
define qlocal (orange.queue)
```

Note: Any text entered in MQSC in lowercase is converted automatically to uppercase unless you enclose it in single quotation marks. This means that if you create a queue with the name `orange.queue`, you must remember to refer to it in any commands outside MQSC as `ORANGE.QUEUE`.

The message MQSeries queue created is displayed when the queue has been created.

You have now defined:

- A default queue manager called `venus.queue.manager`
- A queue called `ORANGE.QUEUE`

6. Stop MQSC by pressing Ctrl-D, or typing **end**, and pressing Enter.

The following message is displayed:

```
One MQSC commands read.  
No commands have a syntax error.  
All valid MQSC commands were processed.
```

7. The command prompt is now displayed again.

To test the queue and queue manager, use the samples **amqsput** (to put a message on the queue) and **amqsget** (to get the message from the queue):

1. Change into the following directory:

```
/opt/mqm/samp/bin
```

2. To put a message on the queue, type the following command and press Enter:

```
amqsput ORANGE.QUEUE
```

The following message is displayed:

```
Sample amqsput0 start  
target queue is ORANGE.QUEUE
```

3. Type some message text and then press Enter **twice**.

The following message is displayed:

```
Sample amqsput0 end
```

Your message is now on the queue and the command prompt is displayed again.

Verifying the Installation

4. If you are not already in the following directory, change to it now:

```
/opt/mqm/samp/bin
```

5. To get the message from the queue, type the following command and press Enter:

```
amqsget ORANGE.QUEUE
```

The sample program starts, your message is displayed, the sample ends, and the command prompt is displayed again.

The verification is complete.

Verifying a Server-to-Server Installation

The steps involved in verifying a server-to-server installation are more complex, because the communications link between the two machines must be checked.

Follow these steps to set up two workstations, one as a sender and one as a receiver.

Sender Workstation:

1. Create a default queue manager called `saturn.queue.manager`:

- At a command prompt in a window, type:

```
crtmqm -q saturn.queue.manager
```

- Press Enter.

Messages are displayed telling you that the queue manager has been created, and that the default MQSeries objects have been created.

Note: In prior releases of MQSeries it was necessary to run a script file called `amqscoma.tst` to define the MQSeries default objects. This step is not required in this release of the product.

2. Start the queue manager:

- Type the following and then press Enter:

```
strmqm
```

A message tells you when the queue manager has started.

3. Enable MQSeries Commands (MQSC) by typing the following command and then pressing Enter:

```
runmqsc
```

Note: MQSC has started when the following message is displayed: Starting MQSeries Commands.

MQSC has no command prompt.

Verifying the Installation

4. Define a local queue to be used as a transmission queue, called TRANSMIT1.QUEUE:

- Type the following and press Enter:

```
define qlocal (transmit1.queue) usage (xmitq)
```

The message MQSeries queue created is displayed when the queue has been created.

5. Create a local definition of the remote queue:

```
define qremote (local.def.of.remote.queue) rname (orange.queue) +  
rqmname ('venus.queue.manager') xmitq (transmit1.queue)
```

Note: The RNAME parameter specifies the name of the queue on the remote machine to which the message is being sent. Therefore, the name specified by the RNAME parameter (ORANGE.QUEUE) must be the same as the name of the queue to which the message is being sent (ORANGE.QUEUE on the receiver workstation).

6. Define a sender channel:

```
define channel (first.channel) chltype (sdr) conname (9.20.11.182) +  
xmitq (transmit1.queue) trptype (tcp)
```

where 9.20.11.182 is the TCP address of the receiver workstation (note that this example is TCP specific).

You have now defined the following objects:

- A default queue manager called saturn.queue.manager
- A transmission queue called TRANSMIT1.QUEUE
- A remote queue called LOCAL.DEF.OF.REMOTE.QUEUE
- A sender channel called FIRST.CHANNEL

7. Stop MQSC by pressing Ctrl-D, or typing **end**, and pressing Enter.

Now set up the receiver workstation.

Receiver Workstation

Note: You must be logged in as a superuser, or as root, to perform step 1 to step 4 on page 27.

1. Edit the file /etc/services. If you do not have the following line in that file, add it as shown:

```
MQSeries      1414/tcp      # MQSeries channel listener
```

2. Edit the file /etc/inetd.conf. If you do not have the following line in that file, add it as shown:

```
MQSeries stream tcp nowait mqm /opt/mqm/bin/amqcrsta amqcrsta
```

Note: If you are not creating venus.queue.manager as the default queue manager on this workstation, you need to add -m venus.queue.manager to the end of this line.

Verifying the Installation

3. Find the process ID of the inetd with the command:

```
ps -ef | grep inetd
```

4. Run the command:

```
kill -1 inetd processid
```

5. Create a default queue manager (in this example called `venus.queue.manager`):

- At the command prompt, type:

```
crtmqm -q venus.queue.manager
```
- Press Enter.

Messages are displayed telling you that the queue manager has been created, and that the default MQSeries objects have been created.

Note: In prior releases of MQSeries it was necessary to run a script file called `amqscoma.tst` to define the MQSeries default objects. This step is not required in this release of the product.

6. Start the queue manager:

- Type the following and then press Enter:

```
strmqm
```

A message tells you when the queue manager has started.

7. Enable MQSC by typing the following command:

```
runmqsc
```

Note: MQSC has started when the following message is displayed: Starting MQSeries Commands.

MQSC has no command prompt.

8. Define a local queue (in this example, called `ORANGE.QUEUE`):

- Type the following and press Enter:

```
define qlocal (orange.queue)
```

The message MQSeries queue created is displayed when the queue has been created.

9. Create a receiver channel:

```
define channel (first.channel) chltype (rcvr) trptype (tcp)
```

You have now defined the following objects:

- A default queue manager called `venus.queue.manager`
- A queue called `ORANGE.QUEUE`
- A receiver channel called `FIRST.CHANNEL`

10. Stop MQSC by pressing Ctrl-D or typing **end** and pressing Enter.

User Exits

Establishing communication between the Workstations:

1. If the queue managers on the two workstations have been stopped for any reason, restart them now (using the **strmqm** command).

2. On the **Sender** workstation start the sender channel:

```
runmqchl -c FIRST.CHANNEL -m saturn.queue.manager
```

The receiver channel on the receiver workstation is started automatically when the sender channel starts.

3. On the **Sender** workstation, use the `amqsput` sample program to send a message to the queue on the receiver workstation:

```
amqsput LOCAL.DEF.OF.REMOTE.QUEUE
```

Note: You put the message to the local definition of the remote queue, which in turn specifies the name of the remote queue.

4. Type the text of the message and press Enter **twice**.
5. On the **Receiver** workstation, use the `amqsget` sample program to get the message from the queue:

```
amqsget ORANGE.QUEUE
```

The message is displayed.

The verification is complete.

User Exits

You **must** relink all your user exits with threaded libraries before you use them on this version of the product, to make them thread-safe.

See Chapter 6, "Introducing the Message Queue Interface" in the *MQSeries Application Programming Guide* for further details on threaded libraries.

Setting the Queue Manager CCSID on MQSeries for Sun Solaris

The coded character set identifier (CCSID) is fixed when the queue manager is created. The CCSID used is the one for the code set, of the locale, that you are using to run the `crtmqm` command.

Examples of setting the CCSID:

```
export LANG=en_US.IS08859-1
  uses the code set IS08859-1
  and will set a CCSID of 819
export LANG=pl_PL.IS08859-2
  uses the code set IS08859-2
  and will set a CCSID of 912
```

To modify an existing queue manager CCSID, follow this procedure:

1. Record the existing queue manager CCSID, using the MQSeries (MQSC) command:
`DISplay QMGR CCSID`
2. Change the CCSID to the new CCSID, with the MQSC command:
`ALTer QMGR CCSID`
3. Stop the queue manager.
4. Restart the queue manager and any channels it uses.

Note: The `ALTer QMGR CCSID` command is a new command supplied with MQSeries for Sun Solaris V5.1. See Appendix C, "Support for Different Code Sets on MQSeries for Sun Solaris" on page 107 for further information about supported code sets. See "Migration to Euro Support" on page 110 for information on migrating to a CCSID that supports the euro character.

Queue Manager CCSID

Chapter 4. Planning to Install MQSeries Clients

This chapter is a summary of the hardware and software required to run the following MQSeries clients:

- MQSeries client for AIX®
- MQSeries client for DOS
- MQSeries client for HP-UX
- MQSeries client for OS/2 Warp
- MQSeries client for Sun Solaris
- MQSeries client for Windows® 3.1
- MQSeries client for Windows 95 and Windows 98
- MQSeries client for Windows NT®

All of these clients can be installed from the MQSeries Client CD. The Client CD also contains other MQSeries components that might need to be installed on a workstation other than the server.

Applications on Version 5.1 clients

A Version 5 client can connect to all queue managers, non-version 5 as well as Version 5. If you are connecting to a non-version 5 queue manager you cannot use the new Version 5.1 features and structures in your MQSeries application on the client.

Migrating from MQSeries Version 2 Products to Version 5.1

The internal format of the channel definition table has been changed in the MQSeries Version 5 products for AIX, HP-UX, OS/2 Warp, Sun Solaris, and Windows NT. A Version 5.1 client will continue to work with a Version 2 client channel definition table (by default, the client channel definition table is called AMQCLCHL.TAB). However, a Version 2 client cannot read a Version 5.1 client channel definition table. If your client channel definition table is produced by a Version 5.1 server, you must use Version 5.1

Hardware and Software, AIX

clients. In this case, reinstall the MQSeries client code from an MQSeries Version 5.1 server. In the case of a DOS MQSeries client, also relink your application.

AIX Client: Hardware and Software Required

This section outlines the hardware and software requirements for an MQSeries client for AIX.

Hardware

An MQSeries client can run on an IBM RISC System/6000®, capable of running AIX V4.2 or later. Any other trademarked AIX system may be used, whether from IBM or other vendors such as Bull, Zenith, or Motorola. There must be enough random access memory (RAM) and disk storage for the programming prerequisites (below), the MQSeries client code, the access methods, and the application programs.

Software

The following are prerequisites for MQSeries applications running on an AIX client.

Minimum supported software levels are shown. Later levels, if any, will be supported unless otherwise stated.

- AIX V4.2 (5765-655 or 5765-C34) or AIX V4.3 or later.

Later levels of some listed products may be required for AIX V4.2, SMP, and SP™. Later levels of operating system may be required to support corequisite products.

Connectivity

For TCP/IP connectivity:

- TCP/IP (in the operating system)

For SNA LU 6.2 connectivity:

- IBM eNetwork™ Communications Server for AIX, V5.0

Optional software

- IBM DCE V2.2. This must be the U.S. Domestic version supporting DES encryption if the user wishes to run the MQSeries-supplied DCE send, receive, or message exits.

Components for AIX

MQSeries DCE names and security modules are provided as part of the MQSeries for AIX product.

Compilers for MQSeries Applications on AIX Clients

The following compilers are supported:

- IBM C for AIX V4.4
- IBM C Set++ for AIX V3.1 (5765-421)
- IBM C and C++ compiler V3.6
- IBM VisualAge C++ Professional for AIX V4.0
- Micro Focus COBOL Compiler for UNIX V4.0
- IBM COBOL Set for AIX V1.1
- IBM PL/I Set for AIX V1.1
- IBM VisualAge Java Enterprise Edition for AIX V2.0

Components for AIX

MQSeries Client

The MQSeries client code for your UNIX platform.

Samples

Sample application programs.

Support for DCE in Samples

The DCE samples support. This should be installed if, and only if, you are going to use DCE.

Runtime component

Support for external applications. This does **not** enable you to write your own applications.

Base

Support to enable you to create and support your own applications. Requires the runtime component to be installed.

MQSeries Client for Java

This allows Java applets running on your client machine to communicate with MQSeries. It includes security exits for encryption and authentication of messages sent across the Web by the MQSeries Client for Java. These exits consist of some Java classes. To use the client for Java you need to have Java runtime code on your machine, at the following (or later compatible) levels:

Hardware and Software, DOS

AIX	Java Version 1.1.1
HP-UX	Java Version 1.1.2
Sun Solaris	Java Version 1.1.1

For information about Java runtime see the *MQSeries Using Java* book.

Note: If it is possible on your platform, at installation time the CLASSPATH environment variable will either get updated if already present or created if not.

MQSeries Internet Gateway documentation

MQSeries Internet Gateway documentation supplied in HTML format.

MQSeries Internet Gateway

Provides access to MQSeries applications via HTML and CGI (only CGI on the Sun Solaris platform). The MQSeries Internet Gateway does not support NSAPI on the HP-UX platform.

DOS Client: Hardware and Software Required

This section outlines the hardware and software requirements for an MQSeries client for DOS.

Hardware

An MQSeries client can run on DOS, on a personal computer. There must be enough random access memory (RAM) and disk storage for the programming prerequisites (below), the MQSeries client code, the access methods, and the application programs.

Software

The following are prerequisites for MQSeries applications running on a DOS client.

Minimum supported software levels are shown. Later levels, if any, will be supported unless otherwise stated.

- DOS V5.0 or later

Connectivity

- TCP/IP (in the operating system)
- SPX

Hardware and Software, HP-UX

Optional software

- Novell Netware Client for DOS/Win, V1.20 and V2.5
- Novell LAN Workplace, V5.1
- FTP PC/TCP for DOS, V5.0

The DOS access kit allows clients access to TCP/IP via programs that run in a DOS window under WIN-OS/2®.

The Novell Netware Client for OS/2 allows clients access to SPX via programs that run in a DOS window under WIN-OS/2.

Compilers for MQSeries Applications on DOS Clients

The following C compiler is supported:

- Microsoft Visual C++ V1.5

Components for DOS

MQSeries Client

The MQSeries client code for your platform.

MQSeries Toolkit

This includes:

- Sample programs - some of these are required for verifying the installation of the MQSeries client/server system
- Header files that you can use when writing applications to run on the client

HP-UX Client: Hardware and Software Required

This section outlines the hardware and software requirements for an MQSeries client for HP-UX.

Hardware and Software, HP-UX

Hardware

An MQSeries client can run on HP-UX on any HP 9000 Series 700 or Series 800 or Stratus Continuum/400 machine with minimum system disk space of 20 MB.

Software

The following are prerequisites for MQSeries applications running an HP-UX client.

Minimum supported levels are shown. Later levels, if any, will be supported unless otherwise stated.

- HP-UX Version 10.20
- HP-UX Version 11

Connectivity

For TCP/IP connectivity:

- TCP/IP (in the operating system)

For SNA connectivity:

- HP SNAplusII

Optional software

- The HP DCE/9000 version appropriate for the level of the HP-UX operating system in use.
- MQSeries DCE names and security modules are provided as part of the MQSeries for HP-UX product.

Compilers for MQSeries Applications on HP-UX Clients

The following compilers are supported:

- Micro Focus COBOL for UNIX Version 4.0
- HP-UX ANSI C compiler
- C Softbench Version 5.0
- HP CFRONT C++ Version 3.1
- IBM C and C++ compilers, Version 3.6
- HP C FRONT C++ Version 3.1
- HP ANSI C++

Components for HP-UX

These are identical to those for AIX. See “Components for AIX” on page 33.

OS/2 Warp Client: Hardware and Software Required

This section outlines the hardware and software requirements for an MQSeries client for OS/2.

Hardware

An MQSeries client can run on OS/2 Warp, on a personal computer. There must be enough random access memory (RAM) and disk storage for the programming prerequisites (below), the MQSeries client code, the access methods, and the application programs.

The system unit must have a CD-ROM device.

Software

The following are prerequisites for MQSeries applications running an OS/2 client.

This is the minimum supported software level. Later levels, if any, will be supported unless otherwise stated.

- OS/2 Warp V4.0 (84H1426) (5622-851)
- OS/2 Warp Server, V4.0 (25H8002)
- OS/2 Warp Server Advanced, SMP feature V4.0
- OS/2 Workspace-on-Demand

Connectivity

- IBM eNetwork Communications Server for OS/2 Warp V5.0
- Novell Netware Client for OS/2, V1.20 (for direct IPX/SPX support)
- IntraNetWare Client for OS/2, V2.12
- TCP/IP for OS/2 V2.0 base kit plus NetBIOS kit, V3.5 (base kit is essential)
- NetWare for OS/2 V4.11

Components for OS/2 Warp

Optional software

- IBM Directory and Security Server for OS/2 Warp Version 4 or later compatible versions. This must be the U.S. Domestic version supporting DES encryption if the user wishes to run the MQSeries-supplied DCE send, receive, or message exits.
- If used as a DCE server this software is known to run adequately in the following environment:
 - On a Pentium® processor running 90 MHz or faster.
 - On a machine with 64 MB or more of memory.
 - Using OS/2 Warp Server V4.0 or later.
- MQSeries DCE names and security modules are provided as part of the MQSeries for OS/2 Warp product.

Compilers for MQSeries Applications on OS/2 Clients

The following compilers are supported:

- IBM VisualAge® COBOL for OS/2 V1.1 (28H2177, 5622-793 (EMEA)) and V2.2
- Micro Focus COBOL compiler Version V4.0 (32 bit)
- IBM VisualAge C++ for OS/2 Version 3.0
- Borland C++ compiler Version 2.0 and V5.02 (C bindings only)
- IBM C and C++ compilers, V3.6
- IBM PL/I for OS/2 Version 1.2
- IBM VisualAge for PL/I for OS/2
- IBM VisualAge Java Enterprise Edition for OS/2 Version 2.0
- IBM VisualAge Java Professional Edition for OS/2 Version 2.0

Components for OS/2 Warp

MQSeries Client

The MQSeries client code for your platform.

MQSeries Toolkit

This includes:

- Sample programs
- Header files that you can use when writing applications to run on the client

MQSeries Client for Java

This allows Java applets running on your client machine to communicate with MQSeries. It includes security exits for encryption and authentication of messages sent across the Web by the MQSeries Client for Java. These exits

Hardware and Software, Sun Solaris

consist of some Java classes. To use the client for Java you need to have Java 1.1.1 (or later compatible version) runtime code on your machine. On Windows NT, you require Microsoft Windows NT Version 4. On OS/2 the MQSeries client for Java must be installed on an HPFS formatted drive. For information about Java runtime see "Available from the Internet" on page 100.

Note: If it is possible on your platform, at installation time the CLASSPATH environment variable will either get updated if already present or created if not.

MQSeries Internet Gateway documentation

MQSeries Internet Gateway documentation supplied in HTML format.

MQSeries Internet Gateway

Provides access to MQSeries applications via HTML and CGI.

Sun Solaris Client: Hardware and Software Required

This section outlines the hardware and software requirements for an MQSeries client for Sun Solaris.

Hardware

An MQSeries client can run only on:

- Sun SPARC desktop or server
- Sun UltraSPARC desktop or server

with a minimum system disk space of 25 MB.

Note: Solaris systems from other manufacturers are not supported.

For connectivity, any communications hardware supporting SNA LU 6.2 or TCP/IP.

Software

The following are prerequisites for MQSeries applications running on a Sun Solaris client.

Minimum supported software levels are shown. Later levels, if any, will be supported unless otherwise stated.

- Sun Solaris V2.6 with patches 105210-13 and 105568-10, and Sun Solaris 7.

Hardware and Software, Windows 3.1

Connectivity

- SunLink SNA Peer-to-Peer Version 9.1
- TCP/IP as part of the base operating system
- If token ring is to be used: SunLink Token Ring Interface /SBus, V3.0.2. This requires patch 102463 or Sun TRI/P Adapter V1.0

Optional software

Transarc DCE, V1.1. This must be the U.S. version supporting DES encryption for users to run the MQSeries-supplied DCE send, receive or message exits.

DCE names and security modules for MQSeries are provided as part of the MQSeries for Sun Solaris product.

Compilers for MQSeries Applications on Sun Solaris Clients

The following compilers are supported:

- Sun WorkShop Compiler C, Version 4.2
- Sun WorkShop Compiler C++, Version 4.2

Components for Sun Solaris

These are identical to those for AIX. See "Components for AIX" on page 33.

Windows 3.1 Client: Hardware and Software Required

This section outlines the hardware and software requirements for an MQSeries client for Windows 3.1.

Hardware and Software, Windows 3.1

Hardware

An MQSeries client can run on Windows 3.1, on a personal computer. There must be enough random access memory (RAM) and disk storage for the programming prerequisites (below), the MQSeries client code, the access methods, and the application programs.

Software

The following are prerequisites for MQSeries applications running on a Windows 3.1 client.

Minimum supported software levels are shown. Later levels, if any, will be supported unless otherwise stated.

- Windows 3.1
- Windows 95 in 16-bit mode
- Windows for Workgroups

Connectivity

- TCP/IP
- SPX

Optional software

- Novell Netware client for DOS/Win31 V1.20
- SunPC NSF V5.1
- OnNet SDK for Windows

Compilers for MQSeries Applications on Windows 3.1 Clients

The following compiler is supported:

- Microsoft Visual C++ Version 1.5

Components for Windows 3.1

These are identical to those for DOS. See “Components for DOS” on page 35.

Windows 95 and Windows 98 Client: Hardware and Software Required

This section outlines the hardware and software requirements for an MQSeries client for Windows 95 and Windows 98.

Hardware

An MQSeries client can run on Windows 95 or Windows 98 on a personal computer. There must be enough random access memory (RAM) and disk storage for the programming prerequisites (below), the MQSeries client code, the access methods, and the application programs.

Software

The following are prerequisites for MQSeries applications running on a Windows 95 and Windows 98 client.

Minimum supported software levels are shown. Later levels, if any, will be supported unless otherwise stated.

- Windows 95 or Windows 98

Connectivity

- TCP/IP (in the operating system)
- SPX (in the operating system)
- NetBIOS (in the operating system)

Components for Windows 95 and Windows 98

Optional software

- IBM DCE for Windows 95, Version 1.1

A DCE security module is provided as part of the MQSeries client for Windows 95 and Windows 98. IBM DCE for Windows 98 does not currently support DES encryption, so you cannot run the DCE send, receive, or message exits supplied by MQSeries

Transaction monitors

- BEA Tuxedo Versions 5.1, 6.1, and 6.4

Compilers for MQSeries Applications on Windows 95 and Windows 98 Clients

The following compilers are supported:

- Micro Focus COBOL Workbench Version 4.0
- IBM VisualAge C++ for Windows Version 3.5 (33H4979)
- Microsoft Visual C++ for Windows 95/NT Version 4.0 and Version 5.0

Components for Windows 95 and Windows 98

MQSeries Client

The MQSeries client code for Windows 95 and Windows 98.

MQSeries Toolkit

This includes:

- Sample programs
- Header files that you can use when writing applications to run on the client

Windows NT Client: Hardware and Software Required

This section outlines the hardware and software requirements for an MQSeries client for Windows NT.

Hardware

An MQSeries client can run on Windows NT on any Intel® 486 processor based IBM PC machine or equivalent (certified as NT compatible). There must be enough random access memory (RAM) and disk storage for the programming prerequisites (below), the MQSeries client code, the access methods, and the application programs.

Software

The following are prerequisites for MQSeries applications running on a Windows NT client.

Minimum supported software levels are shown. Later levels, if any, will be supported unless otherwise stated.

- Microsoft Windows NT V4 or later compatible version, to include TCP/IP, NetBIOS, SNA LU 6.2, and SPX.

Connectivity

- IBM Communications Server for Windows NT, V5.0 (4231747)
- Attachmate Extra! Personal Client, V6.1 and V6.2
- Microsoft SNA Server, V2.11 or V3
- TCP/IP, NetBIOS, and SPX are part of the base operating system

Optional software

- IBM DCE V1.1

Compilers for MQSeries Applications on Windows NT Clients

The following compilers are supported:

- IBM VisualAge COBOL for Windows NT V2.1
- IBM VisualAge COBOL Enterprise V2.2
- IBM VisualAge C++ Version 3.5 (33H4979)
- Microsoft Visual C++ for Windows 95 and Windows NT, V4.0 and V5.0
- IBM C and C++ compilers, V3.6
- IBM PL/I for Windows V1.2
- IBM VisualAge for PL/I for Windows
- IBM VisualAge PL/I Enterprise, V2.1
- IBM Visual Basic for Windows, V4.0 (16-bit) and V5.0 (32-bit) or higher
- IBM VisualAge for Java, e-Business Edition for Windows 95 and Windows NT
- IBM VisualAge Java Enterprise Edition V2.0
- IBM VisualAge Java Professional Edition V2.0

Components for Windows NT

These are identical to those for OS/2 Warp. See "Components for OS/2 Warp" on page 38.

Components for Windows NT

Chapter 5. Installing MQSeries Clients

This chapter describes the following:

- “Installing on AIX”
- “Installing on DOS” on page 54
- “Installing on HP-UX” on page 55
- “Installing on OS/2 Warp” on page 56
- “Installing on Sun Solaris” on page 65
- “Installing on Windows 95 or Windows 98” on page 67
- “Installing on Windows 3.1” on page 71
- “Installing on Windows NT” on page 73

Installing on AIX

To install an MQSeries client on an AIX system you use the MQSeries Client CD-ROM supplied as part of the MQSeries product.

The MQSeries client is installed into the `/usr/mqm` directory. This **cannot** be changed. However, if you do not have enough space in the `/usr/mqm` file system, follow the procedure given in “Creating Another File System for Client Code” on page 48.

If you have a previous version of the MQSeries for AIX client installed on your system, or if a file system remains from a previous AIX client installation, see “Migrating from an Earlier Version of MQSeries for AIX” on page 51.

If you plan to install an MQSeries client and server on the same machine, see “Installing the Server and Client on the Same Machine” on page 21.

Before Installation

Before you can install the MQSeries client on your AIX system you are advised to create and mount a `/var/mqm` file system, or `/var/mqm`, `/var/mqm/log`, and `/var/mqm/errors` file systems.

If you create separate partitions, the following directories **must** be on a local file system:

- `/var/mqm`
- `/var/mqm/log`

Installing on AIX

You can choose to NFS mount the `/var/mqm/errors` directory to conserve space on your local system.

A user ID with the name `mqm` whose primary group is `mqm` is automatically created during installation. You can create the user and group IDs yourself (see “Creating the `mqm` User ID and Group”), but make sure you do this before installing the client. User ID and group must both be `mqm`.

For stand-alone machines, you can create the new user and group IDs locally. For machines administered in a network information services (NIS) domain, you can create the user and group IDs on the NIS master server machine.

After installation, the `mqm` user ID owns the directories and files that contain the resources associated with the product. This group and user must be defined for any machine on which the MQSeries software is to be installed, whether the machine is a client or a server machine.

Creating Another File System for Client Code

If you do not want to have the product code installed in the `/usr/mqm` file system, you can do either of the following things:

1. Create a new file system and mount it as `/usr/mqm`.

or

2. Create a new directory anywhere on your machine that is large enough to contain the client files, and create a symbolic link from `/usr/mqm` to this new directory. For example:

```
mkdir /bigdisk/mqm
ln -s /bigdisk/mqm /usr/mqm
```

Whichever of these options you choose, you **must** do it before installing the client code.

The file system into which the code is installed can be a remote network device, for example NFS, provided that the mount options are defined on that device to allow **setuid** programs (including root access) to be run.

Creating the `mqm` User ID and Group

If you want to create the required IDs yourself, follow this procedure *before you install the client*. You must create both user ID and group as `mqm`.

Create the new IDs using the System Management Interface Tool (SMIT), for which you require root authority. The procedure for this, if you use the SMIT windows, is:

1. Create the `mqm` group. You can display the required window using this sequence:

Installing on AIX

Security & Users
Groups
Add a Group

You can take the default values for the attributes of the new group or change them as required.

2. Create the new user, mqm. You can display the window for doing this using this sequence:

Security & Users
Users
Add a User

Set the primary group for this user to be mqm. You can take the default values for the attributes of the new group or change them if you wish.

3. Add a password to the new user ID. You can display the window for doing this using this sequence:

Security & Users
Passwords

4. Add the newly created group mqm to an existing user ID. You can display the window for doing this using this sequence:

Security & Users
Users
Change / Show Characteristics of a User

When the window is displayed, enter the name of the user who is to have the mqm group added. In the user details window, add mqm to the **Group set** field, which is a comma-separated list of the groups to which the user belongs.

Note: You need not have your primary group set to mqm. As long as mqm is in your set of groups, you can use the commands. If you are running applications that use the queue manager only, you do not need mqm group authority.

Easy Installation

This section describes the 'Easy Installation' procedure using the MQSeries Client CD-ROM.

Note: If you have a previous version of the MQSeries for AIX client installed on your system, or if a file system remains from a previous AIX client installation, see "Migrating from an Earlier Version of MQSeries for AIX" on page 51.

1. Logon as root.
2. Insert the MQSeries Client CD-ROM into the CD-ROM drive.
3. Type `xinstallm -ez`

The MQSeries Welcome window is displayed.

4. **Make sure you are installing the correct client** for your system, as displayed in the Welcome window.

Installing on AIX

A window is then displayed where you can make some choices.

5. Choose the software source: **CD-ROM**.
6. For **Which bundle of software would you like to install?** choose: **Media-defined**.
7. Click on **Install/Update**.

A bundle of software products is created:

```
mqm.Client
```

See “Components for AIX” on page 33 for details of the components.

8. Choose the `mqm.Client` bundle and click on **Install/Update** again.
A work in progress window gives information as the installation proceeds.
9. At the end of installation you can click on the **View log** button and scroll to the bottom of the log to see the filesets that have been installed successfully.

Custom Installation

This section describes custom installation using the System Management Interface Tool (SMIT).

Note: If you have a previous version of the MQSeries for AIX client installed on your system, or if a file system remains from a previous AIX client installation, see “Migrating from an Earlier Version of MQSeries for AIX” on page 51.

You can use SMIT for a custom installation as follows:

1. Logon as root.
2. Go into SMIT and from the shell, type:

```
smit
```
3. Select the device appropriate for your installation using this sequence of windows:

```
Software Installation and Maintenance
  Install and Update Software
    Install and Update from Latest Available Software
```

Alternatively, you can use the fastpath command:

```
smitty install_latest
```

4. Press the **List** button to display the **Single Select List** window.
5. Select:
/dev/cd0 (CD-ROM Drive)
6. Press **Do** to display the parameters for **Install Latest Level**.
7. Press **F4** to get a list of components to install.

Installing on AIX

8. Follow the SMIT instructions to select the components you want to install. See “Components for AIX” on page 33 for details.
9. Press **Enter**.
10. If you have a previous version of the product on your machine, change the **Auto Install prerequisite software** to **No** and **Overwrite existing version** to **Yes**.
11. Press **Do** to install.

Migrating from an Earlier Version of MQSeries for AIX

If you want to migrate from an MQSeries for AIX V5.0 client to an MQSeries for AIX V5.1 client, you must first end all MQSeries activity on the target machine, and remove any shared resources that are used by MQSeries. Do this either by shutting down the system and restarting it, or by issuing the **icprm** command to remove the shared resources.

Note: The migration procedure described in this section applies only to migration from MQSeries for AIX V5.0 clients to MQSeries for AIX V5.1 clients. If you are migrating from an earlier version of MQSeries for AIX, you are advised to uninstall your current version before installing the MQSeries for AIX V5.1 client.

Migration from MQSeries for AIX V5.0 involves updating any currently installed filesets, and installing any new filesets that may be required.

To update currently installed filesets:

1. Go into SMIT for root authority. From the shell, enter:

```
smit
```
2. Select the device appropriate for your installation using the following sequence of windows:

```
Software Installation and Maintenance
  Install and Update Software
    Update Installed Software to Latest Level (Update All)
```

Alternatively, you can use the fastpath command to select the appropriate device:

```
smitty update_latest
```
3. Select the **List** button to display the Single Select List window
4. Select **/dev/cd0 (CD-ROM Drive)**
5. Select **OK** to display the parameters for **Update All**
6. Update all previously installed software for MQSeries by selecting the **_update_all** option in the **Software to update** field
7. Press Enter
8. Select **OK** in the confirmation window to start updating the software

Installing on AIX

Once all previously installed filesets have been updated to the latest level, you can install any additional filesets. See “Custom Installation” on page 50 for more information.

Changes to the Installation Path

Changes in AIX LPP Version 4 packaging mean that the MQSeries for AIX V5.1 client installs into directory **/usr/mqm**, whereas previous versions of the product installed into directory **/usr/lpp/mqm**.

Installation of the MQSeries for AIX V5.1 client will fail if a file system mounted as **/usr/lpp/mqm** is detected. If you are migrating from an earlier version and a file system exists for this directory, you will need to do one of the following things before installing the MQSeries for AIX V5.1 client. Either:

- Uninstall your existing MQSeries client, and either delete the file system or move it to the new install path of **/usr/mqm**

or

- Move the old file system of **/usr/lpp/mqm** to the new installation path of **/usr/mqm** and create a symbolic link from the old path to the new by issuing the following command:

```
ln -s /usr/mqm /usr/lpp/mqm
```

If you uninstall your existing client and either delete or move your existing file system, you can then install the MQSeries for AIX V5.1 client as described in “Custom Installation” on page 50 or “Easy Installation” on page 49.

However, if you move the old file system to the new installation path, you should then perform the migration installation described in “Migrating from an Earlier Version of MQSeries for AIX” on page 51.

Note: If you have already symbolically linked a file system to **/usr/lpp/mqm**, installation of the MQSeries for AIX V5.1 client will destroy the contents of the file system and the symbolic link, leaving an empty file system. If this happens, you are advised to uninstall your existing MQSeries client and either delete the file system or relink it to the new install path of **/usr/mqm**, before installing the MQSeries for AIX V5.1 client.

The installation process for the MQSeries for AIX V5.1 client creates a symbolic link from the old install path (**/usr/lpp/mqm**) to the new install path (**/usr/mqm**). Therefore any existing scripts or makefiles that reference the old path are still valid.

Changing the National Language

The easy installation and the custom installation default to the national language that was specified when your operating system was installed.

It is possible to install the MQSeries client software so that the online help and messages are in another national language. Use SMIT as follows to install the message catalog for another national language:

1. Type `smit`
2. Follow this sequence of windows:
 - Software Installation and Maintenance
 - Install and Update Software
 - Install and Update from ALL Available Software
3. Press the **List** button to display the **Single Select List** window.
4. Select:
 - /dev/cd0 (CD-ROM Drive)**
5. Press the **List** button on the **Software to Install** field.
6. Select the message catalog that you want to install.
7. Press **Do** to install the chosen message catalog or catalogs.

To check the initial locale setting for your machine type:

```
smitty m1e_cc_cust_hdr
```

and press Enter. If this is not one of the national languages provided by MQSeries, you need to select a national language, otherwise you will not get a message catalog installed on your system.

Translated Messages

Messages in U.S. English are always available. If you require another of the languages that is supported by MQSeries for AIX, you **must** ensure that your NLSPATH environment variable includes the appropriate directory.

For example, to select messages in German use the following:

```
export LANG=de_DE
export NLSPATH=/usr/lib/nls/msg/%L/%N
```

Installing on DOS

Removing an MQSeries Client from AIX

Use SMIT as usual to remove all the MQSeries client files that were installed.

Installing on DOS

To install an MQSeries client on a DOS system you use the MQSeries Client CD-ROM supplied as part of the MQSeries product.

Using Setup

1. Insert the MQSeries Client CD-ROM into the CD-ROM drive.
2. Change to the DOS directory on the CD-ROM drive.
3. Copy the **setup.exe** file from the DOS directory to the directory where you want to install the MQSeries client, for example `c:\mqmdos`.
4. Change directory to `c:\mqmdos` and type the command:

```
setup -d
```

This results in a self-exploding file being run to generate a tree of subdirectories containing the DOS client. See "Components for DOS" on page 35 for details.

5. Edit the `autoexec.bat` file using a suitable editor. If the `PATH` statement exists, add the following to it:

```
c:\mqmdos;c:\mqmdos\bin;c:\mqmdos\en_us
```

If the `PATH` statement does not exist, add the following line to the `autoexec.bat` file.

```
SET PATH=c:\mqmdos;c:\mqmdos\bin;c:\mqmdos\en_us
```

Removing an MQSeries Client from DOS

Delete all the files in the directory where you installed the MQSeries client, and then remove the directory.

Installing on HP-UX

To install an MQSeries client on an HP-UX system you use the MQSeries Client CD-ROM supplied as part of the MQSeries product.

Note: If you plan to install an MQSeries client and server on the same machine, see “Installing the Server and Client on the Same Machine” on page 21.

The MQSeries client is installed into the `/opt/mqm` directory. This **cannot** be changed.

Before Installation

Before you can install an MQSeries client on your HP-UX system you:

- Must create a group with the name `mqm`.
- Must create a user ID with the name `mqm`.
- Are recommended to create and mount a `/var/mqm` file system, or `/var/mqm`, `/var/mqm/log`, and `/var/mqm/errors` file systems.

If you create separate partitions, the following directories **must** be on a local file system:

- `/var/mqm`
- `/var/mqm/log`

You can choose to NFS mount the `/var/mqm/errors` and `/var/mqm/trace` directories to conserve space on your local system.

After installation, this user ID (`mqm`) owns the directories and files that contain the resources associated with the product. This group and user must be defined for any machine on which the MQSeries software is to be installed, whether the machine is a client or a server machine.

For stand-alone machines, you can create the new user and group IDs locally. For machines administered in a network information services (NIS) domain, you can create the user and group IDs on the NIS master server machine.

Installation

Use the HP-UX **swinstall** program, or use SAM, after mounting the CD-ROM. For further details see the *HP-UX Administration Guide*. See also “Components for AIX” on page 33.

Installing on OS/2 Warp

If you are using HP-UX V10.x, the depot to use is in the HPUX10/MQS510.000.V10 file under the mount point. If you are using HP-UX V11.x, the depot to use is in the HPUX11/MQS510.000.V11 file under the mount point.

If the files on your CD-ROM appear in uppercase with a “;1” suffix, use this name for the depot.

Kernel Configuration

See the MQSeries family Web site for a SupportPac™ that gives additional performance information - see “Available from the Internet” on page 100.

Translated Messages

Messages in U.S. English are always available. If you require another of the languages that is supported by MQSeries for HP-UX, you **must** ensure that your NLSPATH environment variable includes the appropriate directory.

For example, to select messages in German use the following:

```
export LANG=de_De.iso88591
export NLSPATH=/usr/lib/nls/msg/%L/%N
```

Removing an MQSeries Client from HP-UX

To remove an MQSeries client from your HP-UX system, use the **swremove** command, or use SAM. You can then delete the /var/mqm directory tree.

Installing on OS/2 Warp

To install an MQSeries client on an OS/2 Warp system you use the MQSeries Client CD-ROM supplied as part of the MQSeries product.

Note: If you plan to install an MQSeries client and server on the same machine, see “Installing the Server and Client on the Same Machine” on page 21. If you currently have a manually installed MQSeries client on your OS/2 Warp system from a previous release of MQSeries, you must manually delete it before attempting to install the Version 5.1 client. You must not install the Version 5.1 client onto a system which currently has a Version 2 MQSeries Server installed.

Installing on OS/2 Warp

You can install the version of the MQSeries client software specific to your national language. This means that the installation program, online help and messages will be in your national language.

Installation

Online help is available by selecting the **Help** push button or by pressing PF1.

Before you start, make sure that you have at least 150 KB of free space on the drive containing the operating system. This is required by the installation program.

1. Open an OS/2 window (or start a full-screen session).
2. Insert the CD-ROM and change to the CD-ROM drive. Access the drive and directory containing the installation program if you are installing from a remote drive:
3. At the command prompt, in the root directory, type `INSTALL`, then press Enter.
4. On the MQSeries Language Selection panel select the language of your choice, and click on the **OK** button or press Enter.

The MQSeries Welcome panel is displayed. **Make sure you are installing the correct client** for your system, OS/2 Warp, as displayed in the Welcome panel.

5. The install panel is then displayed. Select the **Update CONFIG.SYS** check box if you want your CONFIG.SYS file updated automatically as part of the installation process. Your original CONFIG.SYS file is renamed to CONFIG.BAK and is stored in the same directory. If you do not select this check box, a CONFIG.ADD file is generated. This file is a copy of CONFIG.SYS with the necessary updates to the LIBPATH and PATH statement. You can rename the CONFIG.ADD file to CONFIG.SYS.
6. Select the **OK** push button to continue. The Install - directories panel is displayed.
7. The list box shows the installation options that you can select. When you select one or more of these options, the **Bytes needed** field shows the amount of disk space required for installation. See "Components for OS/2 Warp" on page 38 for details.

The component **MQSeries Client for Java** should be installed only if you have Java 1.1.1 (or later compatible) runtime code on your machine. Also this component must be installed on an HPFS formatted drive.

8. If there is not enough space on your hard disk to install all the components, select an option that uses less disk space. If there is too little space on your hard disk for any of the MQSeries for OS/2 installation options, a dialog box appears before the Install - directories panel. In this case, cancel the installation by selecting the **OK** push button. Find out which of your existing files you can archive or delete to make more space before proceeding further.

Use the push buttons as necessary:

- To display descriptions of the selected options, select **Descriptions**.

Installing on OS/2 Warp

- To select all of the options, select **Select all**.
- To deselect all of the options, select **Deselect all**.

The **Work and File Directory** field allows you to specify a drive and directory other than the default for the installation files (File directory) and for the working files that may be created when you use the MQSeries client (Working directory).

Select a drive from the list box if required. When you return to the Install - directories panel, your selected drive is shown. Select the **OK** push button to return to the Install - directories panel.

9. Select the **Install** push button to continue. The Install-progress panel is displayed. The panel shows:

- The file currently being installed (source) and the drive and directory to which it is being installed (target).
- A progress bar, indicating the percentage of files already unpacked and installed.
- The elapsed time.
- The status; for example, unpacking, processing, or transferring.

If you select the **Stop** push button, you are asked whether you want to delete the partial system you have installed. Select **Yes** to delete the files already installed and return to the introductory panel. Then, select **Start install** from the **File** menu to start the installation again.

10. A cyclic redundancy check (CRC) is performed on the installed software and any errors are written to a log file. This is the file specified by the /L1 parameter of the INSTALL command by default. If /L1 is not specified, the log file is MQMERR.LOG in the high-level directory chosen for installation.

Note: The log files *must* be on a local drive. If the product has been installed on a remote drive, change the path of the log files in the mqs.ini file.

11. When installation is complete, the Installation and Maintenance panel is displayed. Select **OK**. The introductory MQSeries for OS/2 panel is then displayed. Leave the installation program by selecting the **Exit** push button.
12. When the installation process is complete, a folder is created on the OS/2 desktop, containing objects as follows:

```
READ.ME
MQSeries Installation and Maintenance
MQSeries Information
```

Note that the MQSeries client is a set of services and it does not have to be explicitly run. Therefore the folder does not have an object called a "client".

13. Remove the installation CD-ROM from the drive.
14. If your CONFIG.SYS file has been updated, shut down the system and restart. If the CONFIG.SYS file was not updated, rename the CONFIG.ADD file to CONFIG.SYS before shutting down the system. (CONFIG.ADD will be in the same directory as CONFIG.SYS.)

Unattended Installation on OS/2 Warp

You can install MQSeries clients on OS/2 Warp workstations without user interaction being required at the workstation. You can perform this unattended installation using response files. The sample response file AMQISMC2.RSP shows an example of unattended client installation.

This kind of automatic installation is particularly useful for installing clients over a network because installation can be performed from a redirected drive on a LAN server.

Attention

Note that the INSTALL.EXE file in the root directory of the CD-ROM **must not** be used to perform unattended installations.

You **must** instead invoke the INSTALL.EXE file located in the directory corresponding to the language version of the product that you want to install.

Starting Unattended Installation

The following steps describe how to perform an unattended installation:

1. Connect to the drive containing the MQSeries product software. This can be a local CD-ROM drive containing the MQSeries client CD, or a remote network drive. For example, F:.
2. Change to the directory appropriate to your language. The directories are as follows:

Directory	Language
\OS2\En_Us	U.S. English
\OS2\Pt_BR	Brazilian Portuguese
\OS2\Fr_FR	French
\OS2\De_DE	German
\OS2\It_IT	Italian
\OS2\Ja_JP	Japanese
\OS2\Ko_KR	Korean
\OS2\Es_ES	Spanish
\OS2\Zh_CN	Simplified Chinese
\OS2\Zh_TW	Traditional Chinese

3. Change to the subdirectory `samples`
4. Edit the supplied response file, AMQISMC2.RSP for client installation, or create a new response file. The supplied response file can be used for install and delete actions.
5. Type `install` together with the required parameters (see "Installation and Maintenance Parameters" on page 60). For example:

Installing on OS/2 Warp

```
INSTALL /A:I /R:J:\INSTMQS\AMQISMC2.RSP
        /S:J:\INSTMQS\en_us
        /L1:J:\INSTMQS\MQM.OUT
        /L2:J:\INSTMQS\MQM.HIS /X
```

Note: This example uses U.S. English. To use another language, substitute `en_us` for the directory appropriate to your language. When you issue this command, you must type it as one continuous line. In the example above, for clarity, the command is spread over several lines.

In this example, the MQSeries client is installed from the redirected drive J: on the server, according to the options supplied in the response file `AMQISMC2.RSP`. Note that drive J: must be a writeable drive that you must create, and the `.RSP` file must be put on that drive. The installation log files will also be created on the J: drive. The response file specifies the drive and directory in which to install the client. Errors are logged in `MQM.OUT` and the history log is contained in `MQM.HIS`. The `/X` parameter specifies that the installation is non-interactive.

The example also shows that you must have the `INSTALL.EXE` available to your system. This file is supplied on the client CD-ROM, but is not installed onto your system. You must make a copy of this file available when you perform maintenance updates on your system.

Installation and Maintenance Parameters

The installation and maintenance parameters are as follows:

```
INSTALL /A:action
        /G:include path
        /L1:error log
        /L2:history log
        /R:response file
        /S:source location
        /T:install target directory
        /TU:update CONFIG.SYS directory
        /X
```

Note: You can enter the parameters in any order. Equals signs (=) can be used instead of colons (:) in the parameters. Values can be upper or lower case.

/A:action

Specifies the action to be performed by the installation program. If you specify this parameter, the main window of the installation program is not displayed. Valid values for *action* are:

- D** Delete an installed MQSeries for OS/2 Warp system
- I** Install a new MQSeries for OS/2 Warp system
- R** Restore a backed up MQSeries for OS/2 Warp system

Installing on OS/2 Warp

U Update an installed MQSeries for OS/2 Warp system

Note: The installation program is not supplied with a corrective service medium. Therefore, if you are updating, restoring, or deleting an MQSeries for OS/2 Warp system, you must have access to the installation program used to install it.

/G:*include path*

Specifies the drive and path of a general response file to be included by the specific response file. For more information about response files, see "Installation Response Files" on page 62.

/L1:*error log*

Specifies the drive, path, and file name of the error log file. The error log contains messages associated with installation, including confirmations and error messages. Messages are written to the error log if you specify the /X parameter.

You should specify the drive and path where the installation program is running. If you do not specify the /L1 parameter, no error log is maintained. If the error log already exists, it is appended to.

Example:

```
/L1:D:\LOG\INSMQM.OUT
```

/L2:*history log*

Specifies the drive, path, and file name of the history log file. The history log contains an entry for each file transferred, each object created, and each installation exit run.

You should specify the drive and path where the install program is running. If you do not specify the /L2 parameter, no history log is maintained. If the history log already exists, it is appended to.

Example:

```
/L2:D:\LOG\INSMQM.HIS
```

/R:*response file*

Specifies the drive, path, and file name of a response file; see "Installation Response Files" on page 62.

Example:

```
/R:L:\MQMINS\AMQISAM2.RSP
```

/S:*source location*

Specifies the drive and path containing the source files to be installed or updated.

/T:*install target directory*

Specifies the drive and path that MQSeries for OS/2 Warp files are installed onto. If you specify this parameter, it overrides the FILE path, which must be specified in the response files.

/TU:*update CONFIG.SYS directory*

Specifies the drive and path of the target CONFIG.SYS to be updated.

Installing on OS/2 Warp

/X Specifies that the installation is fully automatic.

When you specify this parameter, no progress indicator panel is shown and error messages are logged in the error log file. (You specify the path name of the error log file using the **/L1** parameter.) If you do not specify all of the information required for the action to complete, an error occurs.

If you do not specify the **/X** parameter, the user is prompted for any information that the install program needs to complete the action. In this interactive mode of operation, progress indication is shown and error messages are displayed in secondary windows.

Installation Response Files

An installation response file is an ASCII text file containing answers to the options that you select when you install or maintain an MQSeries for OS/2 Warp system. This allows installation and maintenance to be performed automatically, without interaction.

In an installation response file you can specify:

- Whether CONFIG.SYS should be updated automatically
- The MQSeries for OS/2 Warp components to be installed
- The path for installation or maintenance
- Whether existing files should be overwritten
- Whether only backup versions of MQSeries for OS/2 Warp should be deleted

The installation response file supplied with MQSeries for OS/2 Warp can be found as a sample file on the installation media. You can use this file to install or delete MQSeries for OS/2 Warp. To perform other actions you must prepare your own installation response files using a suitable editor.

Normally you have only one response file, specified by the **/R** parameter of the installation program. However, if you need to install or update MQSeries for OS/2 Warp on a workstation with different options, you can use two files. One would be a specific response file that contains options specific to a particular workstation, and the other a general installation response file that contains options common to all workstations.

For example, you might use a general and a specific response file to install a particular component only on some workstations.

You use the **/R** parameter to specify the specific installation response file, and the **/G** parameter to specify the location of the general response file.

Keywords for Response Files

The following keywords are supported in response files for MQSeries for OS/2 Warp:

- CFGUPDATE** Specifies whether CONFIG.SYS is updated automatically. Valid values for this keyword are:
- AUTO** Automatically updates CONFIG.SYS
 - MANUAL** Does not update CONFIG.SYS
- Actions:** INSTALL, UPDATE, DELETE.
- COMP** Specifies the name of a component on which to perform an action. See the server or client component lists for valid values. You do not need to enclose the product names within quotes.
- Actions:** ALL actions.
- DELETEBACKUP** Specifies whether to delete only the backup version of MQSeries for OS/2 Warp or the entire product. Valid values are YES and NO.
- When you remove MQSeries for OS/2 Warp, none of your user-supplied information, for example queue manager data, is deleted. This is particularly important if you intend to delete and reinstall the product, because your previous queue manager data will remain.
- This can lead to an unexpected directory and file structure, if you attempt to recreate queue managers with identical names to those used in the previous installation, because such data may already be present.
- Actions:** DELETE.
- FILE** Specifies the drive and directory for MQSeries for OS/2 Warp code. For example the C:\MQM directory. This keyword applies to the installation action only.
- Actions:** INSTALL.
- WORK** Specifies the drive and directory for MQSeries data files. For example the C:\MQM directory. This keyword applies to the installation action only.
- Actions:** INSTALL.
- INCLUDE** Specifies which general response files to include with a specific response file. The format of this keyword is:
- INCLUDE = *filespec*
- Where *filespec* specifies the general response file to be included. If the file specification contains any global characters (* or ?), the first file found that matches the specification is

Installing on OS/2 Warp

included. If the specification is not valid, no general response file is included.

Note: You should not have more than five levels of included response files.

Actions: All actions.

OVERWRITE

Specifies whether to overwrite files automatically during installation. Valid values for this keyword are YES and NO.

Actions: INSTALL, UPDATE.

SAVEBACKUP

Specifies whether to save a backup version of MQSeries for OS/2 Warp when it is updated. Valid values for this keyword are YES and NO.

Actions: UPDATE.

Response File Structure

There are two kinds of line in a response file:

- **Comment lines**

Comment lines are either blank or start with an asterisk (*) or a semicolon (;).

- **Response lines**

Response lines are used to determine the options and configurations to install on the target system.

Response lines have the following syntax:

keyword = value

Keyword-value pairs can be in any order. However, there can be only one pair per line.

You can enter keywords in upper or lower case letters. You cannot include spaces within keywords.

The maximum line length in a response file is 255 characters.

Removing an MQSeries Client from OS/2

Use the MQSeries Installation and Maintenance icon in the MQSeries Client folder on the desktop, and select Actions/Delete. All the MQSeries client files that were there at the time of installation are deleted.

Installing on Sun Solaris

To install an MQSeries client on a Sun Solaris system you use the MQSeries Client CD-ROM supplied as part of the MQSeries product.

Note: If you plan to install an MQSeries client and server on the same machine, see “Installing the Server and Client on the Same Machine” on page 21.

The MQSeries product is installed into the **/opt/mqm** directory. This **cannot** be changed.

Before Installation

Before you can install an MQSeries client on your Sun Solaris system you:

- Must create a group with the name `mqm`.
- Must create a user ID with the name `mqm`.
- Are recommended to create and mount a `/var/mqm` file system, or `/var/mqm`, `/var/mqm/log`, and `/var/mqm/errors` file systems.

If you create separate partitions, the following directories **must** be on a local file system:

- `/var/mqm`
- `/var/mqm/log`

You can choose to NFS mount the `/var/mqm/errors` and `/var/mqm/trace` directories to conserve space on your local system.

After installation, this user ID (`mqm`) owns the directories and files that contain the resources associated with the product. This group and user must be defined for any machine on which the MQSeries software is to be installed, whether the machine is a client or a server machine.

For stand-alone machines, you can create the new user and group IDs locally. For machines administered in a network information services (NIS) domain, you can create the user and group IDs on the NIS master server machine.

Installation

Carry out the following procedure:

1. Check whether Volume Manager is running on your system by typing the following command:

```
/usr/bin/ps -ef | /bin/grep vold
```

Installing on Sun Solaris

If it is running, the CD is mounted on /cdrom/mqclient automatically. If it is not running, mount the CD by typing the following commands:

```
mkdir -p /cdrom/mqclient
mount -F hsfs -r /dev/dsk/cntndnsn /cdrom/mqclient
```

substituting cntndnsn with the name of your CD-ROM device.

2. Use the Sun Solaris **pkgadd** program, to install the MQSeries client software by carrying out the following procedure:
 - a. Type `pkgadd -d /cdrom/mqclient/solaris/mqs510.img`.
 - b. You are prompted for a list of components to be installed. Select the ones you require - if you want to install all the components, select **all**.

See “Components for AIX” on page 33 for details.

The component **MQSeries Client** for Java should be installed only if you have Java 1.1.1 (or later compatible) runtime code on your machine. You also require version 2.6 or later of the Sun Solaris operating system.
 - c. Press the Enter key.

For further information on using **pkgadd** to install software packages, see the Sun Solaris documentation.

Kernel Configuration

See the MQSeries family Web site for a SupportPac that gives additional performance information - see “Available from the Internet” on page 100.

Translated Messages

Messages in U.S. English are always available. If you require another of the languages that is supported by MQSeries for Sun Solaris, you *must* ensure that your NLSPATH environment variable includes the appropriate directory.

For example:

```
export LANG=de
export NLSPATH=/usr/lib/locale/%L/LC_MESSAGES/%N
```

Removing an MQSeries Client from Sun Solaris

If you have previously installed MQSeries on your system, you need to remove the product using the **pkgrm** program.

If the product is present, but not installed correctly, you may need manually to delete the files and directories contained in /var/mqm and /opt/mqm.

Installing on Windows 95 or Windows 98

To install an MQSeries client on a Windows 95 or a Windows 98 system you use the MQSeries Client CD-ROM supplied as part of the MQSeries product.

Note: If you plan to install an MQSeries client and server on the same machine, see “Installing the Server and Client on the Same Machine” on page 21.

Installation

1. Insert the MQSeries Client CD-ROM into the CD-ROM drive.
2. The installation automatically starts and an MQSeries Language Selection window is displayed.

Note: If you have disabled auto-playing of CD-ROMs, run SETUP instead, from the root directory.

This window presents you with a list of the national languages that are available.
3. On the MQSeries Language Selection window select the language of your choice, and click on the **Next** button or press Enter.

The MQSeries Welcome window is displayed.
4. **Make sure you are installing the correct client** for your system, as displayed in the Welcome window.
5. Select Destination Directory requires a destination directory into which the MQSeries files will be installed.

You can change the default shown by selecting the browse button and choosing a different drive and directory, then click on **OK**. Click on the **Next** button or press Enter to continue.
6. Choose MQSeries Components displays a list of components from which you can select the ones you want to be installed. See “Components for Windows 95 and Windows 98” on page 43 for details.

To select a component, click in the box next to it so that a check mark appears (just highlighting the line does not select it). The space needed for each component is shown here, and the space available on the drive you have selected.

Click on the **Next** button or press Enter to continue.
7. Select Program Folder prompts you for a folder name to contain the MQSeries objects. The default name is MQSeries Client for MQSeries client for Windows 95 and Windows 98. You can rename the default or choose an existing folder.

Installing on Windows 95 or 98

8. Start Copying Files displays all the selections you have made. Click on the **Back** button if you want to return to a previous window to change any of your choices.

When you have checked your choices, Click on the **Next** button or press Enter to start the file copying process.

The progress indicator shows which components are being copied and the percentage of copying completed.

9. The next window presents you with the opportunity to view the README file. If you do not want to view the README file at this point, it will be available in the MQSeries client folder.

If you view the README file, close the window of the README to continue the installation process.

10. The installation of the MQSeries client is now complete, and a window is displayed giving you the option of restarting your computer now or leaving it until later. It is recommended that you restart your computer now. Close all the other applications that are running before continuing with this step.

Once this has been done, select Yes, I want to restart my computer now and click on the **Finish** button to complete the setup.

11. When setup is complete, the MQSeries Client folder is added to the desktop, or the location you specified. Note that the MQSeries client is a set of services and it does not have to be explicitly run, so the folder does not have an object called a "client".

Unattended Installation on Windows 95 and Windows 98

You can install a Windows client on a remote machine without interaction. This process is known as *unattended installation*, sometimes referred to as *silent installation*, and uses response files.

An installation response file is an ASCII text file containing values for the options that you select when you install an MQSeries for Windows system.

For unattended installation, you must:

1. Carry out an installation on a machine and record the actions performed to install the product.

In this example, the machine is being used as a server – see "Setting up the Server" on page 69.

Installing on Windows 95 or 98

2. Use the recording you have created to install the product on a remote machine – see “Installing on the Remote Machine” on page 70.

Setting up the Server

To set up your server, follow these steps:

1. Load the MQSeries client CD-ROM. When the *Language Selection* window appears, select **Cancel**.
2. Copy the *win95* or *win98* subdirectory, and all of its subdirectories, from the client CD-ROM to a shared-resource image directory.
3. Select one of the following language directories:
 - SetupCn – Simplified Chinese
 - SetupDe – German
 - SetupEn – English
 - SetupEs – Spanish
 - SetupIt – Italian
 - SetupJp – Japanese
 - SetupKo – Korean
 - SetupPt – Brazilian Portuguese
 - SetupTw – Traditional Chinese
4. From the selected language directory, run `setup.exe`, with the `-r` option to create a response file.

This can be done by:

- a. Highlighting `setup.exe`
- b. Dragging it to the **Start, Run** dialog
- c. Adding `-r` to the end of the fully-qualified file name
- d. Selecting **OK** to start the process
- e. Selecting the components that you want to install

Notes:

- a. When the installation has ended, a response file will have been automatically generated that can be used as the basis for unattended installation.
 - b. The response file `SETUP.ISS` will be generated in the `WINDOWS` directory for Windows 95 or Windows 98.
5. When the response file has been generated, you can copy it to the language dependent directory in the shared-resource image directory for your national language.

For example, to copy the `SETUP.ISS` file from the `WINNT35` directory to `\IMAGE\SETUPEN`, type:

```
COPY C:\WINNT35\SETUP.ISS M:\INSTMQS\IMAGE\SETUPEN
```

where `SETUPEN` is the appropriate language directory for your system.

Installing on Windows 95 or 98

This assumes that the Windows 95 or 98 operating system was installed to the C drive and that the MQSeries installation image on the server resides on M:\INSTMQS.

Now you have set up your server, you can go on to install on the remote machine.

Installing on the Remote Machine

Unattended installation is particularly useful for installing a client over a network because you can do it from a redirected drive on a LAN server.

The remote machine *must* have access to the shared-resource image directory, which includes the:

- Appropriate language subdirectory
- Disk subdirectories

To connect the workstation to the redirected drive on the server, follow these steps:

1. Use the NET USE command as follows:

```
NET USE devicename \\servername\netname
```

For example:

```
NET USE J: \\MQMNT\MQMSHARE
```

where:

- J: is the logical drive name to be connected to the shared resource directory.
 - MQMSHARE is the netname for M:\INSTMQS.
2. Go to the language directory that contains the SETUP.EXE and SETUP.ISS files and run **SETUP -s**.

Notes:

1. The -s parameter indicates that the installation is to be carried out in silent mode. This parameter is mandatory for an unattended installation.
2. SETUP.EXE uses the SETUP.ISS file in the current directory to silently install the product.

Removing an MQSeries Client from Windows 95 and Windows 98

If you want to remove the MQSeries client files from your machine, use Settings/ Control Panel/ Add-Remove programs. First select MQSeries Client which launches

Installing on Windows 3.1

the uninstall program. Then select the components you want to remove and click **Remove**.

If you choose to remove all MQSeries components and are then likely to reinstall MQSeries, you should restart your computer to complete the uninstall process. You cannot reinstall any components until you have restarted.

All the MQSeries client files that were created at installation time are removed by the process.

Installing on Windows 3.1

To install an MQSeries client on a Windows 3.1 system you use the MQSeries Client CD-ROM supplied as part of the MQSeries product.

Installation

1. Insert the MQSeries Client CD-ROM into the CD-ROM drive.
2. Open the directory WIN31.
3. Change to the appropriate language subdirectory for the language you wish to install:
 - setupen - English
 - setuppt - Brazilian Portuguese
 - setupfr - French
 - setupde - German
 - setupit - Italian
 - setupjp - Japanese
 - setupko - Korean
 - setupes - Spanish
 - setupcn - Simplified Chinese
 - setuptw - Traditional Chinese
4. Run setup.exe
The MQSeries Welcome window is displayed.
5. **Make sure you are installing the correct client** for your system, Windows 3.1, as displayed in the Welcome window.
6. Select Destination Directory requires a destination directory into which the MQSeries files will be installed.
You can change the default directory by selecting the browse button and choosing a different drive and directory, then click on **OK**. Click on the **Next** button or press Enter to continue.

Installing on Windows 3.1

7. Choose MQSeries Components displays a list of components from which you can select the ones you want to be installed. See “Components for DOS” on page 35 for details.

To select a component, click in the box next to it so that a check mark appears (just highlighting the line does not select it). The space needed for each component is shown here, and the space available on the drive you have selected. Click on the **Next** button or press Enter to continue.
8. Select Program Folder prompts you for a folder name to contain the MQSeries objects. The default name is MQSeries Client for Windows 3.1. You can rename the default or choose an existing folder.
9. Start Copying Files displays the selection you have made. Click on the **Back** button if you want to return to change your choice.

Now click on the **Next** button or press Enter to start the file copying process.

The progress indicator shows what components are being copied and the percentage of copying completed.
10. The next window presents you with the opportunity to view the README file. If you do not want to view the README file at this point, it will be available in the MQSeries client folder.

If you view the README file, close the window of the README to continue the installation process.
11. The installation of the MQSeries client is now complete, and a window is displayed giving you the option of restarting your computer now or leaving it until later. It is recommended that you restart your computer now. Close all the other applications that are running before continuing with this step.

Select Yes, I want to restart my computer now and click on the **Finish** button to complete the setup.
12. When setup is complete, the MQSeries Client folder is added to the Program Manager. Note that the MQSeries client is a set of services and it does not have to be explicitly run, so the folder does not have an object called a “client”.

Removing an MQSeries Client from Windows 3.1

If you want to remove the MQSeries client files from your machine, use the process provided to do this efficiently.

Run Uninstall from the MQSeries client folder. You are prompted before continuing.

All the MQSeries client files that were created at installation time are removed by the process.

Installing on Windows NT

To install an MQSeries client on Windows NT, use the MQSeries Client CD-ROM supplied as part of the MQSeries product.

Note: If you plan to install an MQSeries client and server on the same machine, see “Installing the Server and Client on the Same Machine” on page 21.

Installation

To install an MQSeries client for Windows NT, you must be logged on to Windows NT as an administrator.

MQSeries checks for any existing MQSeries configuration files (MQS.INI). If it finds any, it automatically migrates configuration information to the Windows NT registry. Otherwise, MQSeries automatically puts its configuration information directly into the Windows NT registry.

Installation Procedure

1. Insert the MQSeries Client CD-ROM into the CD-ROM drive.
2. The installation automatically starts and an MQSeries Language Selection window is displayed.

Note: If you have disabled auto-playing of CD-ROMs, run SETUP instead, from the root directory.

This window presents you with a list of the national languages that are available.

3. On the MQSeries Language Selection window select the language of your choice, and click on the **OK** button or press Enter.

The MQSeries Welcome window is displayed.

4. **Make sure you are installing the correct client** for your system, as displayed in the Welcome window.

5. Choose Installation Folders lets you choose folders into which the MQSeries program files and data files will be installed.

You can change the default shown by selecting the browse button and choosing a different drive and directory, then click on **OK**. Click on the **Next** button or press Enter to continue.

6. Select Components displays a list of components from which you can select the ones you want to be installed.

Installing on Windows NT

The component **Java Client** should be installed only if you have Java 1.1.1 (or later compatible) runtime code on your machine.

To select a component, click in the box next to it so that a check mark appears (just highlighting the line does not select it). The space needed for each component is shown on the panel, and the space available on the drive you have selected.

If you deselect a previously installed component it is removed. Click on the **Next** button or press Enter to continue.

7. Select Program Folder prompts you for a folder name to contain the MQSeries objects. The default name is IBM MQSeries Client. You can rename the default or choose an existing folder.

8. Ready to Copy Files displays all the selections you have made. Click on the **Back** button if you want to return to a previous window to change any of your choices.

When you have checked your choices, click on the **Next** button or press Enter to start the file copying process.

The progress indicator shows which components are being copied and the percentage of copying completed.

9. The Setup Complete window appears when the selected components have been installed. Click the Finish button or press Enter to close the window after optionally selecting Launch Notepad to View the Release Notes.

10. The installation of the MQSeries client is now complete.

11. When setup is complete, the IBM MQSeries folder is added to the Start menu. Note that MQSeries Clients are sets of services and do not have to be explicitly run, so the folder does not have an object called "client".

Installing from a LAN

There are two ways to put MQSeries installation files on a LAN file server for easier access: you can make the MQSeries for Windows NT Client CD-ROM drive shareable, or you can copy the installation files from the CD-ROM to a file server, by following these steps:

1. Create a folder on the LAN file server to store the installation files. For example:

```
md m:\instmq
```

2. Load the MQSeries for Windows NT client CD-ROM. If you have autorun enabled, the language selection panel will appear – cancel this panel.

3. Copy the Winnt directory structure from the CD-ROM to the m:\instmq folder. For example:

```
xcopy e:\Winnt\*. * m:\instmq /e
```

Note: You can save space on the hard drive by copying only the subfolders for the languages that you require. The language subfolders are:

Installing on Windows NT

SetupEn	English
SetupFr	French
SetupDe	German
SetupEs	Spanish
SetupIt	Italian
SetupJp	Japanese
SetupKo	Korean
SetupPt	Brazilian Portuguese
SetupTw	Traditional Chinese
SetupCn	Simplified Chinese

4. Give all licensed users access to the folder that now contains the CD-ROM image (in this example, the m: drive).
5. From a command prompt on the target machine, connect to the appropriate drive and folder using the net use command:

```
net use devicename \\servername\netname
```

For example:

```
net use x: \\mqmnt\instmq
```

where x: is the required mapped drive on the target machine. Alternatively, use Windows NT Explorer or another method to map the shared resource to a drive letter.

6. Change to the installation directory (in this example x), type Setup and press Enter.
7. Follow the prompts.

Using the System Management Server with MQSeries for Windows NT

This section describes how to install, or remove, an MQSeries for Windows NT client using the System Management Server (SMS).

Creating SMS Packages and Jobs for MQSeries

You must create:

- An SMS software package containing the MQSeries software.
- An SMS job to distribute and install the software package (see “Creating the MQSeries SMS Job” on page 77).

For more detailed information on how to create a software package and a job, refer to the Microsoft System Management Server documentation.

Installing on Windows NT

Creating the MQSeries SMS Software Package

To create the SMS software installation package:

1. From the Microsoft SMS Administrator application, open the **Packages** folder and then create a new package.
2. In the SMS **Package Properties** dialog click on the **Import** button to create the software package by importing a Package Definition File (PDF).
3. In the **File Browser** dialog, select the drive where the IBM MQSeries client CD-ROM is located.
4. Select the Winnt directory which contains the package definition file MQSERIES.PDF.
You can also find the MQSERIES.PDF file in the local drive, or shared network drive where you copied the MQSeries Installation software.
5. Select the **MQSERIES.PDF** file and click on the **OK** button.
6. Click on the **Workstation** button. In the **Source Directory** entry field, specify the fully-qualified path name to the MQSeries root directory which contains the MQSeries installation software.
7. Select the appropriate Workstation Command Line:
 - **Automated Uninstallation of IBM MQSeries - Windows NT client**
 - **Automated Installation of IBM MQSeries - Windows NT client (US English)**
8. Click on the **Properties** button for each process and review the **Command Line** entry field to ensure that the parameters are correct.
Note: The `-i<miffilepath>` parameter specifies the full path and file name of a Management Information Format (.mif) file which installation and uninstallation can generate. Remove the parameter if you do not want to generate a .mif file.
9. Click on the **Close** button to close the **Workstation Properties** dialog.
Note: If you specified a local path in the **Source Directory** entry field, you will get a pop-up dialog warning you that the local path you specified may not be accessible to SMS components running on another machine. Click on the **OK** button to continue.
10. Click on the **OK** button to close the **Package Properties** window.
A pop-up dialog appears indicating that SMS will update the software package at all sites. Click the **OK** button to continue.

The software package has been created and can be installed by creating an SMS job.

Installing on Windows NT

Creating the MQSeries SMS Job

You must now create an SMS job to distribute and install the software packages you created, which contain the MQSeries installation software.

Refer to the Microsoft System Management Server documentation for detailed information on how to create and run a job.

Notes:

1. You **must** be logged onto the target machine with Administrator authority in order to install the MQSeries Server.
2. When creating an SMS Job to distribute and install the software package, ensure that you select the appropriate workstation command. The workstation commands are displayed on the **Job Details** dialog in the **Run Phase** section and appear in a drop down list box.

Unattended Installation on Windows NT

You can install MQSeries for Windows NT on a remote machine without interaction, provided that the remote machine can share the client CD-ROM, or a copy of the files on it, and that you can execute a command on the remote machine. This process is called unattended (or silent) installation, and is particularly useful for installing MQSeries for Windows NT clients over a network because you can do it from a shared drive on a LAN file server.

Because there is no user interaction, unattended installation uses a response file. A response file is an ASCII text file containing values for the installation options you want to select. For information on the format of response files, see Appendix A, "Installation response file format for Windows NT" in the *MQSeries Clients* book.

To generate a response file you can:

- Carry out an installation on a machine and record the options selected to install the product in a response file. To do this you must run setup.exe with the -r parameter, and optionally, the -noinst parameter:
 1. Load the MQSeries for Windows NT client CD-ROM. If you have autorun enabled, the language selection panel will appear – cancel this panel.
 2. Run setup.exe from the root folder of the CD-ROM, with the -r parameter. This can be done by dragging setup.exe from Windows NT Explorer to the Run dialog in the start menu and adding -r to the end of the file name, or by typing the fully-qualified path to setup.exe into the Run dialog, followed by -r.
 3. The language selection dialog reappears. Perform an installation as you want it performed on the remote machine.

Installing on Windows NT

4. If you are asked to restart the system, you do not need to do so if you are only generating a response file.

The `-noinst` parameter (which, if present, must precede the `-r` parameter) suppresses the MQSeries installation; Setup generates only a response file. The installation creates a response file called `setup.iss` in the Windows directory, normally `c:\winnt`. Save this file; you can edit it if necessary.

Or

- Edit the response file (`setup.iss`) supplied in the Winnt directory on the MQSeries for Windows NT client CD-ROM, using an ASCII file editor.

Or

- Generate your own response file using an ASCII file editor.

Use the response file you have created to install the product on a remote machine (see "Installing on the Remote Machine").

Installing on the Remote Machine

The remote machine must have access to a shared resource or drive on a file server that contains the client CD-ROM or a copy of it. Perform the procedure described in "Installing from a LAN" on page 74 up to the point where you run Setup. Copy your response file to a location on the file server that is accessible from the remote machine. You can now start the installation on the remote machine:

1. On the remote machine, go to the Setupxx folder on the shared resource.
2. Substitute Setupxx with the name of the language subfolder for the language that you require.
3. Run `setup.exe`:

```
setup -g<logfile> -f1<responsefile> -f2<secondarylogfile> -s
```

This installs in the language specified by Setupxx.

In the above command:

<responsefile>

The full path to the response file you prepared. If you omit the `-f1<responsefile>` parameter, the response file must be in the Setupxx language subdirectory.

<logfile>

The full path to an installation log file. Setup creates an English ASCII text log file containing details of what happens during installation. You should check this file to see if any errors occurred. If you omit the `-g<logfile>` parameter, Setup creates a file called `amqilogn.txt` in the data-files folder on the machine running the installation. In this case, any messages generated before the data-files folder is created are lost. The `-g<logfile>` parameter must be placed before the `-f1` and the `-s` parameters, otherwise it is ignored.

Configuring a Client

<secondarylogfile>

The full path to a secondary installation log file. This file contains less detail than the other log file and should not be used as the primary source for information about the success of the installation. If you omit the -f2<secondarylogfile> parameter, Setup attempts to create a file called setup.log in the language subdirectory (Setupxx), which is not possible if the installation is being performed from a CD-ROM. The -f2<secondarylogfile> parameter must be placed after the -f1 parameter.

-s

This parameter tells Setup to run in silent mode.

Enclose the long path name and file name expressions in double quotes.

Removing an MQSeries Client from Windows NT

If you want to remove the MQSeries client files from your machine, use Settings/ Control Panel/ Add-Remove. First select IBM MQSeries Client, which launches the install program. Alternatively, choose MQSeries Uninstallation from the IBM MQSeries client folder. You can choose to uninstall one or more components or the whole of the MQSeries client, including or excluding data.

For more information on uninstalling an MQSeries for Windows NT client, see Appendix B, "Uninstallation of Windows NT" in the *MQSeries Clients* book.

Configuring and Verifying a Client

After you have installed the client, you need to configure your communications and verify the installation. For information on how to do this, see Chapter 6, "Verifying the installation" in the *MQSeries Clients* book.

Configuring a Client

Chapter 6. Using the MQSeries Command Sets

This chapter introduces the command sets that can be used to perform system administration tasks on MQSeries objects.

Administration tasks include creating, starting, altering, viewing, stopping, and deleting MQSeries objects such as queue managers, queues, processes, channels, and namelists. To perform these tasks, you must select the appropriate command from one of the supplied command sets (see “Introducing Command Sets”).

Introducing Command Sets

MQSeries provides three command sets for performing administration tasks:

- Control commands
- MQSC commands
- PCF commands

This section describes the command sets that are available. Some tasks can be performed using either a control command or an MQSC command, whilst other tasks can be performed using only one type of command. For a comparison of the facilities provided by the different types of command set, see Appendix G, “Comparing command sets” in the *MQSeries System Administration* book.

Control Commands

Control commands fall into three categories:

- *Queue manager commands*, including commands for creating, starting, stopping, and deleting queue managers and command servers.
- *Channel commands*, including commands for starting and ending channels and channel initiators.
- *Utility commands*, including commands associated with authority management and conversion exits.

MQSeries Command Sets

Using Control Commands

In MQSeries in UNIX® environments, you enter control commands in a shell window. In these environments, control commands, including the command name itself, the flags, and any arguments, are case sensitive. For example, in the command:

```
crtmqm -u SYSTEM.DEAD.LETTER.QUEUE jupiter.queue.manager
```

- The command name must be **crtmqm**, not **CRTMQM**.
- The flag must be **-u**, not **-U**.
- The dead-letter queue is **SYSTEM.DEAD.LETTER.QUEUE**.
- The argument is specified as **jupiter.queue.manager**, which is different from **JUPITER.queue.manager**.

Therefore, take care to type the commands exactly as you see them in the examples.

The following list contains a brief description of each of the control commands. You can obtain help for the syntax of any of the commands by entering the command followed by a question mark. MQSeries responds by listing the syntax required for the selected command.

crtmqcvx (data conversion)

Creates a fragment of code that performs data conversion on data type structures.

crtmqm (create queue manager)

Creates a local queue manager and defines the default and system objects.

dltmqm (delete queue manager)

Deletes a specified queue manager.

dmpmqlog (dump log)

Dumps a formatted version of the MQSeries system log.

dspmqaout (display authority)

Displays the current authorizations to a specified object.

dspmqcsv (display command server)

Displays the status of the command server for the specified queue manager.

dspmqls (display MQSeries files)

Displays the real file system name for all MQSeries objects that match a specified criterion.

dspmqrtrc (display MQSeries formatted trace output)

Displays MQSeries formatted trace output.

dspmqrtrn (display MQSeries transactions)

Displays details of in-doubt transactions.

endmqcsv (end command server)

Stops the command server on the specified queue manager.

endmqlsr

Ends a listener process.

endmqm (end queue manager)

Stops a specified local queue manager.

MQSeries Command Sets

endmqtrc (end MQSeries trace)

Ends tracing for the specified entity or all entities.

rcdmqimg (record media image)

Writes an image of an MQSeries object, or group of objects, to the log for use in media recovery.

rcrmqobj (recreate object)

Recreates an object, or group of objects, from their images contained in the log.

rsvmqtrn (resolve MQSeries transactions)

Commits or backs-out internally or externally coordinated in-doubt transactions.

runmqchi (run channel initiator)

Runs a channel initiator process.

runmqchl (run channel)

Runs either a Sender (SDR) or a Requester (RQSTR) channel.

runmqdlq (run dead-letter queue handler)

Starts the dead-letter queue (DLQ) handler, a utility that you can run to monitor and handle messages on a dead-letter queue.

runmqslr (run listener)

Runs a listener process.

runmqsc (run MQSeries commands)

Issues MQSC commands to a queue manager.

runmqtrm (start trigger monitor)

Invokes a trigger monitor.

setmqaut (set/reset authority)

Changes the authorizations to an object or to a class of objects.

strmqcsv (start command server)

Starts the command server for the specified queue manager.

strmqm (start queue manager)

Starts a local queue manager.

strmqtrc (start MQSeries trace)

Enables tracing.

For more information about the syntax and purpose of control commands, see Chapter 17, "MQSeries control commands" in the *MQSeries System Administration* book.

MQSeries (MQSC) Commands

You use the MQSeries (MQSC) commands to manage queue manager objects, including the queue manager itself, channels, queues, and process definitions. For example, there are commands to define, alter, display, and delete a specified queue.

When you display a queue, using the **DISPLAY QUEUE** command, you display the queue *attributes*. For example, the **MAXMSGL** attribute specifies the maximum length of a message that can be put on the queue. The command does not show you the messages on the queue.

Working With Queue Managers

For detailed information about each MQSC command, see Chapter 2, “The MQSeries commands” in the *MQSeries Command Reference* book.

Running MQSC Commands

You run MQSC commands by invoking the control command **runmqsc**. You can run MQSC commands:

- Interactively by typing them at the keyboard
- As a sequence of commands from a text file

For more information about using MQSC commands, see Chapter 7, “Administering local MQSeries objects” in the *MQSeries System Administration* book.

PCF Commands

MQSeries programmable command format (PCF) commands allow administration tasks to be programmed into an administration program. In this way you can create queues and process definitions, and change queue managers, from a program. PCF commands cover the same range of functions that are provided by the MQSC facility. You can therefore write a program to issue PCF commands to any queue manager in the network from a single node. In this way, you can both centralize and automate administration tasks.

Note: Unlike MQSC commands, PCF commands and their replies are not in a text format that you can read.

For a complete description of the PCF data structures and how to implement them, see Chapter 6, “Introduction to Programmable Command Formats” in the *MQSeries Programmable System Management* book.

Working with Queue Managers

This section describes how you can perform operations on queue managers, such as creating, starting, stopping, and deleting them. MQSeries provides control commands for performing these tasks.

Before you can do anything with messages and queues, you must create at least one queue manager.

Creating a Default Queue Manager

The following command:

- Creates a default queue manager called `saturn.queue.manager`.
- Creates the default and system objects automatically.
- Specifies the names of both a default transmission queue and a dead-letter queue.

