



MQSeries for Sun Solaris

Quick Beginnings

Version 5.0



MQSeries for Sun Solaris

Quick Beginnings

Version 5.0

Note

Before using this document, read the general information under Appendix F, "Notices" on page 121.

Second edition (March 1998)

This edition applies to Version 5.0 of *MQSeries for Sun Solaris* and to all subsequent releases and modifications until otherwise indicated in new editions or technical newsletters.

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

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

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

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

© **Copyright International Business Machines Corporation 1995,1998. All rights reserved.**

Note to U.S. Government Users — Documentation related to restricted rights — Use, duplication or disclosure is subject to restrictions set forth in GSA ADP Schedule Contract with IBM Corp.

Contents

Figures	v
Tables	vi
Welcome to MQSeries for Sun Solaris	vii
Road Map	vii
Conventions	viii
What's New	ix

Part 1. The IBM Software Servers 1

Chapter 1. Introducing IBM's Software Servers	3
IBM Communications Server	4
IBM Directory and Security Server	4
IBM Transaction Series	5
IBM MQSeries	5
Tivoli Management Servers	6
IBM Internet Connection Server	6
IBM DB2 Universal Database Server	7
Lotus Domino	7
Combining MQSeries for Sun Solaris with Other Software Servers	8
Chapter 2. More about MQSeries for Sun Solaris	9
Message Queuing	9
Messages and Queues	10
MQSeries Objects	11
Clients and Servers	13
Linking to Lotus Notes	14
Instrumentation Events	15
Transactional Support	16

Part 2. Planning for and Installing MQSeries for Sun Solaris 17

Chapter 3. Planning to Install the MQSeries for Sun Solaris Server	19
Hardware Requirements	19
Software Requirements	20
Connectivity	20
Compilers supported for Solaris applications	21
Options	21
Delivery	22
Installation	22

Contents

MQSeries for Sun Solaris Components	22
README file	25
Chapter 4. Installing the MQSeries for Sun Solaris Server	27
Preparing for installation	27
Installing the MQSeries for Sun Solaris Server	29
Kernel configuration	30
Installing the Server and Client on the Same Machine	32
Translated messages	32
Verifying the Installation of MQSeries for Sun Solaris	32
User Exits	38
Setting the Queue Manager CCSID on MQSeries for Sun Solaris	38
Chapter 5. Planning to Install MQSeries Clients	41
Applications on Version 5 clients	41
Migrating from MQSeries Version 2 products to Version 5	42
AIX Client: Hardware and Software Required	42
Components for AIX	44
DOS Client: Hardware and Software Required	45
Components for DOS	46
HP-UX Client: Hardware and Software Required	47
Components for HP-UX	48
OS/2 Warp Client: Hardware and Software Required	49
Components for OS/2 Warp	50
Sun Solaris Client: Hardware and Software Required	52
Components for Sun Solaris	53
Windows 3.1 Client: Hardware and Software Required	54
Components for Windows 3.1	55
Windows 95 Client: Hardware and Software Required	56
Components for Windows 95	57
Windows NT Client: Hardware and Software Required	58
Components for Windows NT	59
Chapter 6. Installing MQSeries Clients	61
Installing on AIX	61
Installing on DOS	63
Installing on HP-UX	64
Installing on OS/2 Warp	66
Installing on Sun Solaris	74
Installing on Windows NT or Windows 95	76
Installing on Windows 3.1	80
Configuring and Verifying a Client	82
<hr/>	
Part 3. Using MQSeries for Sun Solaris	83
Chapter 7. Using the MQSeries Command Sets	85
Introducing Command Sets	85
Working with Queue Managers	88

Figures

Working with MQSeries Objects	91
Chapter 8. Using the MQSeries World Wide Web Interface	97
Overview of MQSeries Internet Gateway	97
Obtaining More Information	97
Obtaining MQSeries Internet Gateway	98
Chapter 9. Obtaining Additional Information	99
The MQSeries for Sun Solaris Library	99
Viewing Online Information	101
Printing and Ordering Books	102
Available from the Internet	103
<hr/>	
Part 4. Appendixes	105
Appendix A. Installing the MQSeries Product License	107
Enrolling the Trial-Period License	107
Enrolling the Product License	107
Appendix B. Migrating from an Earlier Version of MQSeries	109
Appendix C. Sample MQI Programs and MQSC Files	111
Appendix D. Applying maintenance to MQSeries for Sun Solaris	115
Space Requirements	115
Appendix E. Support for different code sets on MQSeries for Sun Solaris	117
Appendix F. Notices	121
Trademarks	122
Index	123

Figures

1. Kernel parameter values - example setting on a Solaris 2.5.1 system	31
--	----

Tables

Tables

1.	Getting Started Road Map	vii
2.	Summary of Changes March 1998	ix
3.	MQSeries publications – file names	103
4.	MQSC command files	111
5.	Sample programs - source files	111
6.	Samples for transaction processing with CICS and Encina	113
7.	Samples for transaction processing with Tuxedo	113
8.	Samples for Lotus Notes	114
9.	Sample programs - databases	114
10.	Miscellaneous files	114
11.	Locales and CCSIDs	117

Welcome to MQSeries for Sun Solaris

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

Part 1 reviews the IBM software servers and includes an overview of *MQSeries for Sun Solaris*.

Part 2 provides planning and installing information for *MQSeries for Sun Solaris*. Information is provided for both server and client components.

Part 3 explains how to use *MQSeries for Sun Solaris*.

