

MQSeries® for AIX®



# Quick Beginnings

*Version 5.2*



MQSeries<sup>®</sup> for AIX<sup>®</sup>



# Quick Beginnings

*Version 5.2*

**Note!**

Before using this information and the product it supports, be sure to read the general information under "Appendix C. Notices" on page 85.

**Fourth edition (December 2000)**

This edition applies to MQSeries for AIX, V5.2, and to all subsequent releases and modifications until otherwise indicated in new editions.

© Copyright International Business Machines Corporation 1995, 2000. All rights reserved.

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

# Contents

<b>Tables</b> . . . . .	<b>v</b>	Verifying the installation of MQSeries for AIX	18
<b>Welcome to MQSeries for AIX</b> . . . . .	<b>vii</b>	Verification procedure. . . . .	19
Road map . . . . .	vii	User exits . . . . .	24
Conventions . . . . .	vii	Setting the queue manager CCSID on MQSeries for AIX . . . . .	25
<b>What's new in MQSeries for AIX, Version 5 Release 2.</b> . . . . .	<b>ix</b>	<b>Chapter 3. Installing the MQSeries for AIX client</b> . . . . .	<b>27</b>
<b>Part 1. Installing MQSeries for AIX</b>	<b>1</b>	Planning to install the MQSeries for AIX client . . . . .	27
<b>Chapter 1. Planning to install the MQSeries for AIX server</b> . . . . .	<b>3</b>	Hardware requirements . . . . .	27
Hardware requirements . . . . .	3	Software requirements . . . . .	27
Disk Storage . . . . .	4	Connectivity . . . . .	27
Software requirements . . . . .	4	Compilers supported for AIX applications	28
Connectivity . . . . .	4	The installation directory. . . . .	28
Compilers supported for MQSeries for AIX, V5.2 applications. . . . .	5	Before installation . . . . .	28
Options . . . . .	5	Migrating from an earlier version of MQSeries for AIX . . . . .	30
Transaction monitors . . . . .	5	Changes to Signal Handling. . . . .	31
Databases . . . . .	5	Changes to the installation path . . . . .	31
DCE . . . . .	5	Installing the MQSeries for AIX client . . . . .	32
Delivery. . . . .	6	Easy installation . . . . .	32
Installation . . . . .	6	SMIT custom installation. . . . .	33
MQSeries for AIX filesets and components . . . . .	6	Changing the national language . . . . .	33
README file . . . . .	8	Translated messages . . . . .	34
<b>Chapter 2. Installing the MQSeries for AIX server</b> . . . . .	<b>9</b>	Verifying the installation . . . . .	34
Preparing for installation . . . . .	9	How does it work?. . . . .	34
Creating the system default objects. . . . .	9	The installation used for the example . . . . .	35
Before installation . . . . .	10	Setting up the server . . . . .	35
Migrating from an earlier version of MQSeries for AIX . . . . .	12	Setting up the client . . . . .	36
Easy installation . . . . .	15	Putting a message on the queue . . . . .	37
Installation procedure. . . . .	15	Getting the message from the queue . . . . .	38
SMIT custom installation. . . . .	16	Ending verification. . . . .	38
Changing the national language . . . . .	17	<b>Chapter 4. Applying maintenance to MQSeries for AIX.</b> . . . . .	<b>39</b>
Installing the server and client on the same machine . . . . .	17	Remote installation. . . . .	39
Support for Java on MQSeries . . . . .	18	Reinstalling MQSeries for AIX . . . . .	40
Translated messages . . . . .	18	Installing a PTF using SMIT. . . . .	40
Translated books . . . . .	18	Restoring the previous service level . . . . .	41
		Removing the MQSeries Internet Gateway	41
		<b>Chapter 5. Uninstalling MQSeries for AIX</b>	<b>43</b>

---

**Part 2. Getting started with MQSeries . . . . . 45****Chapter 6. About MQSeries . . . . . 47**

Introduction . . . . .	47
Messages, queues, and queue managers. . . . .	48
Messages . . . . .	48
Queues . . . . .	48
Queue managers . . . . .	49
MQSeries configurations . . . . .	49
Channels . . . . .	50
Clients and servers. . . . .	51
Clusters . . . . .	51
MQSeries capabilities . . . . .	52
Transactional support . . . . .	52
Instrumentation events . . . . .	53
Message-driven processing . . . . .	54
Programming MQSeries . . . . .	54

**Chapter 7. Using the MQSeries command sets . . . . . 55**

Introducing command sets . . . . .	55
Control commands. . . . .	55
MQSeries (MQSC) commands . . . . .	57
PCF commands . . . . .	58
Working with queue managers. . . . .	58
Creating a default queue manager . . . . .	58
Starting a queue manager . . . . .	59
Stopping a queue manager . . . . .	59
Deleting a queue manager . . . . .	60
Working with MQSeries objects . . . . .	60
Using the MQSC facility interactively . . . . .	60
Ending interactive input to MQSC . . . . .	61
Defining a local queue . . . . .	62
Displaying default object attributes . . . . .	62
Copying a local queue definition . . . . .	63
Changing local queue attributes . . . . .	64
Clearing a local queue . . . . .	64
Deleting a local queue . . . . .	65
Browsing queues . . . . .	65

<b>Chapter 8. Using the MQSeries Internet Gateway . . . . . 67</b>	
Overview of MQSeries Internet Gateway . . . . .	67
MQSeries Internet Gateway documentation . . . . .	68

<b>Chapter 9. Obtaining Additional Information . . . . . 69</b>	
Hardcopy Books . . . . .	69
Online Information. . . . .	70
Publications supplied with the product . . . . .	70
HTML and PDF Books on the World Wide Web. . . . .	72
BookManager CD-ROMs. . . . .	73
Online Help . . . . .	73

---

**Part 3. Appendixes . . . . . 75**

<b>Appendix A. Sample MQI programs and MQSC files . . . . . 77</b>	
MQSC command file samples . . . . .	77
C and COBOL program samples . . . . .	77
Supporting CICS and Encina for transaction processing . . . . .	79
Supporting Tuxedo for transaction processing . . . . .	79
Supporting databases . . . . .	79
Miscellaneous tools . . . . .	80

<b>Appendix B. Code sets supported on MQSeries for AIX . . . . . 81</b>	
Migration to Euro support . . . . .	83

<b>Appendix C. Notices. . . . . 85</b>	
Trademarks . . . . .	87

<b>Index . . . . . 89</b>
---------------------------

<b>Sending your comments to IBM . . . . . 93</b>
--

---

## Tables

	1.	Getting started road map. . . . .	vii	8.	Samples for transaction processing with		
	2.	Filesets and components . . . . .	6		Tuxedo . . . . .	79	
	3.	MQSeries for AIX books . . . . .	69	9.	Sample programs - databases . . . . .	79	
	4.	MQSeries publications – file names	71	10.	Miscellaneous files . . . . .	80	
	5.	MQSC command files . . . . .	77		11.	Locales and CCSIDs for AIX. . . . .	81
	6.	Sample programs - source files . . . . .	77				
	7.	Samples for transaction processing with					
		CICS and Encina . . . . .	79				





---

# Welcome to MQSeries for AIX

This book describes MQSeries for AIX 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 AIX*.

*Table 1. Getting started road map*

If you want to...	Refer to...
Learn about system requirements for MQSeries for AIX	"Chapter 1. Planning to install the MQSeries for AIX server" on page 3
Install MQSeries for AIX	"Chapter 2. Installing the MQSeries for AIX server" on page 9
Install the MQSeries for AIX client	"Chapter 3. Installing the MQSeries for AIX client" on page 27
Applying Maintenance to MQSeries for AIX	"Chapter 4. Applying maintenance to MQSeries for AIX" on page 39
Uninstalling an MQSeries for AIX, server or client	"Chapter 5. Uninstalling MQSeries for AIX" on page 43
Read about MQSeries for AIX	"Chapter 6. About MQSeries" on page 47
Start using command sets	"Chapter 7. Using the MQSeries command sets" on page 55
Start using the Web Interface	"Chapter 8. Using the MQSeries Internet Gateway" on page 67
View or print online documentation	"Chapter 9. Obtaining Additional Information" on page 69
Contact IBM®	<i>Sending your comments to IBM</i>

---

## 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.

## Conventions

- Monospace type indicates an example (such as a fictitious path or file name) or text that is displayed on the screen.

---

## What's new in MQSeries for AIX, Version 5 Release 2

MQSeries for AIX, Version 5 Release 2 provides the following new and changed functions:

- Enhancements have been made to the performance of MQI function, channels, message logging, and application initialization and termination.
- You can now request immediate update of Object Authority Manager (OAM) data, rather than having to stop and restart the queue manager before authorization changes take effect.
- Changes have been made to the way in which OAM data is held, to improve performance.
- Support for Java™ on MQSeries is separately installable from the CD-ROM included in the MQSeries V5.2 product package. Alternatively, you can download the latest version of support for Java on MQSeries from the MQSeries Web site at:  
<http://www.ibm.com/software/mqseries/>
- Support is included for *pipelining*, which is the ability of the Message Channel Agent (MCA) to transfer messages using multiple threads.
- Channel send-exit programs can reserve space in the transmission buffer for their own use. Typically, this would be used by an exit that wanted to encrypt data and add a security key.
- Dynamic Host Configuration Protocol (DHCP) can now be used in queue manager clusters.
- Management of log files for recovery and restart has been improved.
- The area of main storage used to store information relating to a queue manager cluster can be increased dynamically. A new cluster workload-exit call (MQXCLWLN) is provided to support navigation of MQWDR, MQWQR, and MQWCR records held in dynamically increased storage.
- Minor changes to the MQSeries application programming functions have been made, including: support for MQRFH2 (the version-2 rules and formatting header); improvements to the processing of the *CodedCharSetId* field in MQSeries headers; the addition of a command-level value MQCMD\_LEVEL\_520; and C++ support for MQCNO Version 2 and Version 3.
- Draft 10 threading is supported.
- IBM eNetwork Communications Server for AIX, V5.0 is supported.
- IBM WebSphere™ is supported as an XA coordinator.
- The way in which UNIX® signals are handled by MQSeries has been altered to minimize the impact on user applications.

## What's new

| For a complete description of new and changed function in this product, see  
| the *MQSeries V5.2 Release Guide*.