```
crtmqm -q -d MY.DEFAULT.XMIT.QUEUE -u  
SYSTEM.DEAD.LETTER.QUEUE saturn.queue.manager
```

where:

-q Indicates that this queue manager is the default queue manager.

-d MY.DEFAULT.XMIT.QUEUE
Is the name of the default transmission queue.

-u SYSTEM.DEAD.LETTER.QUEUE
Is the name of the dead-letter queue.

saturn.queue.manager
Is the name of this queue manager. This must be the last parameter specified on the **crtmqm** command.

For more information about these attributes, see “crtmqm (Create queue manager)” in the *MQSeries System Administration* book.

Starting a Queue Manager

Although you have created a queue manager, it cannot process commands or MQI calls until it has been started. Start the queue manager by typing in this command:

```
strmqm saturn.queue.manager
```

The **strmqm** command does not return control until the queue manager has started and is ready to accept connect requests.

Stopping a Queue Manager

To stop a queue manager, use the **endmqm** command. For example, to stop a queue manager called `saturn.queue.manager` use this command:

```
endmqm saturn.queue.manager
```

Working With Queue Managers

Quiesced Shutdown

By default, the above command performs a *quiesced shutdown* of the specified queue manager. This may take a while to complete—a quiesced shutdown waits until all connected applications have disconnected.

Use this type of shutdown to notify applications to stop; you are not told when they have stopped.

You can specify the `-w` flag if you require confirmation that the queue manager has stopped. For example:

```
endmqm -w saturn.queue.manager
```

The command prompt does not return until the queue manager has stopped.

Immediate Shutdown

An *immediate shutdown* allows any current MQI calls to complete, but any new calls fail. This type of shutdown does not wait for applications to disconnect from the queue manager. Use this as the normal way to stop the queue manager, optionally after a quiesce period.

For an immediate shutdown, the command is:

```
endmqm -i saturn.queue.manager
```

Preemptive Shutdown

Do not use this method unless all other attempts to stop the queue manager using the **endmqm** command have failed. This method can have unpredictable consequences for connected applications.

If an immediate shutdown does not work, you must resort to a *preemptive shutdown*, specifying the `-p` flag. For example:

```
endmqm -p saturn.queue.manager
```

This stops all queue manager code immediately.

Restarting a Queue Manager

To restart a queue manager called `saturn.queue.manager`, use the command:

```
strmqm saturn.queue.manager
```

Deleting a Queue Manager

To delete a queue manager called `saturn.queue.manager`, first stop it, then use the following command:

```
dltmqm saturn.queue.manager
```

Note: Deleting a queue manager is a serious step, because you also delete all resources associated with that queue manager, including all queues and their messages, and all object definitions.

Working with MQSeries Objects

This section describes briefly how to use MQSC commands to create, display, change, copy, and delete MQSeries objects.

You can use the MQSC facility interactively (by entering commands at the keyboard) or you can redirect the standard input device (stdin) to run a sequence of commands from a text file. The format of the commands is the same in both cases. The examples included here assume that you will be using the interactive method.

For more information about using MQSC commands, see “Performing local administration tasks using MQSC commands” in the *MQSeries System Administration* book. For a complete description of the MQSC commands, see Chapter 2, “The MQSeries commands” in the *MQSeries Command Reference* book.

Before you can run MQSC commands, you must have created and started the queue manager that is going to run the commands. For more information see “Creating a Default Queue Manager” on page 85.

Using the MQSC Facility Interactively

To start using the MQSC facility interactively, use the `runmqsc` command. Open a shell and enter:

```
runmqsc
```

A queue manager name has not been specified, therefore the MQSC commands will be processed by the default queue manager. Now type in any MQSC commands, as required. For example:

```
DEFINE QLOCAL (ORANGE.LOCAL.QUEUE)
```

Working with Objects

Continuation characters must be used to indicate that a command is continued on the following line:

- A minus sign (-) indicates that the command is to be continued from the start of the following line.
- A plus sign (+) indicates that the command is to be continued from the first nonblank character on the following line.

Command input terminates with the final character of a nonblank line that is not a continuation character. You can also terminate command input explicitly by entering a semicolon (;). (This is especially useful if you accidentally enter a continuation character at the end of the final line of command input.)

Feedback from MQSC Commands

When you issue commands from the MQSC facility, the queue manager returns operator messages that confirm your actions or tell you about the errors you have made. For example:

```
AMQ8006: MQSeries queue created
.
.
.
AMQ8405: Syntax error detected at or near end of command segment below:-
Z
```

The first message confirms that a queue has been created; the second indicates that you have made a syntax error.

These messages are sent to the standard output device. If you have not entered the command correctly, refer to Chapter 1, "Using MQSeries Commands" in the *MQSeries Command Reference* book for the correct syntax.

Ending Interactive Input to MQSC

To end interactive input of MQSC commands, enter the MQSC END command:

```
END
```

Alternatively, you can use the EOF character CTRL+D

If you are redirecting input from other sources, such as a text file, you do not have to do this.

Defining a Local Queue

For an application, the local queue manager is the queue manager to which the application is connected. Queues that are managed by the local queue manager are said to be local to that queue manager.

Use the MQSC command **DEFINE QLOCAL** to create a definition of a local queue and also to create the data structure that is called a queue. You can also modify the queue characteristics from those of the default local queue.

In this example, the queue we define, `ORANGE.LOCAL.QUEUE`, is specified to have these characteristics:

- It is enabled for gets, disabled for puts, and operates on a first-in-first-out (FIFO) basis.
- It is an 'ordinary' queue, that is, it is not an initiation queue or a transmission queue, and it does not generate trigger messages.
- The maximum queue depth is 1000 messages; the maximum message length is 2000 bytes.

The following MQSC command does this:

```
DEFINE QLOCAL (ORANGE.LOCAL.QUEUE) +  
  DESCR('Queue for messages from other systems') +  
  PUT (DISABLED) +  
  GET (ENABLED) +  
  NOTRIGGER +  
  MSGDLVSQ (FIFO) +  
  MAXDEPTH (1000) +  
  MAXMSGL (2000) +  
  USAGE (NORMAL);
```

Notes:

1. Most of these attributes are the defaults as supplied with the product. However, they are shown here for purposes of illustration. You can omit them if you are sure that the defaults are what you want or have not been changed. See also "Displaying Default Object Attributes" on page 90.
2. `USAGE (NORMAL)` indicates that this queue is not an initiation queue or a transmission queue.
3. If you already have a local queue on the same queue manager with the name `ORANGE.LOCAL.QUEUE`, this command fails. Use the `REPLACE` attribute if you want to overwrite the existing definition of a queue, but see also "Changing Local Queue Attributes" on page 91.

Working with Objects

Displaying Default Object Attributes

When you define an MQSeries object, it takes any attributes that you do not specify from the default object. For example, when you define a local queue, the queue inherits any attributes that you omit in the definition from the default local queue, which is called `SYSTEM.DEFAULT.LOCAL.QUEUE`. The default local queue is created automatically when you create the default queue manager. To see exactly what these attributes are, use the following command:

```
DISPLAY QUEUE (SYSTEM.DEFAULT.LOCAL.QUEUE)
```

Note: The syntax of this command is different from that of the corresponding **DEFINE** command.

You can selectively display attributes by specifying them individually. For example:

```
DISPLAY QUEUE (ORANGE.LOCAL.QUEUE) +  
    MAXDEPTH +  
    MAXMSGL +  
    CURDEPTH;
```

This command displays the three specified attributes as follows:

```
AMQ8409: Display Queue details.  
    QUEUE(ORANGE.LOCAL.QUEUE)  
    MAXDEPTH(1000)  
    MAXMSGL(2000)  
    CURDEPTH(0)
```

`CURDEPTH` is the current queue depth, that is, the number of messages on the queue. This is a useful attribute to display, because by monitoring the queue depth you can ensure that the queue does not become full.

Copying a Local Queue Definition

You can copy a queue definition using the `LIKE` attribute on the **DEFINE** command.

For example:

```
DEFINE QLOCAL (MAGENTA.QUEUE) +  
    LIKE (ORANGE.LOCAL.QUEUE)
```

This command creates a queue with the same attributes as our original queue `ORANGE.LOCAL.QUEUE`, rather than those of the system default local queue.

You can also use this form of the **DEFINE** command to copy a queue definition, but substituting one or more changes to the attributes of the original. For example:

```
DEFINE QLOCAL (THIRD.QUEUE) +  
    LIKE (ORANGE.LOCAL.QUEUE) +  
    MAXMSGL(1024);
```

Working with Objects

This command copies the attributes of the queue `ORANGE.LOCAL.QUEUE` to the queue `THIRD.QUEUE`, but specifies that the maximum message length on the new queue is to be 1024 bytes, rather than 2000.

Notes:

1. When you use the **LIKE** attribute on a **DEFINE** command, you are copying the queue attributes only. You are not copying the messages on the queue.
2. If you define a local queue, without specifying **LIKE**, it is the same as `DEFINE LIKE(SYSTEM.DEFAULT.LOCAL.QUEUE)`.

Changing Local Queue Attributes

You can change queue attributes in two ways, using either the **ALTER QLOCAL** command or the **DEFINE QLOCAL** command with the **REPLACE** attribute. In “Defining a Local Queue” on page 89, we defined the queue `ORANGE.LOCAL.QUEUE`. Suppose, for example, you wanted to increase the maximum message length on this queue to 10 000 bytes.

- Using the **ALTER** command:

```
ALTER QLOCAL (ORANGE.LOCAL.QUEUE) MAXMSGL(10000)
```

This command changes a single attribute, that of the maximum message length; all the other attributes remain the same.

- Using the **DEFINE** command with the **REPLACE** option, for example:

```
DEFINE QLOCAL (ORANGE.LOCAL.QUEUE) MAXMSGL(10000) REPLACE
```

This command changes not only the maximum message length, but all the other attributes, which are given their default values. The queue is now put enabled whereas previously it was put inhibited. Put enabled is the default, as specified by the queue `SYSTEM.DEFAULT.LOCAL.QUEUE`, unless you have changed it.

If you decrease the maximum message length on an existing queue, existing messages are not affected. Any new messages, however, must meet the new criteria.

Working with Objects

Clearing a Local Queue

To delete all the messages from a local queue called `MAGENTA.QUEUE`, use the following command:

```
CLEAR QLOCAL (MAGENTA.QUEUE)
```

You cannot clear a queue if:

- There are uncommitted messages that have been put on the queue under syncpoint.
- An application currently has the queue open.

Deleting a Local Queue

Use the MQSC command **DELETE QLOCAL** to delete a local queue. A queue cannot be deleted if it has uncommitted messages on it. However, if the queue has one or more committed messages, and no uncommitted messages, it can be deleted only if you specify the `PURGE` option. For example:

```
DELETE QLOCAL (PINK.QUEUE) PURGE
```

Specifying `NOPURGE` instead of `PURGE` ensures that the queue is not deleted if it contains any committed messages.

Browsing Queues

MQSeries provides a sample queue browser that you can use to look at the contents of the messages on a queue. The browser is supplied in both source and executable formats.

The default file names and paths are:

Source	<code>/opt/mqm/samp/amqsbcg0.c</code>
Executable	<code>/opt/mqm/samp/bin/amqsbcg</code>

The sample requires two input parameters, the queue manager name and the queue name. For example:

```
amqsbcg ORANGE.LOCAL.QUEUE saturn.queue.manager
```

There are no defaults; both parameters are required.

Chapter 7. Using the MQSeries World Wide Web Interface

This chapter introduces MQSeries Internet Gateway. It also explains how to get more information about obtaining and using the product.

Overview of MQSeries Internet Gateway

MQSeries Internet Gateway provides a bridge between the synchronous World Wide Web and asynchronous MQSeries applications. With the gateway, Web server software and MQSeries together provide an Internet-connected Web browser with access to MQSeries applications. This means that enterprises can take advantage of the low-cost access to global markets provided by the Internet, while benefitting from the robust infrastructure and assured message delivery of MQSeries.

User interaction with the gateway is through HTML fill-out form POST requests; MQSeries applications respond by returning HTML pages to the gateway, via an MQSeries queue.

MQSeries Internet Gateway can be installed on AIX, OS/2, OS/390, Sun Solaris, HP-UX, or Windows NT systems.

The MQSeries Internet Gateway supports the CGI, ICAPI, ISAPI, and NSAPI Web server interfaces on all of these platforms, with the following exceptions:

- HP-UX does ***not*** support NSAPI.
- Sun Solaris supports CGI ***only***.

Using the Web Interface

Obtaining More Information

The MQSeries product family Web site is at:

<http://www.software.ibm.com/ts/mqseries/>

The following documentation is accessible from this Web site:

- *Getting Started with MQSeries Internet Gateway*. This is the starting point for the download and installation of MQSeries Internet Gateway.
- *MQSeries Internet Gateway User's Guide*. This is the main documentation for users of the MQSeries Internet Gateway.

Obtaining MQSeries Internet Gateway

The MQSeries Internet Gateway is one of the installable components on the MQSeries Server CD-ROM, and is also available from the Web site.

The following Gateways are available:

- MQSeries Internet Gateway for AIX
- MQSeries Internet Gateway for HP-UX
- MQSeries Internet Gateway for OS/2
- MQSeries Internet Gateway for OS/390 OpenEdition®
- MQSeries Internet Gateway for Sun Solaris
- MQSeries Internet Gateway for Windows NT

If you have downloaded a package from the web, go to the *Installing MQSeries Internet Gateway* documentation for a walkthrough of the installation procedure.

Chapter 8. Obtaining Additional Information

This chapter describes the sources of information that can be useful when you are using MQSeries for Sun Solaris or related products.

The MQSeries for Sun Solaris Library

The MQSeries for Sun Solaris Server CD-ROM contains the following documentation for MQSeries for Sun Solaris.

Order Number	Title	Description
MQSeries Books		
GC33-1870	<i>Quick Beginnings</i>	Gives a brief overview of MQSeries for Sun Solaris, and provides information on planning for, installing, and getting started with the product.
GC33-1349	<i>MQSeries Planning Guide</i>	Describes some key MQSeries concepts, identifies items that need to be considered before MQSeries is installed, including storage requirements, backup and recovery, security, and migration from earlier releases, and specifies hardware and software requirements for every MQSeries platform.
SC33-1872	<i>MQSeries Intercommunication</i>	Defines the concepts of distributed queuing, and explains how to set up a distributed queuing network in a variety of MQSeries environments. In particular, it demonstrates how to (1) configure communications to and from a representative sample of MQSeries products, (2) create required MQSeries objects, (3) create and configure MQSeries channels, and (4) establish MQSeries client/server connections. The use of channel exits is also described.
GC33-1632	<i>MQSeries Clients</i>	Describes how to install, configure, use, and manage MQSeries client systems.

MQSeries Library

Order Number	Title	Description
SC33-1873	<i>MQSeries System Administration</i>	Supports day-to-day management of local and remote MQSeries objects. It includes topics such as security, recovery and restart, transactional support, problem determination, and the dead-letter queue handler. It also includes the syntax of the MQSeries control commands.
SC33-1369	<i>MQSeries Command Reference</i>	Contains the syntax of the MQSC commands, which are used by MQSeries system operators and administrators to manage MQSeries objects.
SC33-1482	<i>MQSeries Programmable System Management</i>	Provides both reference and guidance information for users of MQSeries events, programmable command formats (PCFs), and installable services.
GC33-1876	<i>MQSeries Messages</i>	Describes "AMQ" messages issued by MQSeries. This book is available in softcopy only.
SC33-0807	<i>MQSeries Application Programming Guide</i>	Provides guidance information for users of the message queue interface (MQI). It describes how to design, write, and build an MQSeries application. It also includes full descriptions of the sample programs supplied with MQSeries.
SC33-1673	<i>MQSeries Application Programming Reference</i>	Provides comprehensive reference information for users of the MQI. It includes: data-type descriptions; MQI call syntax; attributes of MQSeries objects; return codes; constants; and code-page conversion tables.
SX33-6095	<i>MQSeries Application Programming Reference Summary</i>	Summarizes the information in the <i>MQSeries Application Programming Reference</i> manual.
SC33-1877	<i>MQSeries Using C++</i>	Provides both guidance and reference information for users of the MQSeries C++ programming-language binding to the MQI.
SC34-5390	<i>MQSeries Administration Interface Programming Guide and Reference</i>	Provides both guidance and reference information for users of the MQSeries Administration Interface.
SC34-5349	<i>MQSeries Queue Manager Clusters</i>	Explains the concepts and terminology of MQSeries clustering, and shows how you can benefit by taking advantage of clusters. It summarizes the syntax of new and changed commands and shows a number of examples of tasks you can perform to set up and maintain clusters of queue managers.

Online Information

Order Number	Title	Description
SC34-5456	<i>MQSeries Using Java</i>	Provides both guidance and reference information for users of the MQSeries Bindings for Java and the MQSeries Client for Java.

Viewing Online Information

This section describes viewing:

- Online help
- Online books
- Books on the World Wide Web

Viewing Online Help

Man pages are provided for all API calls, MQSC commands, and relevant control commands including **crtmqm**, **strmqm**, and **endmqm**.

Viewing Online Books

Most of the MQSeries books are available in both hardcopy and softcopy formats.

The MQSeries online documentation is in Adobe Acrobat Portable Document Format (PDF) and HTML, on the server CD-ROM.

To view the MQSeries online documentation in HTML format directly from the CD-ROM, point your web browser to `/BOOKS/HTML/start.htm/` in the CD-ROM file system.

If you have installed the HTML-format documentation, point your web browser to:

```
/opt/mqm/books/html/start.htm
```

When you read the books in HTML, you can follow hypertext links from one book to another. If you are reading translated books and link to a book that is not available in your national language, the U.S. English version of the book will be opened instead. You can then follow links between the English books only, but cannot link back into a translated book. In order to return to the translated book, you must use your browser's back button.

Online Information

BookManager® Books

The MQSeries library is supplied in IBM BookManager format on a variety of online library collection kits, including the *Transaction Processing and Data* collection kit, SK2T-0730. You can view the softcopy books in IBM BookManager format using the following IBM licensed programs:

- BookManager READ/2
- BookManager READ/6000
- BookManager READ/DOS
- BookManager READ/MVS
- BookManager READ/VM
- BookManager READ for Windows

HTML and PDF Books on the World Wide Web

The books listed in “The MQSeries for Sun Solaris Library” on page 95 are supplied as HTML and Portable Document Format on the MQSeries Server CD-ROM, and on the World Wide Web. For the latest information, see “Available from the Internet” on page 100.

Printing and Ordering Books

For information about ordering publications listed in this book, contact your IBM authorized dealer or marketing representative. In the United States, you can also order publications by dialing **1-800-879-2755**. In Canada, you can order publications by dialing **1-800-IBM-4YOU (1-800-426-4968)**.

Printing Portable Document Format (PDF) Books

The MQSeries for Sun Solaris books are supplied in PDF format, which can be printed on a PostScript printer, on the MQSeries for Sun Solaris Server CD-ROM.

Table 4 shows the file names used for the PDF files.

Note: The fifth character in the name represents the national language of the book.
For example:

- A** U.S. English
- F** French
- J** Japanese
- S** Spanish

Book	File Name
<i>MQSeries Planning Guide</i>	CSQZAB02
<i>MQSeries for Sun Solaris Quick Beginnings</i>	AMQDAC02
<i>MQSeries Intercommunication</i>	CSQZAE02
<i>MQSeries Clients</i>	CSQZAF02
<i>MQSeries System Administration</i>	AMQZAG01
<i>MQSeries Programmable System Management</i>	CSQZAI02
<i>MQSeries Command Reference</i>	CSQZAJ02
<i>MQSeries Application Programming Reference</i>	CSQZAK02
<i>MQSeries Application Programming Guide</i>	CSQZAL02
<i>MQSeries Application Programming Reference Summary</i>	CSQZAM02
<i>MQSeries Using C++</i>	AMQZAN02
<i>MQSeries Messages</i>	AMQZA001
<i>MQSeries Administration Interface Programming Guide and Reference</i>	CSQZAT00
<i>MQSeries Queue Manager Clusters</i>	CSQZAH00
<i>MQSeries Using Java</i>	CSQZAW00

Internet Availability

Available from the Internet

The MQSeries product family Web site is at:

<http://www.software.ibm.com/ts/mqseries/>

By following links from this Web site you can:

- Obtain latest information about the MQSeries product family.
- Access the MQSeries books in HTML and PDF formats.
- Download MQSeries SupportPacs.

Appendix A. Sample MQI Programs and MQSC Files

MQSeries for Sun Solaris provides a set of short sample MQI programs and MQSC command files. You can use these directly or modify them for experimental purposes.

MQSC Command File Samples

Table 5 lists the MQSC command file samples. These are simply ASCII text files containing MQSC commands. You can invoke the **runmqsc** command against each file in turn to create the objects specified in the file.

By default, these files are located in directory `/opt/mqm/samp`.

File name	Purpose
amqscic0.tst	Defines objects for use in the sample CICS® transaction.
amqscos0.tst	Creates a set of MQI objects for use with the C and COBOL program samples.
amqmdefs.tst	Defines objects for the administration application sample.

C and COBOL Program Samples

Table 6 lists the sample MQI source files. By default, the source files are `/opt/mqm/samp` and the compiled versions in directory `/opt/mqm/samp/bin`. To find out more about what the programs do and how to use them, see Chapter 31, “Sample programs (all platforms except OS/390)” in the *MQSeries Application Programming Guide*.

C	COBOL	Purpose
amqsbcg0.c	–	Reads and then outputs both the message descriptor and message context fields of all the messages on a specified queue.
amqsecha.c	amqmechx.cbl	Echoes a message from a message queue to the reply-to queue. Can be run as a triggered application program.

Samples

<i>Table 6 (Page 2 of 2). Sample programs - source files</i>		
C	COBOL	Purpose
amqsgbr0.c	amq0gbr0.cbl	Writes messages from a queue to stdout, leaving the messages on the queue. Uses MQGET with the browse option.
amqsget0.c	amq0get0.cbl	Removes the messages from the named queue (using MQGET) and writes them to stdout.
amqsinqa.c	amqminqx.cbl	Reads the triggered queue; each request read as a queue name; responds with information about that queue.
amqsput0.c	amq0put0.cbl	Copies stdin to a message and then puts this message on a specified queue.
amqsreq0.c	amq0req0.cbl	Puts request messages on a specified queue and then displays the reply messages.
amqsseta.c	amqmsetx.cbl	Inhibits puts on a named queue and responds with a statement of the result. Runs as a triggered application.
amqstrg0.c	–	A trigger monitor that reads a named initiation queue and then starts the program associated with each trigger message. Provides a subset of the full triggering function of the supplied runmqtrm command.
amqsvfcx.c	–	A sample C skeleton of a Data Conversion exit routine.
amqsptl0.c	–	Putting messages to a distribution list.
amqsprma.c	–	Putting reference messages to a queue.
amqsgrma.c	–	Getting reference messages from a queue.
amqsxrma.c	–	Reference message channel exit.
Note: You can create the objects required by these samples using the MQSC command file amqscos0.tst.		

Supporting CICS and Encina for Transaction Processing

The samples include a CICS transaction and some associated headers and initialization programs.

<i>Table 7. Samples for transaction processing with CICS and Encina</i>	
File name	Purpose
amqzscix.c	CICS initialization program
amqscic0.ccs	Sample CICS program
amqzscgx.c	GLUE program for CICS for Solaris
amqscih0.h	Header file for CICS transaction sample amqscic0
Note: Objects to support transaction processing can be created using the MQSC command file amqscic0.tst.	

Supporting Tuxedo for Transaction Processing

The samples include client transactions and some associated definitions and configuration files.

<i>Table 8. Samples for transaction processing with Tuxedo</i>	
File name	Purpose
amqstxsx.c	Sample server
amqstxgx.c	Sample GET client application
amqstxpx.c	Sample PUT client application
amqstxvx.flds	Field definition
ubbstxcx.cfg	Configuration file

Samples

Supporting Databases

The database samples are located in the `xatm` subdirectory within the `samples` directory.

<i>Table 9. Sample programs - databases</i>		
C	COBOL	Purpose
amqsxas0.c	amqsxas0.cbl	Updates a single database within an MQSeries unit of work.
amqsxag0.c	amqsxag0.cbl	amqsxag0.c together with amqsxab0.sqc and amqsxaf0.sqc, or amqsxag0.cbl together with amqsxab0.sqb and amqsxaf0.sqb, update two databases within an MQSeries unit of work.

Miscellaneous Tools

These tool files are provided to support the formatter and code conversion.

<i>Table 10. Miscellaneous files</i>		
File name	Location	Purpose
amqtrc.fmt	/opt/mqm/lib	Defines MQSeries trace formats.
ccsid.tbl	/var/mqm/conv/table	Edit this file to add any newly supported CSSID values to your MQSeries system.

Appendix B. Applying maintenance to MQSeries for Sun Solaris

This appendix tells you how to apply maintenance to MQSeries for Sun Solaris.

Maintenance updates in the form of a Program Temporary Fix (PTF) are supplied on CD-ROM. They can also be downloaded from :

<http://www.software.ibm.com/ts/mqseries/>

Attention

Do not have any queue managers operating during installation of maintenance on MQSeries for Sun Solaris.

To end all running queue managers:

1. End the queue manager by issuing the command:

```
endmqm -i QMgrName
```

2. Check that the queue manager has ended.

Use the command:

```
endmqm -w QMgrName
```

The message returning should show that the queue manager is not available.

Alternatively, use the command:

```
ps -ef | grep mq
```

where | is the pipe symbol. Check that there are no processes listed that are running command lines commencing amq or runmq. Ignore any that start with amqi.

3. Issue the **ipcs -a** command to identify any shared memory segments or semaphore sets that were created by MQSeries. Remove these using the **ipcrm** command.

Restoring Levels

Space Requirements

A PTF requires hard disk space for installation. In addition, the installation process requires an identical amount of disk space to save the previous level. For example, a 16 MB PTF requires 32 MB of space.

This allows a PTF to be removed, and the previous level to be automatically restored. If disk space is limited, the backup can be suppressed by creating an empty flag file called MQPTF_NOSAVE in the directory `/var/sadm/pkg`.

Note that if this option is used, the previous level will not be restored if a PTF is removed. The only way to restore a previous level in this instance is to reinstall the product and then to reapply a previous PTF image.

Applying the maintenance information

1. Mount the CD by typing the following commands:

```
mkdir -p /cdrom/mq_solaris
mount -F hsfs -r /dev/dsk/cntndnsn /cdrom/mq_solaris
```

substituting `cntndnsn` with the name of your CD-ROM device.

2. Install the software by entering the following command:

```
pkgadd -d /cdrom/mq_solaris/mqm/patchname
```

For further information on using **pkgadd** to install software packages, see the Solaris documentation.

Restoring the Previous Service Level

To restore the previous service level:

1. Log in as root, or use the command `su`.
2. Use the **pkgrm** command to remove the latest PTF from the system. For example, to remove PTF U443859 issue the following command:

```
pkgrm MQSERIES.U443859
```

Error messages of the form `<shared pathname not removed>` can be ignored.

Details of the **pkgrm** command can be found in the Solaris documentation, or by using the **man pkgrm** command.

3. If you have installed an MQI client, and the client was updated after installing the PTF that is being removed, then you **must** specifically update your MQI client installation again, after the PTF has been removed.

Appendix C. Support for Different Code Sets on MQSeries for Sun Solaris

MQSeries for Sun Solaris supports most of the code sets used by the locales – that is, the subsets of the user's environment which define the conventions for a specific culture – that are provided as standard on MQSeries for Sun Solaris.

If the locale is not set, the value of the LANG environment variable is used. If neither the locale nor LANG environment variable is set the CCSID used is 819 - the ISO 8859-1 code set.

Note: Not all the locales listed below are supported by all versions of Solaris.

See "Migration to Euro Support" on page 110 for information on support for the euro character.

The CCSID (Coded Character Set Identifier) used in MQSeries to identify the code set used for the message and message header data is obtained by analyzing the LC_CTYPE environment variable.

Table 11 shows the locales and the CCSIDs that are registered for the code set used by the locale.

<i>Table 11 (Page 1 of 3). Locales and CCSIDs</i>			
Locale	Language	code set	CCSID
C	English	ISO 8859-1	819
ar	Arabic	ISO 8859-6	1089
ar_AA	Arabic	ISO 8859-6	1089
bu	Bulgarian	ISO 8859-5	915
bu_BG	Bulgarian	ISO 8859-5	915
cs	Czech	ISO 8859-2	912
cs_CZ	Czech	ISO 8859-2	912
da	Danish	ISO 8859-1	819
da_DK	Danish	ISO 8859-1	819
de	German	ISO 8859-1	819
de_DE	German	ISO 8859-1	819
de_AT	German - Austria	ISO 8859-1	819
de_CH	German - Switzerland	ISO 8859-1	819
el	Greek	ISO 8859-7	813
el_GR	Greek	ISO 8859-7	813

Supported code sets

<i>Table 11 (Page 2 of 3). Locales and CCSIDs</i>			
Locale	Language	code set	CCSID
en	English - United Kingdom	ISO 8859-1	819
en_GB	English - United Kingdom	ISO 8859-1	819
en_UK	English - United Kingdom	ISO 8859-1	819
en_AU	English - Australia	ISO 8859-1	819
en_CA	English - Canada	ISO 8859-1	819
en_US	English - USA	ISO 8859-1	819
es	Spanish	ISO 8859-1	819
es_ES	Spanish	ISO 8859-1	819
fi	Finnish	ISO 8859-1	819
fi_FI	Finnish	ISO 8859-1	819
fr	French - France	ISO 8859-1	819
fr_FR	French - France	ISO 8859-1	819
fr_BE	French - Belgium	ISO 8859-1	819
fr_CA	French - Canada	ISO 8859-1	819
fr_CH	French - Switzerland	ISO 8859-1	819
hr	Croatian	ISO 8859-2	912
hr_HR	Croatian	ISO 8859-2	912
hu	Hungarian	ISO 8859-2	912
hu_HR	Hungarian	ISO 8859-2	912
is	Icelandic	ISO 8859-1	819
is_IS	Icelandic	ISO 8859-1	819
it	Italian - Italy	ISO 8859-1	819
it_IT	Italian - Italy	ISO 8859-1	819
it_CH	Italian - Switzerland	ISO 8859-1	819
iw	Hebrew	ISO 8859-8	916
iw_IL	Hebrew	ISO 8859-8	916
ja	Japanese	eucJP	5050
ja_JP	Japanese	eucJP	5050
ja_JP.PCK	Japanese	PCK	943
ko	Korean	eucKR	970
ko_KR	Korean	eucKR	970
mk	Macedonian	ISO 8859-5	915
mk_MK	Macedonian	ISO 8859-5	915
nl	Dutch - Netherlands	ISO 8859-1	819
nl_NL	Dutch - Netherlands	ISO 8859-1	819
nl_BE	Dutch - Belgium	ISO 8859-1	819
no	Norwegian	ISO 8859-1	819
no_NO	Norwegian	ISO 8859-1	819
pl	Polish	ISO 8859-2	912
pl_PL	Polish	ISO 8859-2	912

Supported code sets

<i>Table 11 (Page 3 of 3). Locales and CCSIDs</i>			
Locale	Language	code set	CCSID
POSIX	English	ISO 8859-1	819
pt	Portuguese	ISO 8859-1	819
pt_PT	Portuguese	ISO 8859-1	819
ro	Romanian	ISO 8859-2	912
ro_RO	Romanian	ISO 8859-2	912
ru	Russian	ISO 8859-5	915
ru_RU	Russian	ISO 8859-5	915
ru_SU	Russian	ISO 8859-5	915
sh	Serbocroatian	ISO 8859-2	912
sh_SP	Serbocroatian	ISO 8859-2	912
sh_YU	Serbocroatian	ISO 8859-2	912
sl	Slovene	ISO 8859-2	912
sl_SL	Slovene	ISO 8859-2	912
sk	Slovak	ISO 8859-2	912
sk_SK	Slovak	ISO 8859-2	912
sr	Serbian Cyrillic	ISO 8859-5	915
sr_SP	Serbian Cyrillic	ISO 8859-5	915
sv	Swedish	ISO 8859-1	819
sv_SE	Swedish	ISO 8859-1	819
tr	Turkish	ISO 8859-9	920
tr_TR	Turkish	ISO 8859-9	920
zh	Simplified Chinese	eucCN	1383
zh_TW	Traditional Chinese	eucTW	964
zh_TW.BIG5	Traditional Chinese	BIG5	950

For further information listing inter-platform support for these locales, see Appendix F, "Code page conversion tables" in the *MQSeries Application Programming Reference* book.

Migration to Euro Support

If you want to use the *euro* character with MQSeries, you should first install any operating system updates necessary to display the euro character.

Now modify your MQSeries system:

- Edit the existing CCSID.TBL file to enable the new euro version of the coded character set identifier (CCSID). To do this, remove the # symbol from the required line of the **CCSID Mapping** section of the CCSID.TBL file. When you have done this, all new queue managers you create will adopt the new euro CCSID.

Note: If you want to create a new queue manager with a CCSID that supports the euro character, select a euro-supporting locale. Full details of the euro-supporting locales for MQSeries for Sun Solaris were not available at the time of print. Refer to the README file supplied with the product, and the MQSeries euro Web site at:

<http://www.software.ibm.com/ts/mqseries/support/euro>

for more information.

- To modify any existing queue managers that do not support the euro character, follow this procedure:
 1. Record the existing queue manager CCSID, with the MQSeries (MQSC) command:
`DISplay QMGR CCSID`
 2. Change the CCSID to the euro support CCSID, with the MQSC command:
`ALTer QMGR CCSID`
 3. Stop the queue manager.
 4. Restart the queue manager and any channels it uses.

Note: The `ALTer QMGR CCSID` command is a new command supplied with MQSeries for Sun Solaris V5.1.

Now any new message issued using the queue manager CCSID uses the new euro CCSID. All messages now received using MQGET with conversion and requesting the queue manager CCSID to be used are converted into the euro CCSID. CCSIDs and object text (for example descriptions, definitions, and exit names), from existing messages are not changed.

Now modify your applications to support the euro character. If these use hard coded CCSIDs, ensure they now use the new euro CCSID.

Appendix D. Notices

This information was developed for products and services offered in the United States. IBM may not offer the products, services, or features discussed in this information in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this information. The furnishing of this information does not give you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
U.S.A.

For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

IBM World Trade Asia Corporation
Licensing
2-31 Roppongi 3-chome, Minato-ku
Tokyo 106, Japan

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law:

INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY, OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the information. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this information at any time without notice.

Any references in this information to non-IBM documentation or non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of

Notices

those documents or Web sites. The materials for those documents or Web sites are not part of the materials for this IBM product and use of those documents or Web sites is at your own risk.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

IBM United Kingdom Laboratories,
Mail Point 151,
Hursley Park,
Winchester,
Hampshire,
England
SO21 2JN.

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this information and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Programming License Agreement, or any equivalent agreement between us.

Trademarks

The following terms are trademarks of International Business Machines Corporation in the United States, or other countries, or both:

AIX	AS/400	BookManager
CICS	DB2	eNetwork
ESCON	FFST	IBM
MQ	MQSeries	MVS/ESA
OpenEdition	OS/2	OS/390
RISC System/6000	SupportPac	SP
TXSeries	VisualAge	VM/ESA
VSE/ESA	VTAM	WIN-OS/2

Lotus Notes and LotusScript are trademarks of Lotus Development Corporation in the United States, or other countries, or both.

Java and all Java-based trademarks and logos are trademarks or registered trademarks of Sun Microsystems, Inc. in the United States and/or other countries.

Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States and/or other countries.

UNIX is a registered trademark in the United States and/or other countries licensed exclusively through X/Open Company Limited.

Other company, product, or service names may be the trademarks or service marks of others.

Notices

Index

A

administration command sets
 control commands 81
 MQSeries commands (MQSC) 83
 programmable command format commands (PCF) 84

AIX client
 custom installation 50
 hardware and software 32
 installing 49
 removing 54

application
 data 2
 time-independent 2

applications
 non-version 5 31
 version 5.1 31

attributes
 ALL attribute 90
 changing 91
 default 90

B

bibliography 95
 BookManager 98
 books, translated 22
 browsing queues 92

C

C and COBOL sample programs 101
 case-sensitive control commands 82
 CCSID (coded character set identifier) 107
 setting 29
 changing queue attributes 91
 channel
 events 7
 message 5
 MQI 5
 clearing a local queue 92
 clients 5
 code set 107

coded character set identifier (CCSID) 107
 setting 29
 command set administration 81
 commands
 control 81
 MQSC
 ALTER QLOCAL 91
 DEFINE QLOCAL 90
 DEFINE QLOCAL LIKE 90
 DEFINE QLOCAL REPLACE 91
 DELETE QLOCAL 92
 using 84
 programmable command format (PCF) 84
 runmqsc 87

compilers 11
 configuration
 kernel 20
 configuration, kernel 56, 66
 control commands
 case-sensitive 82
 runmqsc 87
 controlled shutdown 86
 creating
 file system for product code 18
 groups
 client 47, 55, 65
 server 17
 queue manager 85
 user ID 47, 55, 65
 users 17
 current queue depth (CURDEPTH) 90
 custom installation
 AIX client 50

D

databases 11
 DCE 11
 samples 13
 support 13
 default
 attributes of objects 90
 queue manager commands processed 87
 deleting

Index

deleting (*continued*)

local queue 92
queue manager 87

disk requirements for installation 9

DOS client

hardware and software 34
installing 54

E

earlier versions

migrating from version 2.2, or version 5.0 15

ending

interactive MQSC commands 88
queue manager 86

endmqm command 85

environment variable

LANG 22
NLSPATH 22

error log

for automatic installation 61

error messages 88

euro support, migrating to 110

event-driven processing 2

events

channel 7
instrumentation 6
types of 6

F

feedback from MQSC commands 88

file samples

CICS and Encina 103
miscellaneous 104
MQSC 101
Tuxedo 103

file system, creating for product code 18

first failure support technology (FFST) 20

G

groups, creating 17

on AIX 47
on HP-UX 55
on Sun Solaris 65

H

hard disk requirements 9

hardware requirements

AIX client 32
DOS client 34
HP-UX client 35
OS/2 client 37
Sun Solaris client 39
Windows 3.1 client 40
Windows 95 and 98 client 42
Windows NT client 44

history log 61

HP-UX client

hardware and software 35
installing 55

Hypertext Markup Language (HTML) 98

I

information, ordering publications 99

installation

directory (Sun Solaris) 65
kernel configuration 20
parameters (OS/2) 60
preparation 17
procedure 19
response files
OS/2 62
Windows 95 and Windows 98 69
server 17

installation directory 17

installing

AIX client 49
AIX client (custom install) 50
clients on the server 21
clients on Windows 3.1 71
DOS client 54
error log parameters 61
history log parameter 61
HP-UX client 55
maintenance updates 105
OS/2 client 56
OS/2 Warp client 57
Sun Solaris client 65
Windows 95 client 67
Windows 98 client 67
Windows NT client 73

instrumentation events

description 6

Index

instrumentation events (*continued*)
 types of 6
interactive MQSC
 ending 88
 feedback from 88
 using 87
Internet Gateway 93
introduction to MQSeries 1

K

kernel configuration 20, 56, 66
keywords for response files 63

L

LANG environment variable 22
LIKE attribute 90
link to Lotus Notes 15
linking user exits 28
local queues
 clearing 92
 copying definitions 90
 defining one 89
 deleting 92
locale 107
Lotus Notes link 15

M

maintenance 105
maintenance of MQSeries for Sun Solaris
 installing updates 105
 space requirements 106
maintenance parameters 60
message
 channels 5
 description 2
 descriptor 2
 queuing 1
 translated 53, 56, 66
message length, decreasing 91
message queue interface (MQI) 1
message queuing 1
message-driven processing 1
message, translated 22
migrating client definitions 31
migrating from an earlier version 15
migrating from Version 2 MQSeries client products 31

migrating to euro support 110
monitoring queue managers 6
MQI
 channel 5
 description 1
MQSC commands
 ALTER QLOCAL 91
 DEFINE QLOCAL 90
 DEFINE QLOCAL LIKE 90
 DEFINE QLOCAL REPLACE 91
 DELETE QLOCAL 92
 ending interactive input 88
 issuing interactively 87
 using 84
MQSeries for Sun Solaris
 applying maintenance 105
 components 12
 hardware requirements 9
 overview of 9
 restoring previous service level 105
 software requirements 9
MQSeries for Sun Solaris V2.2, migrating from 15
MQSeries for Sun Solaris V5.0, migrating from 15
MQSeries objects 3

N

namelists, description of 5
national language support 22, 53, 56
NLSPATH environment variable 22, 53, 56
non-version 5 clients 31

O

objects
 default attributes 90
 namelist 5
 process definition 4
 queue manager 3
 working with 87
online books 97
ordering publications 99
OS/2 client
 hardware and software 37
 installing 56
 unattended installation 59
overview of MQSeries for Sun Solaris 9

Index

P

- parameters for unattended installation 60
- performance events 7
- Portable Document Format (PDF) 99
- preemptive queue manager shutdown 86
- process definitions, description of 4
- processing, event-driven 2
- program samples 101
 - C and COBOL 101
 - databases 104
- programmable command format (PCF)
 - administration with 84
- publications 95

Q

- queue depth
 - current 90
 - determining 90
- queue manager
 - creating 85
 - deleting 87
 - description 3
 - events 6
 - immediate shutdown 86
 - monitoring 6
 - objects 3
 - preemptive shutdown 86
 - restart 86
 - shutdown
 - controlled 86
 - immediate 86
 - preemptive 86
 - quiesced 86
 - starting 85
 - stopping 85
- queues
 - attributes 4
 - browsing 92
 - changing attributes 91
 - defining 4
 - description 2
 - local
 - clearing 92
 - copying 90
 - defining 89
 - deleting 92
- quiesced shutdown 86

R

- README file 14
- removing an AIX client 54
- removing clients
 - Windows 3.1 72
 - Windows 95 and Windows 98 70
 - Windows NT 79
- requirements, hardware and software 9
- response files
 - installation 62, 69
 - keywords 63
 - structure 64
- restart queue manager 86
- restoring previous service level 106
- runmqsc
 - ending 88
 - feedback 88
 - using interactively 87

S

- sample files
 - CICS and Encina 103
 - miscellaneous 104
 - MQSC 101
 - Tuxedo 103
- sample programs 101
 - C and COBOL 101
 - databases 104
- server installation 17
- setting the CCSID (coded character set identifier) 29
- shell commands for MQSeries 81
- shutdown queue manager
 - controlled 86
 - immediate 86
 - preemptive 86
 - quiesced 86
- silent install
 - Windows 95 and Windows 98 client 68
 - Windows NT client 77
- SMIT
 - installing AIX client 50
 - using to create IDs 48
- SMS
 - creating the MQSeries job 77
 - creating the software package 76
 - using with MQSeries 75
- softcopy information 97
- software requirements

Index

software requirements (*continued*)
 AIX client 32
 DOS client 34
 HP-UX client 35
 OS/2 client 37
 Sun Solaris client 39
 Windows 3.1 client 40
 Windows 95 and 98 client 42
 Windows NT client 44
space requirements
 installation 9
 maintenance 106
starting a queue manager 85
stopping a queue manager 85
strmqm command 85
structure of response files 64
Sun Solaris client
 hardware and software 39
 installing 65
supported code sets 107
syntax error, in MQSC commands 88
System Management Interface Tool (SMIT)
 installing AIX client 50
 using to create IDs 48

T

time-independent applications 2
transaction monitors 11
translated books 22
translated messages
 client 53, 56, 66
 server 22
trigger monitor 4
types of event 6

U

unattended installation
 OS/2 client 59
 Windows 95 and Windows 98 client 68
 Windows NT client 77
updating MQSeries for Sun Solaris 105
user exits, linking 28
user ID, creating
 on AIX 47
 on HP-UX 55
 on Sun Solaris 65
users, creating 17

V

verify installation 22
Version 2 MQSeries products
 migrating, client 31
Version 5.1 clients 31

W

Windows 3.1 client
 hardware and software 40
 installing 71
 removing clients 72
Windows 95 and 98 client
 hardware and software 42
 installing 67
Windows 95 and Windows 98 client
 removing 70
Windows NT
 removing 79
Windows NT client
 hardware and software 44
 installing 73
World Wide Web interface 93

Y

Year 2000 compatibility 9

Sending your comments to IBM

MQSeries® for Sun Solaris

Quick Beginnings

GC33-1870-02

If you especially like or dislike anything about this book, please use one of the methods listed below to send your comments to IBM.

Feel free to comment on what you regard as specific errors or omissions, and on the accuracy, organization, subject matter, or completeness of this book. Please limit your comments to the information in this book and the way in which the information is presented.

To request additional publications, or to ask questions or make comments about the functions of IBM products or systems, you should talk to your IBM representative or to your IBM authorized remarketer.

When you send comments to IBM, you grant IBM a nonexclusive right to use or distribute your comments in any way it believes appropriate, without incurring any obligation to you.

You can send your comments to IBM in any of the following ways:

- By mail, use the Readers' Comment Form (RCF)
- By fax:
 - From outside the U.K., after your international access code use 44 1962 870229
 - From within the U.K., use 01962 870229
- Electronically, use the appropriate network ID:
 - IBM Mail Exchange: GBIBM2Q9 at IBMMAIL
 - IBMLink: HURSLEY(IDRCF)
 - Internet: idrcf@hursley.ibm.com

Whichever you use, ensure that you include:

- The publication number and title
- The page number or topic to which your comment applies
- Your name/address/telephone number/fax number/network ID.

Readers' Comments

MQSeries® for Sun Solaris

Quick Beginnings

GC33-1870-02

Use this form to tell us what you think about this manual. If you have found errors in it, or if you want to express your opinion about it (such as organization, subject matter, appearance) or make suggestions for improvement, this is the form to use.

To request additional publications, or to ask questions or make comments about the functions of IBM products or systems, you should talk to your IBM representative or to your IBM authorized remarketer. This form is provided for comments about the information in this manual and the way it is presented.

When you send comments to IBM, you grant IBM a nonexclusive right to use or distribute your comments in any way it believes appropriate without incurring any obligation to you.

Be sure to print your name and address below if you would like a reply.

Name

Address

Company or Organization

Telephone

Email

MQSeries® for Sun Solaris

MQSeries for Sun Solaris V5.1 Quick Beginnings GC33-1870-02 

You can send your comments POST FREE on this form from any one of these countries:

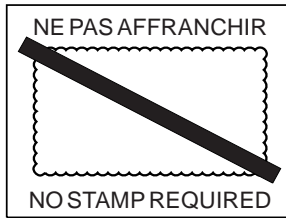
Australia	Finland	Iceland	Netherlands	Singapore	United States
Belgium	France	Israel	New Zealand	Spain	of America
Bermuda	Germany	Italy	Norway	Sweden	
Cyprus	Greece	Luxembourg	Portugal	Switzerland	
Denmark	Hong Kong	Monaco	Republic of Ireland	United Arab Emirates	

If your country is not listed here, your local IBM representative will be pleased to forward your comments to us. Or you can pay the postage and send the form direct to IBM (this includes mailing in the U.K.).

2 Fold along this line

By air mail
Par avion

IBRS/CCRI NUMBER: PHQ - D/1348/SO



REPONSE PAYEE
GRANDE-BRETAGNE

IBM United Kingdom Laboratories
Information Development Department (MP095)
Hursley Park,
WINCHESTER, Hants
SO21 2ZZ United Kingdom

3 Fold along this line

From: Name _____
Company or Organization _____
Address _____

EMAIL _____
Telephone _____

4 Fasten here with adhesive tape

1 Cut along this line

1 Cut along this line





Printed in the United States of America
on recycled paper containing 10%
recovered post-consumer fiber.

GC33-1870-02