Road Map

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

<i>Table 1 (Page 1 of 2). Getting Started Road Map</i>	
If you want to...	Refer to...
Find out about IBM's software servers	Chapter 1, "Introducing IBM's Software Servers" on page 3
Read about <i>MQSeries for Sun Solaris</i> .	Chapter 2, "More about MQSeries for Sun Solaris" on page 9
Learn about system requirements for MQSeries for Sun Solaris	Chapter 3, "Planning to Install the MQSeries for Sun Solaris Server" on page 19
Install MQSeries for Sun Solaris	Chapter 4, "Installing the MQSeries for Sun Solaris Server" on page 27
Learn about system requirements for clients	Chapter 5, "Planning to Install MQSeries Clients" on page 41
Install an MQSeries client	Chapter 6, "Installing MQSeries Clients" on page 61
Start using command sets	Chapter 7, "Using the MQSeries Command Sets" on page 85
Start using the Web Interface	Chapter 8, "Using the MQSeries World Wide Web Interface" on page 97
View or print online documentation	Chapter 9, "Obtaining Additional Information" on page 99

Conventions

If you want to...	Refer to...
Contact IBM	<i>Readers' Comment Form</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.
- Monospace type indicates an example (such as a fictitious path or file name) or text that is displayed on the screen.

What's New

Table 2 describes the changes made in this edition.

Page	Change
24	Addition of section on creating system default objects
38	Addition of a section on setting the queue manager coded character set identifier (CCSID)
116	Addition of a section on restoring a previous service level of MQSeries

This edition incorporates some information included in the README file shipped with MQSeries for Sun Solaris Version 5.0.

What's New

Part 1. The IBM Software Servers

This part of the book introduces IBM software servers and explains the benefits of using them. It includes the following chapters:

- Chapter 1, "Introducing IBM's Software Servers" on page 3
- Chapter 2, "More about MQSeries for Sun Solaris" on page 9

Chapter 1. Introducing IBM's Software Servers

When you choose the newest version of *MQSeries for Sun Solaris*, you will be putting your business on the fast path toward client/server and network computing.

One of the best network enhancements you can make, this exciting new server and client software is part of a family of IBM, Lotus, and Tivoli products. They comprise middleware plus administration and end-user tools. These software products run in hardware servers and are designed to support state-of-the-art applications, so we refer to them as *application servers*.

IBM Software Servers are packaged to help you to install quickly and make use of a family of multifunctional, compatible application servers — designed to work in harmony. With modular IBM Software Servers, you can build a high-function, reliable environment for decision support, messaging and groupware, transaction processing, and other client/server functions vital to today's business. And with new Web functions, you can exploit new technologies and move into the world of network computing.

This family of products provides:

- New Web functions and improved integration, now available in Version 5.0
- Popular software chosen by application providers to run their applications
- Greatest depth and breadth of application servers — providing unequalled depth in each server, as well as the widest breadth of application servers — making it easy to grow from the smallest departmental servers to the largest systems of a modern enterprise
- Open solutions available on multiple platforms — popular clients (for example Windows, Windows 95, Windows NT, Macintosh, OS/2, UNIX, and popular browsers used on the Internet or in intranets) and servers (for example OS/2, Windows NT, and AIX) — and in conformance with all key industry-standard interfaces
- Support for existing investments, since IBM Software Servers work with your existing systems, are thoroughly tested, and work together
- Growth paths for future needs, allowing you to add new servers when needed, move applications to more powerful platforms as demand grows, and extend your existing systems with new technologies in the move to collaborative and network computing
- Ease of installation and simple administration, with common packaging and documentation

The IBM Software Servers family includes several servers, designed to meet your needs.

IBM Communications Server

IBM Communications Server will meet your requirements for interconnecting diverse networks and prove to be a winner for your business. With Communications Server, workstation users and applications can communicate with other workstations and central computer applications, regardless of the networking protocols used in each system. Communications Server has full-function gateways to clients on SNA and TCP/IP networks, plus support of a broad range of industry-standard networking protocols. Communications Server will help you to get users communicating with each other in networks of all sizes, from small workgroups to large corporate headquarters.

Communications Server provides businesses with a variety of opportunities to:

- Expand the use of applications, while protecting current network investments
- Reduce operation and management costs by connecting networks without impacting existing applications
- Increase productivity and convenience by giving employees access to applications while at work, home, or when traveling
- Reduce costs of central computer and peer-to-peer connectivity by sharing communication resources
- Gain efficiency in response times by assigning priority to short, interactive data transmissions rather than to batch-oriented bulk data traffic

IBM Directory and Security Server

With IBM Directory and Security Server, you can simplify the effort involved in having increased control over complex networks, multivendor systems, and multiple platforms, while maintaining security within your network. With Directory and Security Server, you can generate and manage secure, distributed applications that are interoperable and portable. Directory services make it easy to combine multiple directories, even across WANs, into a simple, single view. Plus, it can be done without deleting users and resources with the same names. You get a single integrated view of all available resources, including printers, files, and applications — no matter where they are located.

Directory and Security Server gives systems administrators a single logical view of all LAN and WAN resources — regardless of installed platforms and vendors. And users get convenient single sign-on, providing transparent access to any authorized enterprise resource, which can include the IBM Software Servers.

MQSeries

The security services are based on the Kerberos security model, which uses a third-party authentication server. This server authenticates both the client and server, making impersonation virtually impossible. Another security measure allows only authentication tickets to flow over the network, to protect passwords. Data encryption standard (DES) and commercial data masking facility (CDMF) data encryption technologies are also available for total data privacy.