# Part 1. Installing MQSeries for AIX

<b>Chapter 1. Planning to install the MQSeries for AIX server</b> . . . . .	3	<b>Chapter 3. Installing the MQSeries for AIX client</b> . . . . .	27
Hardware requirements . . . . .	3	Planning to install the MQSeries for AIX client . . . . .	27
Disk Storage . . . . .	4	Hardware requirements . . . . .	27
Software requirements . . . . .	4	Software requirements . . . . .	27
Connectivity . . . . .	4	Connectivity . . . . .	27
Compilers supported for MQSeries for AIX, V5.2 applications . . . . .	5	Compilers supported for AIX applications . . . . .	28
Options . . . . .	5	The installation directory. . . . .	28
Transaction monitors . . . . .	5	Before installation . . . . .	28
Databases . . . . .	5	Creating another file system for product code . . . . .	29
DCE . . . . .	5	Creating the user ID and group . . . . .	29
Delivery. . . . .	6	Migrating from an earlier version of MQSeries for AIX . . . . .	30
Installation . . . . .	6	Changes to Signal Handling. . . . .	31
MQSeries for AIX filesets and components . . . . .	6	Changes to the installation path . . . . .	31
README file . . . . .	8	Installing the MQSeries for AIX client . . . . .	32
<b>Chapter 2. Installing the MQSeries for AIX server</b> . . . . .	9	Easy installation . . . . .	32
Preparing for installation . . . . .	9	Installation procedure. . . . .	32
Creating the system default objects. . . . .	9	SMIT custom installation. . . . .	33
Before installation . . . . .	10	Changing the national language . . . . .	33
Creating another file system for product code. . . . .	11	Translated messages . . . . .	34
Creating the user ID and group . . . . .	11	Verifying the installation . . . . .	34
Migrating from an earlier version of MQSeries for AIX . . . . .	12	How does it work?. . . . .	34
Changes to Signal Handling. . . . .	14	The installation used for the example . . . . .	35
Changes to the installation path . . . . .	14	What the example shows. . . . .	35
Easy installation . . . . .	15	Setting up the server . . . . .	35
Installation procedure. . . . .	15	Setting up the client . . . . .	36
SMIT custom installation. . . . .	16	Defining a client-connection channel, using MQSERVER . . . . .	37
Changing the national language . . . . .	17	Putting a message on the queue . . . . .	37
Installing the server and client on the same machine . . . . .	17	Getting the message from the queue . . . . .	38
Support for Java on MQSeries . . . . .	18	Ending verification. . . . .	38
Translated messages . . . . .	18	<b>Chapter 4. Applying maintenance to MQSeries for AIX</b> . . . . .	39
Translated books . . . . .	18	Remote installation. . . . .	39
Verifying the installation of MQSeries for AIX . . . . .	18	Reinstalling MQSeries for AIX . . . . .	40
Verification procedure. . . . .	19	Installing a PTF using SMIT. . . . .	40
Verifying a local installation. . . . .	19	Restoring the previous service level . . . . .	41
Verifying a server-to-server installation . . . . .	21	Removing the MQSeries Internet Gateway . . . . .	41
User exits . . . . .	24	<b>Chapter 5. Uninstalling MQSeries for AIX</b> . . . . .	43
Setting the queue manager CCSID on MQSeries for AIX . . . . .	25		



---

## Chapter 1. Planning to install the MQSeries for AIX server

This chapter is a summary of the requirements for running MQSeries for AIX including:

- Network protocols
- Compilers
- Delivery media
- Various components of the product

The following information applies to the server environment only. For information about installing the MQSeries for AIX client, see “Chapter 3. Installing the MQSeries for AIX client” on page 27.

### 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 for AIX, V5.2 runs on any machine that is capable of running AIX V4.3.2, and that has sufficient storage to meet the combined requirements of

- The programming prerequisites
- MQSeries for AIX, V5.2
- The access methods

## Hardware requirements

- The application programs  
whether from IBM or other vendors. For example:
- IBM RS/6000® POWERserver®
- IBM RS/6000 POWERstation
- IBM Scalable POWERparallel® systems
- Bull DPX/20 (RISC)
- Bull ESCALA (SMP)

### 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, the number and size of the messages on the queues, and whether the messages are persistent. You also require archiving capacity on disk, tape, or other media.

These are the approximate storage requirements:

- Base code and server: A minimum of 52 MB of disk space must be available for the product code and data.

**Note:** Working data for MQSeries for AIX is stored by default in `/var/mqm`. You need to create two file systems before installation – `/var/mqm/qmgrs` and `/var/mqm/log`.

- HTML files: 35MB

**Note:** Use the `df` command to determine the amount of free space on your disk.

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

- AIX Versions 4.3.2, and 4.3.3

**Note:** To determine what level operating system you are running use the `oslevel` command.

---

## Connectivity

The network protocols supported by MQSeries for AIX are:

- TCP/IP
- SNA LU 6.2



In MQSeries for AIX, V5.2, enhancements are introduced which make it more practical to combine the use of DHCP with MQSeries queue manager clusters.

For SNA LU 6.2 you need IBM eNetwork™ Communications Server for AIX V6.0

---

## Compilers supported for MQSeries for AIX, V5.2 applications

- IBM C compiler for AIX Version 4.4
- IBM VisualAge® C++ Professional for AIX Version 5.0
- Merant Server Express, V1.1 (was Micro Focus)

---

## Options

You may use MQSeries for AIX with the following options.

### Transaction monitors

The following transaction processing monitors (coordination through X/Open XA interface) are supported:

- BEA TUXEDO Version 6.4 or 6.5
- WebSphere 3.0x

**Note:** MQSeries for AIX, V5.2 supports WebSphere as an XA coordinator. For more information about the MQSeries application adaptor, and about how to write Component Broker applications please see the *WebSphere Application Server Enterprise Edition Component Broker MQSeries Application Adaptor Development Guide SC09-4444*

### Databases

- Oracle 8i and 8iR2 (8.1.5 or 8.1.6)
- DB2® Universal Database V5.0, V6.1 and 7
- Sybase V11:
  - Adaptive Server Enterprise (A.S.E.), V11.5
  - Open Client (ctlib and dblib), V11.1
  - Embedded SQL/C, V11.0
  - XA Server, V11.1
- Sybase Adaptive Server Enterprise, V12, with DTM option.

For more information on how to set up XA coordination please refer to the *MQSeries System Administration* book.

### DCE

- This must be a DCE product that supports DES data encryption if you want to run the MQSeries- supplied DCE send, receive, or message exits.
- IBM DCE V2.2
- DCE for AIX, 3.1

## Options

DCE names and security modules are provided with MQSeries for AIX, V5.2.

---

## Delivery

MQSeries for AIX, V5.2 is supplied on CD-ROM.

Support for Java on MQSeries is separately installable from the CD-ROM included in this product package. Alternatively, you can download support for Java on MQSeries from the MQSeries Web site at <http://www.ibm.com/software/mqseries>

where the latest version of this support is always available

---

## Installation

MQSeries for AIX, V5.2 is installed using **smit**. Detailed instructions are provided in “Chapter 2. Installing the MQSeries for AIX server” on page 9

The installation can be performed in approximately 15 minutes.

---

## MQSeries for AIX filesets and components

MQSeries for AIX, V5.2 contains the following filesets (sometimes called components):

*Table 2. Filesets and components*

Title	Description	Fileset
MQSeries Base Kit for Client and Server. Also known as the Software Development Kit (SDK)	Enables you to create and support your own applications. Requires the runtime component to be installed.	mqm.base.sdk
MQSeries Runtime for Client and Server	Support for external applications. This does <b>not</b> enable you to write your own applications.	mqm.base.runtime
MQSeries Samples	Sample application programs.	mqm.base.samples
MQSeries Client for AIX	The MQSeries for AIX Client can be installed on the server machine, enabling you to have the MQSeries server and client on the same machine.	mqm.client.rte
MQSeries Support for DCE in Samples	This should be installed <i>only</i> if you are going to use DCE.	mqm.dce.samples

Table 2. Filesets and components (continued)

Title	Description	Fileset
MQSeries Support for DCE in Server	Support for DCE names or security on the server. This should be installed <i>only</i> if you are going to use DCE.	mqm.dce.server
MQSeries Internet Gateway — Runtime		mqm.gateway.rte
MQSeries Internet Gateway — Runtime Data		mqm.gateway.rte.data
MQSeries Internet Gateway — Samples	Provides access to MQSeries applications through ICAPI, ISAPI, NSAPI, HTML and CGI.	mqm.gateway.samples
MQSeries Internet Gateway — Samples Data		mqm.gateway.samples.data
MQSeries Online Documentation	This documentation is provided in the following languages: Base Brazilian Portuguese French German Italian Japanese Korean Simplified Chinese Spanish Traditional Chinese U.S. English	mqm.html.base.doc mqm.html.pt_BR.doc mqm.html.fr_FR.doc mqm.html.de_DE.doc mqm.html.it_IT.doc mqm.html.ja_JP.doc mqm.html.ko_KR.doc mqm.html.zh_CN.doc mqm.html.es_ES.doc mqm.html.zh_TW.doc mqm.html.en_US.doc
MQSeries Man Pages — U.S.English	Man pages for the following commands: • Control commands • Message Queue Interface (MQI) • MQSeries commands (MQSC)	mqm.man.en_US.data
MQSeries Server	Support for client connections. Requires the runtime component to be installed.	mqm.server.rte
MQSeries Client Bundle	Contains: MQSERIES Runtime for Client and Server MQSERIES Base Kit for Client and Server MQSERIES Samples MQSERIES Client for AIX	mqm.Client.Bnd mqm.base.runtime  mqm.base.sdk  mqm.base.samples mqm.client.rte

## MQSeries for AIX filesets

Table 2. Filesets and components (continued)

Title	Description	Fileset
MQSeries Server Bundle	Contains: MQSERIES Runtime for Client and Server MQSERIES Base Kit for Client and Server MQSERIES Samples MQSERIES Server for AIX	mqm.Server.Bnd mqm.base.runtime  mqm.base.sdk  mqm.base.samples mqm.server.rte

---

### README file

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

To do this you must first add the file system for the root directory using smit.

---

## Chapter 2. Installing the MQSeries for AIX server

This chapter tells you how to install the MQSeries for AIX server using either “Easy Installation” or “SMIT Custom Installation” options, and how to verify that your installation has been successful. It also tells you how to migrate from an earlier version of MQSeries for AIX - see “Migrating from an earlier version of MQSeries for AIX” on page 12.

The “Easy Installation” option gives a minimal configuration on your machine, consisting of the following filesets:

- MQSeries Runtime for Client and Server
- MQSeries Base Kit for Client and Server
- MQSeries Server
- MQSeries Samples

If you want any of the other components, such as online documentation, you *must* use the “SMIT Custom Installation” option – see “SMIT custom installation” on page 16.

For MQSeries for AIX, the name of the installation directory is `/usr/mqm`. 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 product code” on page 11.

---

### Preparing for installation

This section guides you through the steps you must perform before you install MQSeries for AIX, V5.2

If you already have MQSeries for AIX, V5.0, or MQSeries for AIX, V5.1 make sure you refer to “Migrating from an earlier version of MQSeries for AIX” on page 12 before installing MQSeries for AIX, V5.2.

### 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 V5.0, and you want to use the same settings with V5.2 of the product:

1. Save your copy of `amqscoma.tst`
2. Install MQSeries V5.2

## Preparation

3. Load your copy of amqscoma.tst and use the file to recreate your default objects.

### Before installation

Before you can install MQSeries for AIX you are recommended to create and mount a /var/mqm file system, or /var/mqm, /var/mqm/log, and /var/mqm/errors file systems for your data.

**Note:** These are journalized 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.

For a single file system, use the sum of these figures as a guide.