IBM Transaction Series

IBM Transaction Series lets you make flexible and strategic use of new technology, while continuing to meet your business needs using established technology. Transaction Series coordinates and integrates the functions of multiple application servers, locally or throughout your enterprise. This helps you to produce and manage very high-performance applications. It also enables organizations to get and use the operational data needed on customer transactions and stay a jump ahead of the competition.

Transaction Series provides the ideal environment for applications to run anywhere, any time, no matter where they reside in the network. No matter how demanding your applications and requirements are, Transaction Series gives you the safety, security, and integrity you need for your business.

Transaction Series offers two flexible programming styles. The CICS API is available for those who want to leverage existing CICS applications. Encina supports those who want to exploit an open, standards-based API.

IBM MQSeries

In an open-systems environment, communications coding is complex and difficult to manage. IBM MQSeries simplifies things, giving you time to develop the solutions that deliver your business mission. IBM MQSeries messaging software enables business applications to exchange information across different operating system platforms in a way that is straightforward and easy for programmers to implement.

IBM's MQSeries simplifies the task of connecting your applications across unlike environments. Programs communicate using the IBM MQSeries API, an easy-to-use, high-level program interface that shields your programmers from the complexities of different operating systems and underlying networks. You focus on the business logic, while IBM MQSeries manages your connections to the computer systems.

Internet Connection Server

IBM MQSeries also provides an immensely flexible infrastructure, on which you can build mobile communications, multimedia applications, transaction-based systems, and a whole new generation of network applications.

Tivoli Management Servers

Controlling distributed computing resources can be difficult, expensive, and time consuming. Companies need to be business-driven, not technology-driven.

With Tivoli Management Servers, you can take control of your distributed computing environment and concentrate on what is important — the business that makes you money. These servers bring state-of-the-art systems management to your desktop.

Tivoli Management Servers provide total, open, flexible systems management. Some functions include:

- Host connectivity
- Help desk utilities
- Asset management
- Virus protection
- Software distribution
- License metering
- System monitoring
- User administration
- Performance management
- Job scheduling

These are just a few! With Tivoli Management Servers, you get the best of IBM SystemView technology and Tivoli's TME10 technology.

IBM Internet Connection Server

If you have browsed the Internet, you know what a valuable resource it can be. If you want to market your products and services around the world, IBM has the server software to get you started — IBM Internet Connection Server.

With Internet Connection Server, you can host and manage HyperText Markup Language (HTML) pages and other online documents, and create your own corporate Web site. From its corporate home page, your company can:

Lotus Domino

- Distribute up-to-date information to the world using text, high-quality graphics, audio, and video
- Create information that customers and suppliers can access through electronic forms or e-mail
- Publish product descriptions and price lists with electronic order forms, so customers can purchase your products and services with a credit card — right from their own computers
- Track how customers, suppliers, and personnel use your published information, so you can tell when you are reaching your target audience
- Gain a competitive edge by creating a worldwide presence for your company

IBM DB2 Universal Database Server

IBM DB2 Universal Database Server gives you the foundation for reliable data management, offering complete integrity, high availability, and fast performance. DB2 Universal Database Server makes it easy to create multimedia applications to present your data through video, audio, and image techniques. Local and remote users can store, retrieve, manipulate, manage, and share data — easily and reliably, using the newest and best-tested technology in the relational database industry.

With the IBM DB2 family of products, you can access data on multiple platforms. Access to IBM and non-IBM relational and nonrelational data is available through optional components of the DB2 family.

Lotus Domino

Lotus Domino transforms Lotus Notes into an interactive Web application server, allowing any Web client to participate in Notes applications securely. Domino bridges the open networking environment of Internet standards and protocols with the powerful application development facilities of Notes. You get a solution that combines messaging, groupware, and the Internet — all built-in.

With Domino, you can rapidly develop a broad range of business applications for the Internet and intranet. Plus, you get support for many popular operating systems for both clients and servers, as well as Web browsers. Domino can be your central access point for finding and sharing the information you need — whether it is centrally located in an e-mail message, a relational database, a host-based system, in your favorite desktop application, or on the World Wide Web. How? You can combine with Lotus

Combining Software Servers

Domino any of the other Software Servers, plus other vendors' application servers and applications.

Combining MQSeries for Sun Solaris with Other Software Servers

When you combine MQSeries for Sun Solaris with other software in this family, you get an integrated software environment that will support your applications today and lay the foundation for the future. Other software suppliers have expressed interest in or have already built integrated solutions using members of this IBM software family.

This document describes MQSeries for Sun Solaris, which provides flexible, reliable communications for your network. Take advantage of your Sun SPARC workstations as you access existing applications on S/390 and AS/400 hosts. Plus, an extensive set of application programming interfaces makes it ideal for client/server computing.

MQSeries for Sun Solaris can be installed with Sun Solaris Version 2.5.1, or later Version 2.X. (See "Software Requirements" on page 20 for a complete list of software requirements.)

To learn how to install MQSeries for Sun Solaris, refer to "Installing the MQSeries for Sun Solaris Server" on page 29.

Chapter 2. More about MQSeries for Sun Solaris

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

- “Message Queuing”
- “Messages and Queues” on page 10
- “MQSeries Objects” on page 11
- “Clients and Servers” on page 13
- “Linking to Lotus Notes” on page 14
- “Instrumentation Events” on page 15
- “Transactional Support” on page 16

For more detailed explanations of these topics, refer to the *MQSeries Planning Guide*.

Message Queuing

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

MQI – a Common Application Programming Interface

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

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

Messages and Queues

Time-Independent Applications

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

Message-Driven Processing

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

Messages and Queues

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

What is a Message?

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

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

What is a Queue?

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

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

MQSeries Objects

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

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

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

MQSeries Objects

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

- Queue managers
- Queues
- Process definitions
- Channels

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

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

Queue Managers

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

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

MQSeries Objects

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

Queues

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

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

Process Definitions

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

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

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

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

Channels

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

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

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

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

For more information on channels and how to use them, see 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 MQI calls from applications and pass them to an *MQI server* machine. There they are processed by a queue manager. Typically, the client and server reside on different machines, but they can also exist on the same machine.

An *MQI server* is a queue manager that provides queuing services to one or more clients. All the MQSeries objects, for example queues, exist only on the queue manager machine, that is, on the MQI server machine. A server can support local MQSeries applications as well.

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

Linking to Lotus Notes

Linking to Lotus Notes

MQSeries provides a Lotus Notes server add-in task that gives Lotus Notes applications access to MQSeries messaging. This allows Lotus Notes users to communicate with other systems connected by MQSeries.

Lotus Notes is a networked application that users can use to share information. Lotus Notes has two main components; the server and the client. The Lotus Notes server provides services to Lotus Notes clients and to other servers. The services provided include storage and replication of shared databases and mail routing. Lotus Notes clients connect to a Lotus Notes server to use shared databases, and also to read and send mail.

The Lotus Notes server add-in task, provided by MQSeries, recognizes and interprets:

- Data from documents that Lotus Notes wants to send to MQSeries
- Messages from MQSeries sent in reply and used to update a Lotus Notes document

A Lotus Notes application consists of a database containing specially constructed documents. These documents contain formulas (or macros) that can be executed by the user. A formula that makes a link to MQSeries transfers parts of the document to a Mail-In database associated with the server add-in task.

The Lotus Notes server add-in task monitors the Mail-In database. Documents found there are used to construct MQSeries messages. The link database contains entries describing the relationship of a Lotus Notes document to an MQSeries message; that is, how a Lotus Notes document is mapped into an MQSeries message. The link database must be set up to define the mapping required for each type of document you want to use with MQSeries.

The Lotus Notes server add-in task can also monitor response queues based on any outstanding replies. The replies are interpreted and used to update the user document in the user database.

To provide access to MQSeries from Lotus Notes, Lotus Domino.Connect Release 2 includes MQSeries and CICS Connections for Domino 1.0, consisting of MQSeries Enterprise Integrator (MQEI) and MQSeries link LotusScript Extension (MQLSX). Both MQLSX and MQEI provide access to MQSeries from LotusScript. See your IBM or Lotus representative for more details.

The Lotus Notes add-in task (MQLink) is provided for migration purposes of MQLink applications from earlier versions of Lotus Notes only. For current Lotus Notes support, you should upgrade to MQLSX, which includes an MQLSX Link sample application written in LotusScript, that provides similar function to the add-in task, using the same link database.

Instrumentation Events

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

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

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

Other conditions that can give rise to instrumentation events include:

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

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

Types of Event

MQSeries events are categorized as follows:

Queue manager events

These events are related to the definitions of resources within queue managers. For example, if an application attempts to open a queue but the associated user ID is not authorized to perform that operation, a queue manager event is generated.

Performance events

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

Channel events

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

Transactional Support

Transactional Support

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

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

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

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

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

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

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

Part 2. Planning for and Installing MQSeries for Sun Solaris

This part of the book gives you advice on planning and preparing for the installation of MQSeries for Sun Solaris. It also gives you step-by-step guidance on the installation process for MQSeries for Sun Solaris and for MQSeries clients. It contains the following chapters:

- Chapter 3, "Planning to Install the MQSeries for Sun Solaris Server" on page 19
- Chapter 4, "Installing the MQSeries for Sun Solaris Server" on page 27
- Chapter 5, "Planning to Install MQSeries Clients" on page 41
- Chapter 6, "Installing MQSeries Clients" on page 61

Chapter 3. Planning to Install the MQSeries for Sun Solaris Server

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

Hardware Requirements

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

Disk storage

The installation requirements depend on which components you install and how much working space you need. This, in turn, depends on the number of queues that you use, the number and size of the messages on the queues, and whether the messages are persistent or not. You also require archiving capacity on disk, tape, or other media.

These are the approximate storage requirements:

- Server:
 - A minimum of 25 MB of disk space must be available for the product code and data in the filesystem containing the /opt directory.
 - In addition, if you install the online books in HTML format you require 35 MB of storage for the books in the /opt directory.
 - After installation the books are placed in the /opt/mqm/html directory.

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

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

Connectivity

Software Requirements

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

- Sun Solaris Version 2.5.1 or any later Version 2.x (includes TCP/IP)

If you install MQSeries Client for Java and MQSeries Bindings for Java, you need the following patches in addition to Sun Solaris Version 2.5.1:

103566-08
103600-13
103640-08

These components also require the following, or later compatible versions of Java run-time code to be installed on your machine:

MQSeries Client for Java – Java Version 1.1.1
MQSeries Bindings for Java – Java Version 1.1.3 (Native threads), which is available from java.sun.com/products/jdk/1.1.

Connectivity

Network protocols supported are SNA LU 6.2 and TCP/IP.