#### Notes:

1. To determine the size of the /var/mqm file system you should consider:
  - The maximum number of messages in the system at one time
  - Contingency for message build-ups, if there is a system problem
  - The average size of the message data, plus 500 bytes for the message header
  - The number of queues
  - The size of log files and error messages if these are not going to be in a separate file system
2. It is better to use all the space available, monitor the usage for a few weeks and, if appropriate, reduce the amount of space allocated. If the initial file system is too small, this can cause problems later.
3. If you create separate file systems, the following directories *must* be on a local file system:
  - /var/mqm
  - /var/mqm/log

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

  - The size of the log file depends upon the log settings that you use. The size recommended is for circular logging (you will need to check if this appropriate) using the default settings. For further information on log sizes see the *MQSeries System Administration* book.
4. 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 user ID and group” on page 11 for more information), but make sure you do this before installing the server. User ID and group must both be mqm. For

stand-alone machines, you can create the new user and group IDs locally, and for machines administered in a network information services (NIS) domain, you must create the 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.

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`.

It might be necessary to modify the IPCC pool segment number if you wish to use a middleware product such as Websphere. This modification, made through the `mq5.ini` file, is described in the informational Apar II12418.

The `ini` value must not be changed unless you have one of the exact product combinations listed.

### Creating another file system for product code

If you do not want to have the product code installed in the `/usr/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 `/usr/mqm`.
2. Create a new directory anywhere on your machine that is large enough to contain the product, and create a symbolic link from `/usr/mqm` to this new directory. For example:

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

### 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.

### Creating the user ID and group

Use the following procedure *before you install the server*, if you want to create the required IDs yourself. For example, you might wish to do this if you are setting up all security groups before installing MQSeries.

You must create both user ID and group as `mqm`. If you do not want to create the IDs yourself, they are created automatically when you install the server.

## Preparation

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:

```
smit
  Security & Users
    Groups
      Add a Group
```

Set the name field to `mqm`.

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

```
smit
  Security & Users
    Users
      Add a User
```

Set the name field to `mqm`

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

```
smit
  Security & Users
    Change a Users Password
      Passwords
```

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

```
smit
  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 Name field. Add `mqm` to the **Group SET** field, which is a comma-separated list of the groups to which the user belongs.

**Note:** You must add **root** to the `mqm` group. Users need not have their primary group set to `mqm`. As long as `mqm` is in their set of groups, they can use the commands. Users who are running only applications that use the queue manager do not need `mqm` group authority.

### Migrating from an earlier version of MQSeries for AIX

It is strongly recommended that you backup all

- Queue manager data
- Queue manager logs



- Queue manager object definitions

before starting to migrate to a new version of MQSeries.

To migrate from an earlier version of MQSeries for AIX to MQSeries for AIX V5.2 you need to:

- End all MQSeries activity on the target machine
- Update any currently installed filesets, and install any new filesets that might be required

To update your currently installed filesets, you should:

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

```
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, by typing:

```
smit update_latest
```

3. Press the **List** button to display the **Single Select List** window, and select **/dev/cd0 (CD-ROM Drive)**.
4. Press **OK** to display the parameters for Update All.
5. For correct migration, you must update all previously installed MQSeries software. To achieve this, select the **\_update\_all** option in the **Software to Update** field.
6. Press **Enter**.
7. Press **OK** on the confirmation window to begin the software updating process.

Once all previously installed filesets have been updated to the latest level, any additional filesets can be installed. Use the method "SMIT custom installation" on page 16 to install additional filesets.

Version 5.2 of MQSeries for AIX uses the POSIX standard threading library (which was not available on AIX V4.2) to match the implementation on other UNIX platforms. Existing MQSeries applications built on AIX 4.2 using the draft 7 level of POSIX threads are not affected by this new implementation and will continue to run unchanged. However, MQSeries exits and installable services should be recompiled and relinked using the `xlC_r` compiler on AIX 4.3 to use the final level of the `pthread` standard definition (also known as the

## Migrating from an earlier version

draft 10 level). It is recommended that any new threaded applications on AIX 4.3 also be written to use this level of the pthreads standard.

### Changes to Signal Handling

In MQSeries for AIX, V5.2, the way in which signals are handled has changed. These changes, and their effects on your existing applications, are described in the *MQSeries V5.2 Release Guide*.

### Changes to the installation path

Changes in AIX packaging mean that MQSeries for AIX, V5.1 and V5.2 installs into directory `/usr/mqm`, whereas MQSeries for AIX V5.0 installs into directory `/usr/lpp/mqm`.

Installation of MQSeries for AIX, V5.2 fails if a filesystem mounted as `/usr/lpp/mqm` is detected. If you are migrating from an earlier version and a filesystem exists for this directory, you need to do one of the following things before installing MQSeries for AIX, V5.2. Either:

- Uninstall your existing MQSeries product, and either delete the filesystem or move it to the new install path of `/usr/mqm`
- or
- Move the old filesystem 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 MQSeries product and either delete or move your existing filesystem, you can then install MQSeries for AIX, V5.2 as described in “SMIT custom installation” on page 16 or “Easy installation” on page 15.

However, if you move the old filesystem to the new installation path, you should perform the migration installation described in “Migrating from an earlier version of MQSeries for AIX” on page 12.

**Note:** If you have already symbolically linked a filesystem to `/usr/lpp/mqm`, installation of MQSeries for AIX, V5.2 destroys the filesystem content and the link, and you are left with an empty filesystem. If this is the case, you are recommended to uninstall your existing MQSeries for AIX product and delete the filesystem or relink it to the new install path of `/usr/mqm`, before installing MQSeries for AIX, V5.2.

The installation process for MQSeries for AIX, V5.2 creates a symbolic link from the old `/usr/lpp/mqm` path to the new `/usr/mqm` path. Therefore any existing scripts or makefiles that reference the old path are still valid.

## Easy installation

This section describes the “Easy Installation” process from the server CD-ROM.

### Notes:

1. The process described in this section places a minimal typical installation on your machine. If you want to install additional components including the online documentation, and application development support, you *must* use the System Management Interface Tool (SMIT). See “SMIT custom installation” on page 16.
2. If you already have MQSeries for AIX installed, see “Migrating from an earlier version of MQSeries for AIX” on page 12.
3. To change your installation or back out a failed installation you will need to uninstall using smit.

### Installation procedure

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

**Note:** The `xinstallm` program is available in the `X11.vsm.rte` optional fileset.

The MQSeries Welcome window is displayed, followed by a second window where you can make some choices:

- Choose the software source: **CD-ROM**.
- For **Which bundle of software would you like to install?** choose: **Media-defined**.

4. Click on **Install/Update**.

Two bundles of software products are created. You will find them among all the other bundles that are defined on your system:

```
mqm.Client      mqm.Server
```

5. To install the server code, choose the `mqm.Server` bundle and click on **Install/Update** again.

A bundle of software products is created.

A work in progress window gives information as the installation proceeds.

6. 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.

## SMIT installation

---

### SMIT custom installation

**Note**

If you already have MQSeries for AIX installed, see “Migrating from an earlier version of MQSeries for AIX” on page 12

Use SMIT for the installation as follows:

1. Go into SMIT with root authority. From the shell, type:  
smit
2. 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

You can use the alternative fastpath command instead:

```
smitty install_latest
```

3. Press the **List** button to display the input device or directory for the software.
4. Select:  
/dev/cd0 (CD-ROM Drive)  
then press **OK**.
5. Use the **Select software to install** field to obtain a list of components to install.  
  
**Note:** The filesets are displayed on more than one page.
6. Follow the SMIT instructions to select the components you want to install. See “MQSeries for AIX filesets and components” on page 6 for a list of the components.
7. If you have a previous version of the product on your machine, change **AUTOMATICALLY Install prerequisite software** to **No** and **OVERWRITE existing version** to **Yes**.
8. Press **OK** to install.

**Note:** You do not need to install all the filesets. For example, if you want to use U.S. English only, just install the appropriate fileset.

---

### 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.

To check the initial locale setting for your machine, type:

```
smitty mle_cc_cust_hdr
```

It is possible to install the MQSeries 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, and then **OK**.

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.

---

### Installing the server and client on the same machine

To install an MQSeries for AIX 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.

## Installing server and client

For information about installing the client on a different machine from the server, see “Chapter 3. Installing the MQSeries for AIX client” on page 27.

---

## Support for Java on MQSeries

If you have applications that require support for Java on MQSeries, you will need to install it separately after installing MQSeries.

Support for Java on MQSeries is separately installable from the CD-ROM included in this product package. Alternatively, you can download support for Java on MQSeries from the MQSeries Web site at

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

, where the latest version of this support is always available

---

## Translated messages

Messages in U.S. English are always available. If you require another of the languages that is supported by MQSeries for AIX, V5.2, you *must* ensure that your NLSPATH environment variable includes the appropriate directory, and the NLV message catalog is installed.

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

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

**Note:** To find out which language is currently installed, type `echo $LANG`.

---

## 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 might 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 “Online Information” on page 70 for more information about hypertext linking between translated books.

---

## Verifying the installation of MQSeries for AIX

This section describes how to verify that MQSeries for AIX has been correctly installed and configured. You do this by following the steps outlined in “Verification procedure” on page 19.

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 the *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 21. For information on verifying a client/server installation, see “Chapter 3. Installing the MQSeries for AIX client” on page 27.

### 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 AIX 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.

## Verifying the installation

**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 then press Enter.

A message tells you when the queue manager has started.

4. Enable MQSC commands by typing `runmqsc` and then press Enter.

**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 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 **amqspout** (to put a message on the queue) and **amqsget** (to get the message from the queue):

1. Change into the following directory (if it is not in your PATH):

```
/usr/mqm/samp/bin
```

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

```
./amqspout 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.

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

```
/usr/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 some 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 `strmqm` and then press Enter.

A message tells you when the queue manager has started.

## Verifying the installation

3. Enable MQSeries Commands (MQSC) by typing `runmqsc` and then press Enter.

**Note:** MQSC has started when the following message is displayed:

```
Starting MQSeries Commands
```

MQSC has no command prompt.

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)
```

The value `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 typing **end**, and pressing Enter.

Now set up the receiver workstation.

### Receiver workstation:

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
```

**Note:** You must be logged in as a superuser, or as root, to perform step 1 on page 22 to step 3.

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 /usr/mqm/bin/amqcrsta amqcrsta
```

**Note:** If you are not creating `venus.queue.manager` as the **default** queue manager (in step 4) on this workstation, add `-m venus.queue.manager` to the end of this line to specify the name of the queue manager to use.

3. Enter the command `refresh -s inetd`.
4. 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.

5. Start the queue manager:
  - Type the following and then press Enter:

```
strmqm
```

A message tells you when the queue manager has started.

6. Enable 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.

7. 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.

## Verifying the installation

8. Create a receiver channel:

```
define channel (first.channel) chltype (rcvr) trdtype (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`

9. Stop MQSC by typing **end** and pressing Enter.

### 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 by entering the following command:

```
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

Check that your user exits are linked with threaded libraries before using them on this version of the product.

See the *MQSeries Application Programming Guide* for further details on threaded libraries.

---

## Setting the queue manager CCSID on MQSeries for AIX

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
  uses the code set IS08859-1
  and will set a CCSID of 819
export LANG=En_US
  uses the code set IBM-850
  and will set a CCSID of 850
export LANG=C (this is the default locale)
  uses the code set IS08859-1
  and will set a CCSID of 819
```

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

1. Enable MQSeries (MQSC) commands by typing `runmqsc`.
2. Record the existing queue manager CCSID, using the MQSC command:  
`DISplay QMGR CCSID`
3. Change the CCSID to the new CCSID with the MQSC command:  
`ALTer QMGR CCSID (no. of ccsid)`
4. Stop the MQSC commands by typing `END`.
5. Stop the queue manager by typing `endmqm VENUS.QUEUE.MANAGER` (or whatever you call your queue).
6. Restart the queue manager and any channels it uses.