- Any communications hardware supporting SNA LU 6.2 and/or TCP/IP

For SNA connectivity:

- SunLink SNA Peer-to-Peer Version 9.0 or later Version 9.X. Requires patches 102713, 102690, 102312, 102874.

If token ring is used, SunLink Token Ring Interface/ SBus 3.0.2 Requires patch 102463.

- TCP/IP is part of the base operating system.

Compilers supported for Solaris applications

- C using SunSPARC compiler R4.0 and R4.2
- C++ using SunSPARC compiler C++ R4.1 or R4.2
On Version 4.0, MQ should be built with C 4.0 or C++ 4.1
- Micro Focus COBOL for UNIX version 3.2

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

Options

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

Transaction Monitors

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

- IBM CICS for Solaris Version 2.1.1 (84H1895)
- BEA Tuxedo Version 5.1 or 6.1
- Transarc Encina V2.5

Databases

- IBM DB2 for Solaris V2.1.1
- Oracle, Release 7.3.2.3

IBM Software Servers

- Internet Connection Server for Sun Solaris V4.1 (84H1611)
- Internet Connection Security Server for Solaris V4.1 (84H1616)

MQSeries for Sun Solaris Components

DCE

Transarc DCE V1.1 and later.

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

DCE names and security modules are provided with MQSeries for Sun Solaris V5.0.

Lotus Notes

- Lotus Notes V4.1 and upwardly compatible versions

Delivery

MQSeries for Sun Solaris V5.0 is supplied on CD-ROM.

Installation

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

MQSeries for Sun Solaris Components

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

Runtime component

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

Base

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

MQSeries for Sun Solaris Components

Server

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

MQSeries Clients

MQSeries Client for Sun Solaris and MQSeries client for Java:

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

Notes:

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

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

If you want to run your old applets, remove the MQSeries Client for Java subdirectory (/opt/mqm/java/lib) and restart your web browser.

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

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

DCE support

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

Note to users

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

DCE samples

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

mqlink

Link to Lotus Notes

MQSeries Online Documentation

Online versions of the books for MQSeries for Sun Solaris in HTML formats. Included are:

- MQSeries Documentation
- MQSeries Internet Documentation

MQSeries for Sun Solaris Components

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

MQSeries Internet Gateway

Provides access to MQSeries applications through HTML and CGI.

Man

Man pages for the following commands:

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

Samples

Sample application programs.

MQSeries code

The MQSeries code in the following National Languages:

- US English
- Brazilian Portuguese
- French
- German
- Italian
- Japanese
- Korean
- Spanish
- Simplified Chinese
- Traditional Chinese

Note: The “base” product is automatically installed.

Creating the System Default Objects

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

If you used `amqscoma.tst` to customize your settings for MQSeries Version 2, and you want to use the same settings with Version 5.0 of the product:

1. Save your copy of `amqscoma.tst`
2. Install MQSeries Version 5.0
3. Load your copy of `amqscoma.tst` and use the file to recreate your default objects.

README file

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

README

Chapter 4. Installing the MQSeries for Sun Solaris Server

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

The MQSeries product is installed into the `/opt/mqm` directory. This *cannot* be changed. However, if you do not have enough space in the `/opt/mqm` file system, follow the procedure given in “Creating another file system for product code” on page 28.

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

Preparing for installation

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

Before installation

Before you can install MQSeries for Sun Solaris you:

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

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

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

Preparing for installation

Notes:

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

You can choose to NFS mount the `/var/mqm/errors` and `/var/mqm/trace` directories to conserve space on your local system.
3. The size of the log file depends upon the log settings that you use. The size recommended is for circular logging using the default settings. For further information on log sizes see the *MQSeries System Administration* book.

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

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

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

Creating another file system for product code

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

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

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


Notes:

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

Installing the MQSeries for Sun Solaris Server

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

Notes:

1. These installation instructions assume that you have not used the product as part of a 90-day trial period. If you are installing the product for the trial period, or have previously used the trial period, see Appendix A, “Installing the MQSeries Product License” on page 107. This tells you how to enroll the product license in these situations.
2. If you have previously installed MQSeries on your system, you need to remove the product using the **pkgrm** program.
3. If the product is present, but not installed correctly, you may need to manually delete the files and directories contained in:

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

Carry out the following procedure:

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

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

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

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

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

2. Use the Solaris **pkgadd** program, to install the software by carrying out the following procedure:
 - a. Type `pkgadd -d /cdrom/mq_solaris/mqm/mqs500.img`.

Kernel configuration

- b. You are prompted for a list of components to install. Select the ones you require – if you want to install the entire MQSeries product, select **all**.
- c. Press the Enter key.

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

Kernel configuration

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

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

If you attempt to use MQSeries without increasing `semnmi`, the number of semaphores, the queue manager fails and produces a First Failure Support Technology (FFST) file. This file indicates that the system call `semop` received an argument that was not valid. An example of a possible set of actual kernel values is given in Figure 1 on page 31. However, for Sun Solaris Version 2.5.1, the number of semaphores and semaphore sets has to be such that their control structures occupy less than 25% of the kernel storage.

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

```
sysdef -i
```

To change the values, add a `set parameter = value` line to the `/etc/system` file. For further information on setting up the system, see the *Sun Solaris Version 2.5 System Administration Guide*.

Kernel configuration

```
set shmsys:shminfo_shmmax = 4194304
set shmsys:shminfo_shmseg = 1024
set shmsys:shminfo_shmmni = 1024
set shmsys:shminfo_shmem = 1
set semsys:seminfo_sema = 1
set semsys:seminfo_semaem = 16384
set semsys:seminfo_semvmx = 32767
set semsys:seminfo_semmni = 1024 (semmni < semmns)
set semsys:seminfo_semmap = 1026 (semmni +2)
set semsys:seminfo_semmns = 16384
set semsys:seminfo_semmnl = 100
set semsys:seminfo_semopm = 100
set semsys:seminfo_semmnu = 2048
set semsys:seminfo_semume = 256
set msgsys:msginfo_msgmni = 50
set msgsys:msginfo_msgmap = 1026
set msgsys:msginfo_msgmax = 4096
set msgsys:msginfo_msgmnb = 4096
set msgsys:msginfo_msgssz = 8
set msgsys:msginfo_msgtql = 40
set msgsys:msginfo_msgseg = 1024
set maxusers = 32
```

Figure 1. Kernel parameter values - example setting on a Solaris 2.5.1 system

Notes:

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

For more details see the relevant SupportPac, which is available on the MQSeries family home page. See "Available from the Internet" on page 103 for further details.

Multiple Threading

If you have an MQSeries Client for Java, or an MQSeries Bindings for Java, application that is going to make heavy use of multiple threading, you may need to set the THREADS_FLAG environment variable to **native**.

Failure to do so can result in premature failure of your application. For more information, see the README file supplied with the "Native Threads Pack", which is available from java.sun.com/products/jdk/1.1.

Once you have made the changes, you must restart the system.

Verifying the Installation

Installing the Server and Client on the Same Machine

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

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

Translated messages

Messages in U.S. English are always available. If you require another of the languages that is supported by MQSeries for Sun Solaris Version 5, you *must* ensure that your NLSPATH environment variable includes the appropriate directory.

For example:

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

Verifying the Installation of MQSeries for Sun Solaris

This section describes how to verify that MQSeries for Sun Solaris has been correctly installed and configured. You do this by following the steps outlined in "Verification Procedure" on page 33

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

Verifying the Installation

the required communications protocols have been installed (and configured) on *both* machines.

The supported protocols are TCP/IP and SNA.

Note: The following examples assume that you will be using a TCP/IP connection; for information about using other protocols, see the *MQSeries Intercommunication* manual.

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 35 . For information on verifying a client/server installation, see the *MQSeries Clients* manual.

Verifying a Local Installation

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

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

```
crtmqm -q venus.queue.manager
```
 - Press Enter.
 - Messages are displayed telling you that the queue manager has been created, and that the default MQSeries objects have been created.

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

Verifying the Installation

- Type the following and then press Enter:

```
strmqm
```

- A message tells you when the queue manager has started.

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

```
runmqsc
```

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

MQSC has no command prompt.

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

- Type the following and press Enter:

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

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

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

You have now defined:

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

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

The following message is displayed:

```
1 MQSC commands read.  
0 commands have a syntax error.  
0 commands cannot be processed.
```

7. The command prompt is now displayed again.

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

1. Change into the following directory :

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

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

```
amqsput ORANGE.QUEUE
```

3. The following message is displayed:

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

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

Verifying the Installation

5. The following message is displayed:

```
Sample amqspu0 end
```

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

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

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

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

```
amqsget ORANGE.QUEUE
```

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

The verification is complete.

Verifying a Server-to-Server Installation

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

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

Sender Workstation:

1. Create a default queue manager called *saturn.queue.manager*.

- At a command prompt in a window, type:

```
crtmqm -q saturn.queue.manager
```
- Press Enter.
- Messages are displayed telling you that the queue manager has been created, and that the default MQSeries objects have been created.

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

2. Start the queue manager:

- Type the following and then press Enter:

```
strmqm
```
- A message tells you when the queue manager has started.

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

```
runmqsc
```

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

MQSC has no command prompt.

Verifying the Installation

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

- Type the following and press Enter:

```
define qlocal (transmit1.queue) usage (xmitq)
```
- The message MQSeries queue created is displayed when the queue has been created.

5. Create a local definition of the remote queue:

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

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

6. Define a sender channel:

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

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

You have now defined the following objects:

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

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

Now set up the receiver workstation.

Receiver Workstation

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

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

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

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

```
MQSeries stream tcp nowait mqm /opt/mqm/bin/amqcrsta amqcrsta  
[-m venus.queue.manager]
```

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

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

4. Run the command:

Verifying the Installation

```
kill -1 inetd processid
```

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

- At the command prompt, type:

```
crtmqm -q venus.queue.manager
```
- Press Enter.
- Messages are displayed telling you that the queue manager has been created, and that the default MQSeries objects have been created.

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

6. Start the queue manager:

- Type the following and then press Enter:

```
strmqm
```
- A message tells you when the queue manager has started.

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

```
runmqsc
```

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

MQSC has no command prompt.

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

- Type the following and press Enter:

```
define qlocal (orange.queue)
```
- The message MQSeries queue created is displayed when the queue has been created.