See “Appendix B. Code sets supported on MQSeries for AIX” on page 81 for further information about supported code sets. See “Migration to Euro support” on page 83 for information on migrating to a CCSID that supports the euro character.



---

## Chapter 3. Installing the MQSeries for AIX client

This chapter describes how to:

- Plan for the installation of the MQSeries for AIX client.
- Install the MQSeries for AIX client on a different machine from the MQSeries for AIX server, using the MQSeries client CD-ROM.
- Verify communication between the MQSeries for AIX client and an MQSeries server.

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

---

### Planning to install the MQSeries for AIX client

This section identifies:

- The hardware and software required by the MQSeries for AIX client
- Tasks you must perform before installing the MQSeries for AIX client

#### Hardware requirements

The MQSeries for AIX client can be installed on any machine capable of running AIX 4.3.2 and that has sufficient storage to meet the combined requirements of:

- Programming requisites
- MQSeries for AIX, client code
- Access methods
- Application programs

#### Software requirements

The MQSeries for AIX client can be installed on AIX versions 4.3.2, and 4.3.3.

**Note:** To determine what version of the operating system you are running, use the `oslevel` command.

#### Connectivity

The network protocols supported by MQSeries for AIX, client are:

- TCP/IP configured and initialized in the operating system.
- SNA LU 6.2

For SNA LU 6.2 you need IBM eNetwork Communications Server for AIX V6.0

## Planning client installation

### Compilers supported for AIX applications

- IBM C compiler for AIX Version 4.4
- IBM VisualAge C++ Professional for AIX Version 5.0
- Merant Server Express, V1.1 (was Micro Focus)

### The installation directory

The MQSeries for AIX client is installed into the `/usr/mqm` directory. This cannot be changed.

### Before installation

To install the MQSeries for AIX client you must use the MQSeries client CD-ROM, supplied as part of the MQSeries product.

#### Notes:

1. If you plan to install the client on the same machine as the server, please see “Installing the server and client on the same machine” on page 17.
2. If you already have an MQSeries for AIX client installed see “Migrating from an earlier version of MQSeries for AIX” on page 30.

Before you can install the MQSeries for AIX client, you:

- Must check the README file for latest information.
- Must create a group with the name `mqm`.
- Must create a user ID with the name `mqm`.
- Create and mount either a `/var/mqm` file system, or `/var/mqm`, `/var/mqm/log`, and `/var/mqm/errors` file systems.

#### Notes:

1. If you create separate partitions, the directory `/var/mqm` must be on a local file system.  
You can NFS mount the `/var/mqm/errors` and `/var/mqm/trace` directories to conserve space on your local system.
2. After installation, the 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.
3. For stand-alone machines, you can create the new user and group IDs locally (or allow them to be created automatically during the installation). For machines administered in a network information services (NIS) domain, you must create the user and group IDs on the NIS master server machine before beginning the installation.



### Creating another file system for product code

If you do not want to have the product code installed in the `/usr/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 `/usr/mqm`.
2. Create a new directory anywhere on your machine that is large enough to contain the product, and create a symbolic link from `/usr/mqm` to this new directory. For example:

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

### 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.

### Creating the user ID and group

Use the following procedure *before you install the client*, if you want to create the required IDs yourself. You must create both user ID and group as `mqm`. If you do not want to create the IDs yourself, they are created automatically when you install the client.

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:

```
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:

## Planning client installation

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, fill in 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 must add **root** to the `mqm` group. Users need not have their primary group set to `mqm`. As long as `mqm` is in their set of groups, they can use the commands. Users who are running only applications that use the queue manager do not need `mqm` group authority.

---

## Migrating from an earlier version of MQSeries for AIX

It is strongly recommended that you backup all

- Queue manager data
- Queue manager logs
- Queue manager object definitions

before starting to migrate to a new version of MQSeries.

To migrate from an earlier version of MQSeries for AIX to MQSeries for AIX V5.2 you need to:

- End all MQSeries activity on the target machine
- Update any currently installed filesets, and install any new filesets that might be required

To update your currently installed filesets, you should:

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

```
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, by typing:

```
smitty update_latest
```

3. Press the **List** button to display the **Single Select List** window, and select **/dev/cd0 (CD-ROM Drive)**.
4. Press **OK** to display the parameters for Update All.
5. For correct migration, you must update all previously installed MQSeries software. To achieve this, select the **\_update\_all** option in the **Software to Update** field.
6. Press **Enter**.
7. Press **OK** on the confirmation window to begin the software updating process.

Once all previously installed filesets have been updated to the latest level, any additional filesets can be installed. Use the method “SMIT custom installation” on page 33 to install additional filesets.

MQSeries for AIX 5.2 supports draft 10 and draft 4 threads. Draft 7 threads are no longer supported, and it is necessary to recompile your applications using the draft 10 libraries. For more information, see the *MQSeries V5.2 Release Guide*.

### Changes to Signal Handling

In MQSeries for AIX, V5.2, the way in which signals are handled has changed. These changes, and their effects on your existing applications, are described in the *MQSeries V5.2 Release Guide*.

### Changes to the installation path

Changes in AIX packaging mean that MQSeries for AIX, V5.1 and V5.2 installs into directory **/usr/mqm**, whereas MQSeries for AIX V5.0 installs into directory **/usr/lpp/mqm**.

Installation of MQSeries for AIX, V5.2 fails if a filesystem mounted as **/usr/lpp/mqm** is detected. If you are migrating from an earlier version and a filesystem exists for this directory, you need to do one of the following things before installing MQSeries for AIX, V5.2. Either:

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

or

- Move the old filesystem 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 MQSeries product and either delete or move your existing filesystem, you can then install MQSeries for AIX, V5.2 as described in “SMIT custom installation” on page 16 or “Easy installation” on page 15.

## Migrating from an earlier version

However, if you move the old filesystem to the new installation path, you should perform the migration installation described in “Migrating from an earlier version of MQSeries for AIX” on page 12.

**Note:** If you have already symbolically linked a filesystem to `/usr/lpp/mqm`, installation of MQSeries for AIX, V5.2 destroys the filesystem content and the link, and you are left with an empty filesystem. If this is the case, you are recommended to uninstall your existing MQSeries for AIX product and delete the filesystem or relink it to the new install path of `/usr/mqm`, before installing MQSeries for AIX, V5.2.

The installation process for MQSeries for AIX, V5.2 creates a symbolic link from the old `/usr/lpp/mqm` path to the new `/usr/mqm` path. Therefore any existing scripts or makefiles that reference the old path are still valid.

---

## Installing the MQSeries for AIX client

If you already have an MQSeries for AIX client installed see “Migrating from an earlier version of MQSeries for AIX” on page 30.

### Easy installation

This section describes the “Easy Installation” process from the client CD-ROM.

#### Notes:

1. The process described in this section places a minimal typical installation on your machine. If you want to install additional components including the on-line documentation, and application development support, you *must* use the System Management Interface Tool (SMIT). See “SMIT custom installation” on page 33.

#### Installation procedure

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 followed by a second window where you can make some choices.
4. *Make sure you are installing the correct client for your system, as displayed in the Welcome Window.*
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 is created:

|           mqm.Client

- |           8. To install the client code, choose the `mqm.Client` bundle and click on **Install/Update** again.

|           A bundle of software products is created.

|           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.

|           Now go to “Verifying the installation” on page 34.

## | **SMIT custom installation**

|           Use SMIT for the installation as follows:

- |           1. Go into SMIT with root authority. From the shell, type:

|           smit

- |           2. 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

|           You can use the alternative fastpath command instead:

|           smitty install\_latest

- |           3. Press the **List** button to display the **Single Select List** window.
- |           4. Select:
- |            /`dev/cd0` (**CD-ROM Drive**)
- |           5. Press **Do** to display the parameters for **Install Latest Level**.
- |           6. Press **F4** to obtain a list of components to install.
- |           7. Follow the SMIT instructions to select the components you want to install. See “MQSeries for AIX filesets and components” on page 6 for a list of the components.
- |           8. Press **Enter**.
- |           9. 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**.
- |           10. Press **Do** to install.

|           Now go to “Verifying the installation” on page 34.

---

## | **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.

## Changing national language

It is possible to install the MQSeries 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 mle_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 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.iso88591
export NLSPATH=/usr/lib/nls/msg/%L/%N
```

---

## Verifying the installation

You can verify your MQSeries client and server installation using the supplied sample PUT and GET programs. These verify that your installation has been successfully completed and that the communication link is working.

### How does it work?

Instructions are given on how to use the supplied sample PUT and GET programs to verify that an MQSeries client has been installed correctly, by guiding you through the following tasks:

1. Setting up the server
2. Setting up the MQSeries client

3. Putting a message on the queue
4. Getting the message from the queue
5. Ending verification

### The installation used for the example

These instructions assume that:

- The full MQSeries product has been installed on a UNIX server machine. If you are connecting the MQSeries for AIX client to a non UNIX MQSeries server, please see the *MQSeries Clients* book for verification instructions.
- The MQSeries for AIX client software has been installed on the client machine.
- TCP/IP is configured and initialized on both the server and the client machines.

### What the example shows

The following example shows how to create a queue manager called *queue.manager.1*, a local queue called *QUEUE1*, and a server-connection channel called *CHANNEL1* on the server. It shows how to create the client-connection channel on the MQSeries client workstation; and how to use the sample programs to put a message onto a queue, and then get the message from the queue.

**Note:** MQSeries object definitions are case-sensitive. You must type the examples *exactly* as shown.

**Security:** The example does *not* address any client security issues. See the *MQSeries Clients* book for details if you are concerned with MQSeries client security issues.

### Setting up the server

On the server, create a directory to hold working files (called *mqverify*, for example), and make this the current directory. Follow these steps to set up the server workstation:

1. Create a default queue manager called *saturn.queue.manager* by entering the following command at the command prompt:  

```
crtmqm -q saturn.queue.manager
```
2. Start the queue manager by typing *strmqm* and then press Enter.
3. Start MQSeries commands (MQSC) by entering *runmqsc* and the press Enter.
4. MQSC does not provide a prompt, but should respond with the message:  
Starting MQSeries Commands
5. Create a local queue called *QUEUE1* by entering the following command:  

```
DEFINE QLOCAL(QUEUE1)
```

## Verifying the installation

6. Create a server-connection channel by entering the following command:

```
DEFINE CHANNEL(CHANNEL1) CHLTYPE(SVRCONN) TRPTYPE(TCP) MCAUSER(' ')
```

7. Stop MQSC by typing end and pressing Enter.

8. Configure a listener to start the MQI channels.

You can use either the standard UNIX TCP listener or the MQSeries listener to start the MQI channels. For performance reasons, the MQSeries listener is likely to be the better choice.

To configure the MQSeries listener to start MQI channels, use the **runmqtsr** control command:

```
runmqtsr -m queue manager -t TCP -p port number
```

To run the job in the background, add an ampersand character (&) to the end of the command.

To configure the inetd daemon to start the MQI channels, you must be logged in as a superuser, or as root.

- a. 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
```

1414 is the default port number. If you are using a different port, specify its number instead.

- b. 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 saturn.queue.manager is not the default queue manager on this workstation, you need to add -m saturn.queue.manager to the end of this line.

- c. Run the command:

```
kill -1 processid of inetd daemon
```

## Setting up the client

When an MQSeries application is run on the MQSeries client, the information it requires is:

- The name of the MQI channel
- The communications protocol
- The address of the server

You provide this information by defining a client-connection channel. The name used for this channel must be the name used for the server-connection channel defined on the server. In this example the MQSERVER environment variable is used to define the client-connection channel. This is the simplest way, although not the only one.



Before starting, type `ping server-address` (where `server-address` is the TCP/IP host name of the server) to confirm that your MQSeries client and server TCP/IP sessions have been initialized. You can use the network address, in the format `n.n.n.n`, in the **ping** command instead of the host name.

If the **ping** command fails, check that your TCP/IP software is correctly configured and has been started.

### Defining a client-connection channel, using MQSERVER

Create a client-connection channel by setting the `MQSERVER` environment variable, as follows:

```
export MQSERVER=CHANNEL1/TCP/'server-address(port)'
```

where:

#### CHANNEL1

Is the name of the server-connection channel already defined on the server.

**TCP** Is the communications protocol.

*server-address*

Is the TCP/IP host name of the server.

*(port)* Is optional and is the TCP/IP port number the server is listening on. If you do not give a port number MQSeries uses the one specified in the `QM.INI` file. If no value is specified in the `QM.INI` file, MQSeries uses the port number identified in the TCP/IP services file for the service name `MQSeries`. If this entry in the services file does not exist, a default value of 1414 is used. It is important that the port number used by the client and the port number used by the server listener program are the same.

### Putting a message on the queue

On the MQSeries client workstation, put a message on the queue using the `AMQSPUTC` sample program:

1. From a command prompt window, change to the directory (`/usr/mqm/samp/bin`) containing the sample program `amqsputc`.
2. Enter the following command:

```
amqsputc QUEUE1 saturn.queue.manager
```

where `saturn.queue.manager` is the name of the queue manager on the server.

The following message is displayed:

```
Sample AMQSPUT0 start
target qname is QUEUE1
```

## Verifying the installation

3. Type some message text and press Enter twice.

The following message is displayed:

```
Sample AMQSPUT0 end
```

The message is now on the queue.

### Getting the message from the queue

On the MQSeries client workstation, get a message from the queue using the `amqsgetc` sample program:

- From the directory containing the sample programs, enter the following command:

```
amqsgetc QUEUE1 saturn.queue.manager
```

Where `saturn.queue.manager` is the name of the queue manager on the server.

The message is removed from the queue and displayed.

### Ending verification

The verification process is now complete.

You can stop the queue manager on the server by typing:

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

If you want to delete the queue manager on the server, enter

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

---

## Chapter 4. Applying maintenance to MQSeries for AIX

This chapter tells you how to apply maintenance to MQSeries for AIX.

Service updates are supplied on CD-ROM. SMIT can be used to transfer PTFs (program temporary fixes), also known as CSDs, from a CD-ROM to your file system. They can then be applied, and committed, from either CD-ROM or an install directory.

### Attention

Do not have any queue managers operating during installation of maintenance on MQSeries for AIX. To end all running queue managers issue the commands:

```
endmqm -i Qmname  
endmq1sr -m Qmname
```

---

### Remote installation

To install MQSeries for AIX on a remote machine, you can use standard AIX techniques. First, you have to create an installation image in backup format, and then use **installp** or **SMIT** to install the product.

To do this, you must be logged on to both systems as root. The product CD-ROM must be in the CD-ROM drive of the machine from which you are going to take the copy. Follow this procedure for each target machine on which you want to install the product:

1. Create a "cdrom" file system on the local machine, by issuing the command `smitty crcdrfs`.
2. Mount the cdrom file system on the local machine by issuing the command `mount /cdrom`.
3. Log on to the remote machine and issue the NFS command `mount local_machine:/cdrom` to mount the cdrom file system that you created. where `local_machine` is the Id of the machine from which you are installing.
4. Use the standard AIX installation procedures, for example SMIT, to install MQSeries for AIX from the target directory that you specified.

## Remote installation

### Reinstalling MQSeries for AIX

If you reinstall MQSeries for AIX, a check is made to see if the MQSeries configuration file (`mqs.ini`) exists. If the file exists, it is kept and used with the newly installed system. If the file does not exist, an empty `mqs.ini` file is placed in the directory `/var/mqm`.

---

### Installing a PTF using SMIT

Installing a PTF (program temporary fix):

1. Log on as root (or use command `su`)
2. Select the device appropriate for your installation using this sequence of windows:

```
smit
Software Installation and Maintenance
  Install and Update Software
    Install/Update From All Available Software
```

Alternatively, enter the fastpath command:

```
smitty install_update
```

Type or select a value for the entry field:

```
INPUT device / directory for software
```

Use PF4 to display a list of valid values:

```
/dev/fd0 (Diskette Drive)
/usr/sys/inst.images (Install Directory)
/dev/rmt0.1 (Tape Drive)
/dev/cd0 (CD-ROM Drive)
```

3. Complete the following fields:

```
SOFTWARE to install
```

Enter **ALL** to install all applicable fileset updates to your installation.

**Note:** Although there is an option to apply only selected fileset updates for MQSeries for AIX, this still results in all applicable fileset updates for the PTF being applied.

4. This will apply the PTF, and save a backup copy of the current level of MQSeries for AIX.

If you might want to commit the PTF updates or reject the PTF updates and return to the backup level at a later time, you *must* ensure that:

- The 'COMMIT software updates' value is set to 'no'.
- The 'SAVE replaced files' value is set to 'yes'.

Press Enter to display a confirmation message, prior to starting the update. The command will run, display progress messages, and should complete

without problems. These progress messages terminate with an Installp Summary table, confirming which components of MQSeries for AIX have been updated.

If the command does not complete successfully, then a full error log is saved in the file `smit.log` in your HOME directory.

5. The same process applies to MQI client installations. Refer to the *MQSeries Clients* book for more information about client installation.

## Restoring the previous service level

When, after installing a PTF, any component of MQSeries for AIX is in APPLIED state, the PTF can be backed out and these selectable components can be restored to the previous service/install level.

To back out a PTF:

1. Log on as root (or use command `su`)
2. To display the appropriate SMIT panel, enter:

```
smit
Software Installation and Maintenance
Maintain Installed Software
Reject Applied Software Updates (Use Previous Version)
```

Alternatively, enter the fastpath command:

```
smitty install_reject
```

3. Complete the following fields:

```
SOFTWARE name
```

Enter **MQM** to restore all applicable fileset updates to your installation.

**Note:** Although there is an option to restore only selected fileset updates for MQSeries for AIX, this still results in all applicable fileset updates for the PTF being restored.

4. You are recommended to use the displayed default values for all other fields. This will reject the current PTF level and reinstate the previous service or install level.

Press ENTER to display a confirmation message, prior to starting the reject. The command will run, display progress messages, and should complete successfully. These progress messages terminate with an Installp Summary table, confirming which components of MQSeries for AIX have been rejected.

## Removing the MQSeries Internet Gateway

After removing the Web server configuration, as described in the *MQSeries Internet Gateway User's Guide* (available on the MQSeries Web site):

1. Log on as root (or use the command `su`).

## PTF installation

2. To display the appropriate SMIT panel, enter:

```
smit
Software Installation and Maintenance
  Maintain Installed Software
    Remove Software Products
```

Add the following to the Remove Software Products screen:

```
SOFTWARE name      dmqqgate.runtime
PREVIEW only       no
```

Leave the other entries with the default settings, and press Enter to remove the product.

---

## Chapter 5. Uninstalling MQSeries for AIX

Use SMIT in the usual way to remove all MQSeries for AIX files that have been installed.

If for any reason the product was not properly installed, you will have to delete the files and directories contained in `/usr/mqm`.





---

## Part 2. Getting started with MQSeries

<b>Chapter 6. About MQSeries</b> . . . . .	47	<b>Chapter 8. Using the MQSeries Internet Gateway</b> . . . . .	67
Introduction . . . . .	47	Overview of MQSeries Internet Gateway . . . . .	67
Messages, queues, and queue managers. . . . .	48	MQSeries Internet Gateway documentation . . . . .	68
Messages . . . . .	48		
Queues . . . . .	48		
Queue managers . . . . .	49	<b>Chapter 9. Obtaining Additional Information</b> . . . . .	69
MQSeries configurations . . . . .	49	Hardcopy Books . . . . .	69
Channels . . . . .	50	Online Information. . . . .	70
Clients and servers. . . . .	51	Publications supplied with the product . . . . .	70
Clusters . . . . .	51	HTML . . . . .	70
MQSeries capabilities . . . . .	52	PDF. . . . .	71
Transactional support . . . . .	52	HTML and PDF Books on the World Wide Web. . . . .	72
Instrumentation events . . . . .	53	BookManager CD-ROMs. . . . .	73
Message-driven processing . . . . .	54	Online Help . . . . .	73
Programming MQSeries . . . . .	54		
<b>Chapter 7. Using the MQSeries command sets</b> . . . . .	55		
Introducing command sets . . . . .	55		
Control commands. . . . .	55		
Using control commands. . . . .	55		
MQSeries (MQSC) commands . . . . .	57		
Running MQSC commands . . . . .	57		
PCF commands . . . . .	58		
Working with queue managers. . . . .	58		
Creating a default queue manager . . . . .	58		
Starting a queue manager . . . . .	59		
Stopping a queue manager . . . . .	59		
Quiesced shutdown . . . . .	59		
Immediate shutdown . . . . .	59		
Preemptive shutdown. . . . .	60		
Deleting a queue manager . . . . .	60		
Working with MQSeries objects . . . . .	60		
Using the MQSC facility interactively . . . . .	60		
Feedback from MQSC commands . . . . .	61		
Ending interactive input to MQSC . . . . .	61		
Defining a local queue . . . . .	62		
Displaying default object attributes . . . . .	62		
Copying a local queue definition . . . . .	63		
Changing local queue attributes . . . . .	64		
Clearing a local queue . . . . .	64		
Deleting a local queue . . . . .	65		
Browsing queues . . . . .	65		



---

## Chapter 6. About MQSeries

This chapter introduces IBM MQSeries. It describes its basic functions and its relationships with operating systems, applications, and other middleware products.

---

### Introduction

MQSeries is a communications system that provides assured, asynchronous, once-only delivery of data across a broad range of hardware and software platforms.

These characteristics make MQSeries the ideal infrastructure for application-to-application communication, and make it an appropriate solution whether the applications run on the same machine or on different machines that are separated by one or more networks.