9. Create a receiver channel:

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

You have now defined the following objects:

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

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

Establishing communication between the Workstations:

1. If the queue managers on the two workstations have been stopped for any reason, restart them now (using the *strmqm* command).
2. On the **Sender** workstation start the sender channel:

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

Queue Manager CCSID

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
```

6. The message is displayed.

The verification is complete.

User Exits

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

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

Setting the Queue Manager CCSID on MQSeries for Sun Solaris

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

Note: The queue manager CCSID **cannot** be changed once the queue manager has been created.

To change the CCSID, you must:

- Delete the queue manager
- Change the locale

Queue Manager CCSID

- Recreate the queue manager

Examples of setting the CCSID:

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

You can display the queue manager CCSID using the DIS QMGR command from within **runmqsc**.

See Appendix E, "Support for different code sets on MQSeries for Sun Solaris" on page 117 for further information about supported code sets.

Queue Manager CCSID

Chapter 5. Planning to Install MQSeries Clients

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

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

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

Applications on Version 5 clients

The MQSeries Version 5 products include new features, for example distribution lists, handling large messages and using external resource managers. All the new features are described in the *MQSeries Application Programming Reference*. If you install an MQSeries client supplied with a Version 5 product you can use the new features and structures in your MQSeries application on the client when connected to a Version 5 queue manager on your server.

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

Migrating from MQSeries Version 2 products to Version 5

The internal format of the channel definition table has been changed in the MQSeries Version 5 products for AIX, HP-UX, OS/2 Warp, Sun Solaris and Windows NT.

A Version 5 client will continue to work with a Version 2 client channel definition table (by default, the client channel definition table is called AMQCLCHL.TAB). However, a Version 2 client cannot read a Version 5 client channel definition table. If your client channel definition table is produced by a Version 5 server, you must use Version 5 Clients. In this case, reinstall the MQSeries client code from an MQSeries Version 5 server. In the case of a DOS MQSeries client, also relink your application.

AIX Client: Hardware and Software Required

Hardware

An MQSeries client can run on an IBM RISC System/6000, capable of running AIX V4.1.4. Any other trademarked AIX system may be used, whether from IBM or other vendors such as Bull, Zenith, or Motorola. There must be enough random access memory (RAM) and disk storage for the programming prerequisites (below), the MQSeries client code, the access methods, and the application programs.

Programming Requirements

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

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

Connectivity

For TCP/IP connectivity:

- TCP/IP (in the operating system)

For SNA connectivity:

Hardware and Software, AIX

- IBM Communications Server for AIX, V4.0 (5765-652)

Workstation MQSeries clients

MQSeries client code for AIX workstations is distributed with the server code for all MQSeries Version 5 products. Later levels of some listed products may be required for AIX V4.2 and/or SP. Later levels of the operating system may be required to support corequisite products. The MQSeries client code operates under:

- AIX V4.1.4 (5765-393 or 5765-C34) or AIX Version 4.2 (5765-655 or 5765-C34) or later.

Note: For V4.1, PTF U449790 is required if user data conversion of Greek, Cyrillic, Eastern European, Turkish, Japanese, or Korean language text longer than 2000 bytes is required.

For V4.2 level 4.2.1 should be used for the same languages.

Options, not prerequisites

- IBM Directory and Security Server for AIX (5765-639), V4 and later compatible versions. This must be the U.S. Domestic version supporting DES encryption if the user wishes to run the MQSeries-supplied DCE send, receive, or message exits.

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

Compilers for MQSeries Applications on AIX Clients

The following C compilers are supported:

- IBM C for AIX V3.1.4 (5765-423)
- IBM C Set++ for AIX V3.1 (5765-421)

The following COBOL compilers are supported:

- IBM COBOL Set for AIX V1.0 (5765-548 28H2176/33H4408)
- Micro Focus COBOL for UNIX Version 3.1 and v4.0

The following C++ compilers are supported:

- IBM C Set++ for AIX V3.1 (5765-421)

The following PL/I compilers are supported:

- IBM PL/I set for AIX V1.1

Components for AIX

Components for AIX

MQSeries Client

The MQSeries client code for your UNIX platform.

Samples

Sample application programs.

Support for DCE in Samples

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

Runtime component

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

Base

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

MQSeries Client for Java

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

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

For information about Java runtime see "Available from the Internet" on page 103.

On Sun Solaris, you require version 2.5.1 of the Solaris Operating System plus the following patches: 103566-08, 103600-13, 103640-08.

On HP-UX, you require HP-UX Version 10.20.

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

MQSeries Internet/ Java documentation

MQSeries Java Client and Internet Gateway documentation. This is supplied in HTML format.

MQSeries Internet Gateway

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

DOS Client: Hardware and Software Required

Hardware

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

Programming Requirements

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

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

Connectivity

- NetBIOS
- IBM TCP/IP V2.1.1 for DOS (87G7184 5621-219 (EMEA))

Workstation clients

MQSeries client code for DOS workstations is distributed with the server code for all MQSeries products (except MQSeries for AS/400, MQSeries for MVS/ESA, and MQSeries for Tandem NonStop Kernel). It operates under:

- DOS 5.0

Options, not prerequisites

- IBM TCP/IP for DOS: NetBIOS, V2.1.1 (87G7186 5622-048 (EMEA))