MQSeries supports all the important communication protocols and even provides routes between networks that use different protocols. MQSeries bridges and gateway products allow easy access (with little or no programming) to many existing systems and application environments—for example, Lotus<sup>®</sup> Notes<sup>™</sup>, Web browsers, Java applets, and many others.

The assured delivery capability reflects the many functions built in to MQSeries to ensure that data is not lost because of failures in the underlying system or network infrastructure. Assured delivery enables MQSeries to form the backbone of critical communication systems and to be entrusted with delivering high-value data. There are also options that allow you to select a less robust quality of service, where this is appropriate. For example, there might be circumstances where you might prefer faster delivery with less emphasis on assured delivery.

The asynchronous processing support in MQSeries means that the exchange of data between the sending and receiving applications is time independent. This allows the sending and receiving applications to be decoupled so that the sender can continue processing, without having to wait for the receiver to acknowledge that it has received the data. In fact, the target application does not even have to be running when the data is sent. Likewise, the entire network path between the sender and receiver may not need to be available when the data is in transit.

## Introduction

Once-only delivery of data is a vital consideration, particularly in financial and business applications where duplicate requests to move large sums of money from one account to another are precisely what you do not want to happen!

---

## Messages, queues, and queue managers

The three fundamental concepts in MQSeries that you need to understand are:

- Messages
- Queues
- Queue managers

### Messages

A *message* is a string of bytes that has meaning to the applications that use it. Messages are used for transferring data 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.

### Queues

A *queue* is a data structure in which messages are stored. The messages may be put on, or got from, 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 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 to, and receive messages from, queues. For example, one application can put a message on a queue, and another application can get the message from the same queue.

Each queue has *queue attributes* that determine what happens when applications reference the queue. 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)

### 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 details received.
- Special events (such as instrumentation events or triggering) are generated when the appropriate conditions are met.
- Messages are put on the correct queue, as requested by the application. The application is informed if this cannot be done, and an appropriate reason code is given.

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 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.

---

### MQSeries configurations

In the simplest configurations, MQSeries is installed on a machine and a single queue manager is created. This queue manager then allows you to define queues. Local applications can then use these queues to exchange messages.

Communication by applications with queues managed by another queue manager requires *message channels* to be defined. It is not necessary to define a channel directly to the target queue manager and it is often appropriate to define one only to the next hop (that is, an intermediate queue manager). Message channels available from that queue manager will be used to deliver the message to the target queue manager (or even to a subsequent hop).

## MQSeries configurations

More complex configurations can be created using a client-server structure. The MQSeries product can act as an MQSeries server to MQSeries clients. The clients and server do not need to be on the same platform. MQSeries supports a broad range of client platforms. The MQSeries products typically include clients for a variety of platforms. Additional MQSeries clients are available from the MQSeries Web site.

In a client-server configuration, the MQSeries server provides messaging and queuing services to the clients, as well as to any local applications. The clients are connected to the server through dedicated channels (known as *client channels*) for clients. This is a cost-effective deployment method because a server can support hundreds of clients with only a single copy of the MQSeries server product. However, the client channel must be continuously available whenever the MQSeries applications on the client are running. This contrasts with the message channels, which need not be continuously available to support MQSeries applications running on the server.

See “Channels” for more information.

MQSeries also supports the concept of *clusters* to simplify setup and operation. A cluster is a named collection of queue managers and any one queue manager can belong to none, one, or several such clusters. The queue managers in a cluster can exist on the same or different machines.

There are two major benefits from the use of clusters:

1. Communication between members of a cluster is greatly simplified, particularly because the channels required for exchanging messages are automatically defined and created as needed.
2. Some or all of the queues of participating queue managers can be defined as being cluster queues, which has the effect of making them automatically known and available to all other queue managers in the cluster.

See “Clusters” on page 51 for more information.

### Channels

A channel provides a communication path to a queue manager. 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 transmitting 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. Two message channels are required if two-way communication is required between two queue managers.

A *client channel* (also known as an *MQI channel*) connects an MQSeries client to a queue manager on a server machine and is bidirectional.

If you want to read more information about channels and how MQSeries uses them to communicate across the systems in your network, see the *MQSeries Intercommunication* book.

### 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 MQSeries calls from applications and pass them to an *MQSeries 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 *MQSeries 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 manager machine (that is, on the MQSeries server machine). A server can support local MQSeries applications as well.

The difference between an MQSeries server and an ordinary queue manager is that the MQSeries server can support MQSeries clients, and each MQSeries client application has a dedicated communication link with the MQSeries server.

For more information about client support, see the *MQSeries Clients* book.

### Clusters

A cluster is a named collection of queue managers.

Clusters require that at least one of the queue managers in the cluster be defined as a *repository* (that is, a place where the shared cluster information can be held). More typically, two or more such repositories are usually designated to provide continued availability in the case of system failure. MQSeries makes sure that the information in the repositories is synchronized.

When a queue is defined as a cluster queue, it can be regarded as a public queue in that it is freely available to other queue managers in the cluster. This contrasts with noncluster queues, which are accessible only when a local definition of them is available. Thus, a non-cluster queue has the characteristics of a private queue, accessible only to those queue managers that have been configured to know about them.

Public queues with the same name in the same cluster are regarded as equivalent. If a message is sent to that queue name, MQSeries (by default)

## MQSeries configurations

sends it to any one of the instances, using a load-balancing algorithm. If you do not want this to happen, you can use the queue manager and queue name in the address, thus forcing the message to be delivered to a specific queue manager. Alternatively, you can replace the load-balancing routine with a different implementation. This is typical of MQSeries, in that there are many examples of where standard behavior can be changed by implementing user code in exits designed for this purpose.

You can read a full explanation in the *MQSeries Queue Manager Clusters* book.

---

## MQSeries capabilities

MQSeries can be used to create many different types of solutions. Some exploit the platform support, or the bridge and gateway capabilities, to connect existing systems in an integrated way or to allow new applications to extract information from, or interchange information with, existing systems. Other solutions support business application servers, where a central pool of MQSeries applications can manage work sent across networks. Complex routing of information for workflow scenarios can be supported. Publish/subscribe or “send and forget” are other application scenarios that use different message flows. Load balancing and hot-standby systems can be built using the power and flexibility of MQSeries, which includes specific functions to support many of these diverse scenarios.

See the *MQSeries Application Programming Guide* for more information about writing MQSeries applications.

### Transactional support

An application program may need to group a set of updates into a *unit of work*. Such updates are usually logically related and must all be successful for data integrity to be preserved. Data integrity would be lost if one update in the group succeeded while another failed. MQSeries supports transactional messaging.

A unit of work *commits* when it completes successfully. At this point all updates made within that unit of work are made permanent and irreversible. Alternatively, all updates are *backed out* if the unit of work fails. *Syncpoint coordination* is the process by which a unit of work is 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 IBM CICS<sup>®</sup>, IBM Transaction Server, IBM TXSeries<sup>™</sup>, Transarc Encina, 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 MQSeries 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.

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 when the queue manager restarts.

### 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, a *Queue Full* event message is generated if: Queue Full events are enabled for a specified queue; an application issues an MQPUT call to put a message on that queue; and 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

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.

## Capabilities

MQSeries instrumentation events are categorized as follows:

### Queue manager events

These 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.

### Performance events

These 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 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.

## Message-driven processing

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

---

## Programming MQSeries

MQSeries applications can be developed using a variety of programming languages and styles. Procedural and object-oriented programming is supported, depending on the MQSeries platform, using, for example, Visual Basic<sup>®</sup>, C, C++, Java, COBOL, and PL/I.

MQSeries function is logically divided into what is normally required by applications (such as putting messages on a queue) and what is necessary for administration (such as changing queue or queue manager definitions). Application function is known as the *MQI* (message queue interface). Administration function is known as the *MQAI* (message queuing administration interface). Applications can mix MQI and MQAI functionality, as required.

The administration functions can be implemented in two ways:

1. Most often, using MQAI language bindings
2. Sending messages to administration queues, to achieve the same results as with the MQAI, using programmable command formats (PCFs)

---

## Chapter 7. 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.

---

### 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, but 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 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.

### Using control commands

In MQSeries in UNIX environments, you enter MQ 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**.

## MQSeries command sets

- 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.

### **dspmqsrv (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.

### **dspmqrn (display MQSeries transactions)**

Displays details of in-doubt transactions.

### **endmqcsrv (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.

### **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.

**runmqlsr (run listener)**

Runs a listener process.

**runmqsc (run MQSeries commands)**

Issues MQSC commands to a queue manager.

**runmqtmc (start client trigger monitor)**

Invokes a trigger monitor for a client.

**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.

For more information about the syntax and purpose of control commands, see 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.

For detailed information about each MQSC command, see the *MQSeries MQSC 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

## MQSeries command sets

For more information about using MQSC commands, see 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 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 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
```

#### **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
```

## Working with queue managers

### 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.

### 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 the *MQSeries System Administration* book. For a complete description of the MQSC commands, see the *MQSeries MQSC 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 58.

### 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)
```

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 the *MQSeries MQSC 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.

## Working with objects

### 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".
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 64.

### 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 62, 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.

## 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**

```
/usr/mqm/samp/amqsbcg0.c
```

**Executable**

```
/usr/mqm/samp/bin/amqsbcg
```

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 8. Using the MQSeries Internet Gateway

This chapter introduces the MQSeries Internet Gateway. It also explains how to get more information about using the 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 MQSeries Web site.

The following Gateways are available:

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

---

### Overview of MQSeries Internet Gateway

MQSeries Internet Gateway provides a bridge between the synchronous World Wide Web and asynchronous MQSeries applications. With the MQSeries Internet 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 MQSeries Internet Gateway is through HTML fill-out form POST requests; MQSeries applications respond by returning HTML pages to the MQSeries Internet Gateway, via an MQSeries queue.

The MQSeries Internet Gateway supports the following Web server interfaces:

- Common Gateway Interface (CGI)
- Internet Connection Application Programming Interface (ICAPI)
- Internet Services Application Programming Interface (ISAPI)
- Netscape Connection Application Programming Interface (NSAPI)

Note that:

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

### MQSeries Internet Gateway documentation

The MQSeries product family Web site is at:

<http://www.ibm.com/software/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.



---

## Chapter 9. Obtaining Additional Information

This chapter describes the documentation for MQSeries for AIX. It starts with a list of the publications, and then discusses:

- “Hardcopy Books”
- “Online Information” on page 70

MQSeries for AIX is described in the following books:

*Table 3. MQSeries for AIX books*

Order Number	Title
<b>AIX Specific Books</b>	
GC33-1867	<i>MQSeries for AIX Quick Beginnings</i>
<b>MQSeries Family Books</b>	
GC34-5761	<i>MQSeries V5.2 Release Guide</i>
SC33-1872	<i>MQSeries Intercommunication</i>
SC34-5349	<i>MQSeries Queue Manager Clusters</i>
GC33-1632	<i>MQSeries Clients</i>
SC33-1873	<i>MQSeries System Administration</i>
SC33-1369	<i>MQSeries MQSC Command Reference</i>
SC33-1482	<i>MQSeries Programmable System Management</i>
SC34-5390	<i>MQSeries Administration Interface Programming Guide and Reference</i>
GC33-1876	<i>MQSeries Messages</i>
SC33-0807	<i>MQSeries Application Programming Guide</i>
SC33-1673	<i>MQSeries Application Programming Reference</i>
SX33-6095	<i>MQSeries Programming Interfaces Reference Summary</i>
SC33-1877	<i>MQSeries Using C++</i>

---

### Hardcopy Books

The book that you are reading now is *MQSeries for AIX, V5.2 Quick Beginnings*. This book and the *MQSeries V5.2 Release Guide* are the only books that are supplied in hardcopy with the product. However, all books listed in Table 3 are available for you to order or print.

You can order publications from the IBMLink™ Web site at:

## Hardcopy Books

<http://www.ibm.com/ibmlink>

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)**.

For further information about ordering publications contact your IBM authorized dealer or marketing representative.

For information about printing books, see “PDF” on page 71.

---

## Online Information

This section describes:

- “Publications supplied with the product”
- “HTML and PDF Books on the World Wide Web” on page 72
- “BookManager CD-ROMs” on page 73
- “Online Help” on page 73

### Publications supplied with the product

On the product CD-ROM there is a directory called books. The books directory contains MQSeries books in HTML and PDF formats. To access them point your Web browser to `books/start.htm`.

#### HTML

You can view the MQSeries online documentation in HTML format directly from the CD-ROM. All books except for the *MQSeries Programming Interfaces Reference Summary* are available in U.S. English and also in some or all of the following national languages:

- French
- German
- Italian
- Japanese
- Korean
- Spanish
- Simplified Chinese

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.

### PDF

A PDF (Portable Document Format), corresponding to each hardcopy book, is available on the CD-ROM. You can read PDFs using Adobe Acrobat Reader. Also, you can download them to your own file system, or you can print them on a PostScript printer. If you have a Web browser, you can access the PDFs on the product CD-ROM by pointing your browser to `books/start.htm`.

The PDFs are available in U.S. English and also in some or all of the following national languages:

- French
- German
- Italian
- Japanese
- Korean
- Spanish
- Simplified Chinese

To find out which ones are available in your language, look for the appropriate directory on the CD-ROM. The PDFs are in a subdirectory called `ll_LL`, where `ll_LL` is one of the following:

- `en_US` (English)
- `fr_FR` (French)
- `de_DE` (German)
- `it_IT` (Italian)
- `ja_JP` (Japanese)
- `ko_KR` (Korean)
- `es_ES` (Spanish)
- `zh_CN` (Simplified Chinese)

Within these directories, you can find the complete set of PDFs that are available. Table 4 shows the file names used for the PDF files.

*Table 4. MQSeries publications – file names*

<b>Book</b>	<b>File Name</b>
<i>MQSeries for AIX Quick Beginnings</i>	AMQAAC03
<i>MQSeries V5.2 Release Guide</i>	AMQZAY00
<i>MQSeries Intercommunication</i>	CSQZAE04
<i>MQSeries Queue Manager Clusters</i>	CSQZAH02
<i>MQSeries Clients</i>	CSQZAF04
<i>MQSeries System Administration</i>	AMQZAG01
<i>MQSeries MQSC Command Reference</i>	CSQZAJ04

## Online Information

Table 4. MQSeries publications – file names (continued)

Book	File Name
<i>MQSeries Programmable System Management</i>	CSQZAI03
<i>MQSeries Administration Interface Programming Guide and Reference</i>	CSQZAT01
<i>MQSeries Messages</i>	AMQZA001
<i>MQSeries Application Programming Guide</i>	CSQZAL04
<i>MQSeries Application Programming Reference</i>	CSQZAK04
<i>MQSeries Programming Interfaces Reference Summary</i>	CSQZAM04
<i>MQSeries Using C++</i>	AMQZAN03

---

### HTML and PDF Books on the World Wide Web

The MQSeries books are available on the World Wide Web as well as on the product CD-ROM. They are available in PDF and HTML format. The MQSeries product family Web site is at:

<http://www.ibm.com/software/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.

---

## BookManager CD-ROMs

The MQSeries library is supplied in IBM BookManager<sup>®</sup> 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<sup>®</sup>

---

## Online Help

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



---

## Part 3. Appendixes





---

## Appendix A. Sample MQI programs and MQSC files

MQSeries for AIX 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 **/usr/mqm/samp**.

*Table 5. MQSC command files*

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 **/usr/mqm/samp** and the compiled versions in directory **/usr/mqm/samp/bin**. To find out more about what the programs do and how to use them, see the *MQSeries Application Programming Guide*.

*Table 6. Sample programs - source files*

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.

## C and COBOL

Table 6. Sample programs - source files (continued)

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 <b>runmqtrm</b> 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.
<b>Note:</b> 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.

*Table 7. Samples for transaction processing with CICS and Encina*

File name	Purpose
amqzscix.c	CICS initialization program
amqscic0.ccs	Sample CICS program
amqzscgx.c	GLUE program for CICS for AIX
amqscih0.h	Header file for CICS transaction sample amqscic0
amqsxae0.c	Encina transaction
<b>Note:</b> You can create objects to support transaction processing using the MQSC command file amqscic0.tst.	

---

## Supporting Tuxedo for transaction processing

The samples include client transactions and some associated definitions and configuration files.

*Table 8. Samples for transaction processing with Tuxedo*

File name	Purpose
amqstxsx.c	Sample server
amqstxgx.c	Sample GET client application
amqstpx.c	Sample PUT client application
amqstvx.flds	Field definition
ubbstxc.cfg	Configuration file

---

## Supporting databases

The database samples are located in the xatm subdirectory within the samples directory.

*Table 9. Sample programs - databases*

C	COBOL	Purpose
amqsxas0.c	amqsxas0.cbl	Updates a single database within an MQSeries unit of work.

## Databases

Table 9. Sample programs - databases (continued)

C	COBOL	Purpose
amqxsag0.c	amqxsag0.cbl	amqxsag0.c, with amqxsab0.sqc and amqsfaf0.sqc, or amqxsag0.cbl, with amqxsab0.sqb and amqsfaf0.sqb, updates two databases within an MQSeries unit of work.

---

## Miscellaneous tools

These tool files are provided to support the formatter and code conversion.

Table 10. Miscellaneous files

File name	Location	Purpose
amqtrc.fmt	/usr/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. Code sets supported on MQSeries for AIX

MQSeries for AIX supports most of the code sets used by the AIX locales – that is, the subsets of the user’s environment that define the convention for a specific culture – that are provided as part of the AIX operating system.

The locale can be selected by setting the **LANG** environment variable to the locale value. If the LANG environment variable is not set, AIX selects the ‘C’ locale.

The CCSID (Coded Character Set Identifier), used in MQSeries to identify the code set used for the message and message header data, is determined by reading the code set name associated with the locale and using the CCSID number registered by IBM for that code set.

Table 11 lists most of the locales supported on AIX, and the CCSID that MQSeries for AIX uses for that locale. New locales are constantly being added for AIX. For a definitive list see the AIX documentation.

### Notes:

1. Not all the locales are supported by all versions of AIX.  
To get more information on the locales supported and their code sets, see the AIX InfoExplorer topic *Understanding Locale*.
2. Installing national language locales is optional and some can be large. If you wish to install a large language locale, you might have to install an AIX fileset to hold it.  
Use the AIX command `locale -a` to see which locales are installed on your AIX system.  
Use the AIX command `locale` to see which locale you have selected.

Table 11. *Locales and CCSIDs for AIX*

Locale	code set	CCSID	Locale	code set	CCSID
ar_AA	ISO8859-6	1089	Ar_AA	IBM-1046	1046
be_BY	ISO8859-5	915	bg_BG	ISO8859-5	915
C	ISO8859-1	819	ca_ES.IBM-1252	IBM-1252	1252
ca_ES.ISO8859-1	ISO8859-1	819	Ca_ES	IBM-850	850
cs_CZ	ISO8859-2	912	da_DK	ISO8859-1	819
Da_DK	IBM-850	850	de_CH	ISO8859-1	819
De_CH	IBM-850	850	de_DE	ISO8859-1	819

## Supported code sets

Table 11. Locales and CCSIDs for AIX (continued)

Locale	code set	CCSID	Locale	code set	CCSID
de_DE.IBM-1252	IBM-1252	1252	De_DE	IBM-850	850
el_GR	ISO8859-7	813	en_GB	ISO8859-1	819
En_GB	IBM-850	850	en_US	ISO8859-1	819
En_US	IBM-850	850	es_ES	ISO8859-1	819
es_ES.IBM-1252	IBM-1252	1252	Es_ES	IBM-850	850
Et_EE	IBM-922	922	fi_FI	ISO8859-1	819
fi_FI.IBM-1252	IBM-1252	1252	Fi_FI	IBM-850	850
fr_BE	ISO8859-1	819	fr_BE.IBM-1252	IBM-1252	1252
Fr_BE	IBM-850	850	fr_CA	ISO8859-1	819
Fr_CA	IBM-850	850	fr_CH	ISO8859-1	819
Fr_CH	IBM-850	850	fr_FR	ISO8859-1	819
fr_FR.IBM-1252	IBM-1252	1252	Fr_FR	IBM-850	850
hr_HR	ISO8859-2	912	hu_HU	ISO8859-2	912
is_IS	ISO8859-1	819	Is_IS	IBM-850	850
it_IT	ISO8859-1	819	it_IT.IBM-1252	IBM-1252	1252
It_IT	IBM-850	850	iw_IL	ISO8859-8	916
Iw_IL	IBM-856	856	ja_JP	IBM-eucJP	5050/33922
Ja_JP.IBM-932	IBM-932	932	Ja_JP.IBM-943	IBM-943	943
ko_KR	IBM-eucKR	970	Lt_LT	IBM-921	921
LT_LT	UTF-8	1208	Lv_LV	IBM-921	921
LV_LV	UTF-8	1208	mk_MK	ISO8859-5	915
nl_BE	ISO8859-1	819	nl_BE.IBM-1252	IBM-1252	1252
Nl_BE	IBM-850	850	nl_NL	ISO8859-1	819
nl_NL.IBM-1252	IBM-1252	1252	Nl_NL	IBM-850	850
no_NO	ISO8859-1	819	No_NO	IBM-850	850
pl_PL	ISO8859-2	912	POSIX	ISO8859-1	819
pt_BR	ISO8859-1	819	pt_PT	ISO8859-1	819
pt_PT.IBM-1252	IBM-1252	1252	Pt_PT	IBM-850	850
ro_RO	ISO8859-2	912	ru_RU	ISO8859-5	915
sh_SP	ISO8859-2	912	sk_SK	ISO8859-2	912
sl_SI	ISO8859-2	912	sq_AL	ISO8859-1	819
sr_SP	ISO8859-5	915	sv_SE	ISO8859-1	819

Table 11. Locales and CCSIDs for AIX (continued)

Locale	code set	CCSID	Locale	code set	CCSID
Sv_SE	IBM-850	850	th_TH	TIS-620	874
tr_TR	ISO8859-9	920	Uk_UA	IBM-1124	1124
Vi_VN	IBM-1129	1129	zh_CN	IBM-eucCN	1383
Zh_CN.GBK	GBK	1386	ZH_CN	UTF-8	1208
zh_TW	IBM-eucTW	964	Zh_TW	big5	950

## Migration to Euro support

To use the *euro* character with MQSeries, first install any operating system updates necessary to display the euro character.

Now modify your MQSeries system:

- Edit the existing CCSID.TBL file, in `/var/mqm/conv/table`, to enable the new euro version of the CCSID. To do this, remove the first # 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. For more information refer to the MQSeries Web site at:  
<http://www.ibm.com/software/mqseries/>

- To modify existing queue managers that do not support the euro character, follow this procedure:
  1. Enable MQSeries commands (MQSC) by typing `runmqsc`.
  2. Record the existing queue manager CCSID, with the MQSC command:  
`DISplay QMGR CCSID`
  3. Change the CCSID to the euro support CCSID, with the MQSC command:  
`ALTer QMGR CCSID(no. of ccsid)`
  4. Stop the MQSC commands by typing `END`.
  5. Stop the queue manager.
  6. Restart the queue manager and any channels it uses by typing `strmqm`.

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

## Euro support

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 C. 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

## Notices

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 Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

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.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

---

**Trademarks**

The following are trademarks of International Business Machines Corporation in the United States, or other countries, or both:

AIX	AS/400	BookManager
CICS	DB2	First Failure Support Technology (FFST)
IBM	IBMLink	MQSeries
OpenEdition	OS/2	OS/390
POWERparallel	POWERserver	RS/6000
SupportPac	TXSeries	VisualAge
VSE/ESA	WebSphere	

Lotus Notes is a trademark 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.

ActiveX, Microsoft, Visual Basic, Windows and Windows NT are registered 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, and service names may be trademarks or service marks of others.



---

# Index

## A

administration command sets  
  control commands 55  
  MQSeries commands  
  (MQSC) 57  
  programmable command format  
  commands (PCF) 58  
AIX at a glance 3  
amqsgetc sample program 38  
amqsputc sample program 37  
attributes  
  ALL attribute 62  
  changing 64  
  default 62

## B

bibliography 69  
BookManager 73  
books  
  ordering 69  
  printing 71  
books, translated 18  
browsing queues 65

## C

C and COBOL sample programs 77  
capabilities of MQSeries 52  
case-sensitive control commands 55  
CCSID (coded character set  
  identifier) 81  
  setting 25  
CD-ROM, installing from 16, 33  
changes to signal handling 14, 31  
changes to the installation path 14,  
  31  
changing queue attributes 64  
changing the national language 17,  
  33  
channel  
  events 54  
  message 50  
  MQI 36, 50  
clearing a local queue 64  
client channel 50  
client-connection channel,  
  example 37  
client installation 27  
  easy 32  
  with SMIT 33  
client-server configurations 51

client setup, example 36  
clients 51  
clusters 51  
code sets 81  
coded character set identifier  
  (CCSID) 81  
  setting 25  
command set administration 55  
commands  
  control 55  
  MQSC  
    ALTER QLOCAL 64  
    DEFINE QLOCAL 63  
    DEFINE QLOCAL LIKE 63  
    DEFINE QLOCAL  
      REPLACE 64  
    DELETE QLOCAL 65  
    using 58  
  programmable command format  
  (PCF) 58  
  runmqsc 61  
compilers 5, 28  
configurations 49  
connectivity 4  
control commands  
  case-sensitive 55  
  runmqsc 61  
controlled shutdown 59  
creating  
  file system for product code 11,  
  29  
  groups  
    client 28, 29  
    server 11  
  queue manager 58  
  users 11, 28, 29  
creating the system default  
  objects 9  
current queue depth  
  (CURDEPTH) 63

## D

databases 5  
DCE 5  
default  
  attributes of objects 62  
  queue manager commands  
  processed 61  
deleting  
  an MQSeries server or client 43

deleting (*continued*)  
  local queue 65  
  queue manager 38, 60  
disk requirements for installation 4  
  client 27  
dltmqm command 38

## E

earlier versions  
  migrating from version 5.0 12,  
  30  
easy client installation 32  
easy installation 15  
ending  
  interactive MQSC commands 61  
  queue manager 59  
endmqm command 38, 59  
environment variable  
  LANG 18  
  MQSERVER 37  
  NLSPATH 18  
error messages 61  
euro support, migrating to 83  
events 53  
  channel 54  
example  
  client-connection channel,  
  defining 37  
  client setup 36  
  getting the message from the  
  queue  
  on the MQSeries client 38  
inetd setup 36  
local queue, creating 35  
MQSC, starting 35  
MQSC, stopping 36  
putting a message on the queue  
  on the MQSeries client 37  
queue manager  
  creating 35  
  starting 35  
server-connection channel,  
  creating 36  
  verification, ending 38

## F

feedback from MQSC  
  commands 61  
file samples  
  CICS and Encina 79

file samples (*continued*)  
    miscellaneous 80  
    MQSC 77  
    Tuxedo 79  
file system, creating for product  
    code 11, 29

**G**  
groups, creating 11, 28, 29

**H**  
hard disk requirements 4  
hardware requirements  
    MQSeries for AIX client 27  
HTML books 70  
Hypertext Markup Language  
    (HTML) 72

**I**  
inetd setup 36  
information, ordering  
    publications 69  
installation  
    before 10  
    client 28  
    client example 35  
    components 6  
    directory, client 28  
    easy 15  
    filesets 6  
    from CD ROM 16, 33  
    planning 27  
    preparation 9, 28  
    procedure 32  
    reinstallation 40  
    remote 39  
    verification 18  
    with SMIT 16  
installation path, changes to 14, 31  
installation procedure 15  
installation verification  
    of local installation 19  
    of server to server 21  
    procedure 19  
installing  
    clients on the server 17  
installing a PTF (program temporary  
    fix) 40  
instrumentation events 53  
interactive MQSC  
    ending 61  
    feedback from 61  
    using 60  
Internet Gateway 67  
    removing 41  
introduction to MQSeries 47

**J**  
Java  
    support for MQSeries 18

**L**  
LANG environment variable 18  
LIKE attribute 63  
linking user exits 24  
local queue 49  
local queue manager 49  
local queues  
    clearing 64  
    copying definitions 63  
    defining one 62  
    deleting 65  
locale 81

**M**  
maintenance 39  
message  
    channels 50  
    description 48  
    descriptor 48  
message, translated 18  
message-driven processing 54  
message length, decreasing 64  
migrating from an earlier  
    version 12, 30  
migrating to euro support 83  
monitoring queue managers 53  
MQAI (MQSeries administration  
    interface) 54  
MQI channel 50  
MQSC commands  
    ALTER QLOCAL 64  
    DEFINE QLOCAL 63  
    DEFINE QLOCAL LIKE 63  
    DEFINE QLOCAL REPLACE 64  
    DELETE QLOCAL 65  
    ending interactive input 61  
    issuing interactively 60  
    using 58  
MQSeries for AIX  
    applying maintenance 39  
    client hardware requirements 27  
    client installation 27  
    client software requirements 27  
    components  
        client 32  
    components and filesets 6  
    restoring previous service  
        level 39  
MQSeries for AIX, V5.0, migrating  
    from 12, 30

MQSeries for AIX, V5.0 at a  
    glance 3  
MQSeries Web site 68  
MQSERVER environment  
    variable 37

**N**  
national language, changing 17, 33  
national language support 18  
NLSPATH environment variable 18

**O**  
objects  
    default attributes 62  
    working with 60  
objects, creating the system  
    defaults 9  
online books 70  
online help 73  
ordering books 69  
ordering publications 69

**P**  
PDF (Portable Document  
    Format) 71  
performance events 54  
Portable Document Format  
    (PDF) 71  
preemptive queue manager  
    shutdown 60  
printing books 71  
program samples 77  
    C and COBOL 77  
    databases 79  
programmable command format  
    (PCF)  
    administration with 58  
programming with MQSeries 54  
PTF (program temporary fix),  
    installing 40  
    publications 69

**Q**  
queue depth  
    current 63  
    determining 63  
queue manager  
    creating 58  
    definition 35  
    deleting 38, 60  
    description 49  
    events 54  
    immediate shutdown 59  
    monitoring 53  
    preemptive shutdown 59, 60  
    shutdown  
        controlled 59

- queue manager (*continued*)
  - immediate 59
  - preemptive 59
  - quiesced 59
  - starting 35, 59
  - stopping 38, 59

- queues
  - attributes 48
  - browsing 65
  - changing attributes 64
  - description 48
  - local
    - clearing 64
    - copying 63
    - defining 62
    - deleting 65
- quiesced shutdown 59

## R

- README file 8
- reinstalling 40
- remote installation 39
- remote queue 49
- remote queue manager 49
- removing
  - an MQSeries server or client 43
- removing the Internet Gateway 41
- requirements
  - disk storage 4
  - hardware 3
  - MQSeries for AIX, V5.2
    - software 4
- runmqsc
  - ending 61
  - feedback 61
  - using interactively 60

## S

- sample files
  - CICS and Encina 79
  - miscellaneous 80
  - MQSC 77
  - Tuxedo 79
- sample programs 77
  - C and COBOL 77
  - databases 79
- server-client configurations 51
- server-connection channel,
  - example 36
- setting the CCSID (coded character set identifier) 25
- setting up the server, example 35
- shell commands for MQSeries 55
- shutdown queue manager
  - controlled 59
  - immediate 59

- shutdown queue manager (*continued*)
  - preemptive 59
  - quiesced 59
- signal handling, changes to 14, 31
- softcopy information 70
- software requirements
  - MQSeries for AIX, V5.2 4
  - MQSeries for AIX client 27
- starting a queue manager 59
- stopping a queue manager 38, 59
- strmqm command 59
- supported code sets 81
- syntax error, in MQSC
  - commands 61
- System Management Interface Tool (SMIT)
  - installing MQSeries client
    - with 33
  - installing MQSeries with 16

## T

- transaction monitors 5
- transactional support 52
- translated books 18
- translated messages 34
  - server 18
- triggering 54

## U

- uninstalling
  - an MQSeries server or client 43
- user exits, linking 24
- user ID, creating
  - on client 29
  - on server 11
- users, creating 11, 28, 29

## V

- verification
  - communication between workstations 24
  - of local installation 19
  - of server to server
    - installation 21
    - procedure 19
    - receiver workstation 22
- verification, ending 38
- verify installation 18, 34

## W

- WebSphere 5
- World Wide Web interface 67

## Y

- Year 2000 compatibility 3





---

## Sending your comments to IBM

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 make comments about the functions of IBM products or systems, 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, to this address:  
User Technologies Department (MP095)  
IBM United Kingdom Laboratories  
Hursley Park  
WINCHESTER,  
Hampshire  
SO21 2JN  
United Kingdom
- 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 method you use, ensure that you include:

- The publication title and order number
- The topic to which your comment applies
- Your name and address/telephone number/fax number/network ID.







Printed in the United States of America  
on recycled paper containing 10%  
recovered post-consumer fiber.

GC33-1867-03



Spine information:



MQSeries<sup>®</sup> for AIX<sup>®</sup>

MQSeries for AIX V5.2 Quick Beginnings

Version 5.2