- IBM TCP/IP for OS/2 V3.0 is part of the base operating system
- Internet Connection Corporate kit for Windows 3.1 and Windows 95, V5.0
- Novell Netware Client for DOS/Win, V1.20
- Novell Netware Client for OS/2, V2.10
- Novell LAN Workplace, V5.1
- FTP TCP for DOS

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

Components for DOS

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

Compilers for MQSeries Applications on DOS Clients

The following C compilers are supported:

- Microsoft C V7.0
- Microsoft Visual C++ V1.5

Components for DOS

MQSeries Client

The MQSeries client code for your platform.

MQSeries Toolkit

This includes:

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

HP-UX Client: Hardware and Software Required

Hardware

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

Programming Requirements

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

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

Connectivity

For TCP/IP connectivity:

- TCP/IP (in the operating system)

For SNA connectivity:

- HP SNAplus2

Workstation MQSeries clients

MQSeries client code for HP-UX workstations is distributed with the server code for all MQSeries Version 5 products. It operates under:

- HP-UX Version 10.10, or later Version 10.

Options, not prerequisites

- The HP DCE/9000 version appropriate for the level of the HP-UX operating system in use, providing this is compatible with DCE Version 1.4.1 as supplied for HP-UX 10.10. This must be the U.S. Domestic version supporting DES encryption if the user wishes to run the MQSeries-supplied DCE send, receive, or message exits.

Note: On HP-UX 10.10, it is critical to apply HP service otherwise the HP DCE product will not work. Patches are listed in the Release Note delivered with the HP DCE software; however you are recommended to contact your local HP support center to obtain an up-to-date list of the required patches.

Components for HP-UX

MQSeries DCE names and security modules are provided as part of the MQSeries for HP-UX product.

Compilers for MQSeries Applications on HP-UX Clients

The following COBOL compilers are supported:

- Micro Focus COBOL for UNIX Version 3.1
- COBOL Softbench Version 4.0

The following C compilers are supported:

- The compiler supplied with the operating system
- HP-UX ANSI compiler
- C Softbench Version 5.0
- HP C++ Version 3.1

The following C++ compilers are supported:

- HP C++ Version 3.1

Components for HP-UX

These are identical to those for AIX. See "Components for AIX" on page 44.

OS/2 Warp Client: Hardware and Software Required

Hardware

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

Programming Requirements

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

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

Workstation MQSeries clients

MQSeries client code for OS/2 Warp workstations is distributed with the server code for all MQSeries Version 5 products. It operates under:

- OS/2 Warp V4.0 (84H1426)

Options, not prerequisites

- Communications Manager/2 V1.11 for OS/2 (this includes LU 6.2 and NetBIOS) (79G0257/79G0258)
- IBM Communications Server for OS/2 V4.0 (84H1802)
- Novell Netware Client for OS/2, V1.20 (for direct IPX/SPX support)
- IBM Directory and Security Server for OS/2 Warp Version 4 or later compatible versions. This must be the U.S. Domestic version supporting DES encryption if the user wishes to run the MQSeries-supplied DCE send, receive, or message exits.
- If used as a DCE server this software is known to run adequately in the following environment:
 - On a Pentium® processor running 90 MHz or faster.
 - On a machine with 64 MB or more of memory.
 - Using OS/2 Warp Server V4.0 or later.

Components for OS/2 Warp

- MQSeries DCE names and security modules are provided as part of the MQSeries for OS/2 Warp product.

Compilers for MQSeries Applications on OS/2 Clients

The following COBOL compilers are supported:

- IBM VisualAge COBOL for OS/2 V1.1 (28H2177, 5622-793 (EMEA))
- Micro Focus COBOL Compiler Version 3.0.54 and V4.0 (32 bit)

The following C compilers are supported:

- IBM C Set++ for OS/2 V2.1 (82G3732/82G3735, 5604-535 EMEA))
- IBM VisualAge C++ for OS/2 Version 3.0
- Borland C++ Compiler Version 2.0

The following C++ compilers are supported:

- IBM VisualAge C++ for OS/2 Version 3.0

The following PL/I compilers are supported:

- IBM PL/I for OS/2 Version 1.2
- IBM VisualAge for PL/I for OS/2

Components for OS/2 Warp

MQSeries Client

The MQSeries client code for your platform.

MQSeries Toolkit

This includes:

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

MQSeries Client for Java

This allows Java applets running on your client machine to communicate with MQSeries. It includes security exits for encryption and authentication of messages sent across the Web by the MQSeries Client for Java. These exits consist of some Java classes. To use the client for Java you need to have Java 1.1.1 (or later compatible version) runtime code on your machine. On Windows NT, you require Microsoft Windows NT Version 4. On OS/2 the MQSeries client

Components for OS/2 Warp

for Java must be installed on an HPFS formatted drive. For information about Java runtime see "Available from the Internet" on page 103.

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

MQSeries Internet/ Java documentation

MQSeries Java Client and Internet Gateway documentation. This is supplied in HTML format.

MQSeries Internet Gateway

Provides access to MQSeries applications via HTML and CGI.

Sun Solaris Client: Hardware and Software Required

Hardware

An MQSeries client can run only on:

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

with a minimum system disk space of 25 MB.

Connectivity

- Any communications hardware supporting SNA/LU 6.2 and/or TCP/IP

Programming Requirements

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

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

- Sun Solaris V2.5.1 or later 2.X to include TCP/IP

It is recommended that the latest levels of all relevant patches to the base operating system are installed.

MQSeries Clients

Client code for Sun Solaris workstations is distributed with the server code for all MQSeries Version 5 products.

Connectivity

- SunLink SNA Peer-to-Peer Version 9.0 or later V9.X
- TCP/IP as part of the base operating system
- If token ring is to be used: SunLink Token Ring Interface /SBus, V3.0.2. This requires patch 102463

Compilers for MQSeries Applications on Sun Solaris Clients

The following COBOL compiler is supported:

- Micro Focus COBOL for UNIX Version 3.2

The following C compiler is supported:

- SPARCompiler C Version 4.0

The following C++ compiler is supported:

- SPARCompiler C++ Version 4.1

Components for Sun Solaris

These are identical to those for AIX. See "Components for AIX" on page 44.

Windows 3.1 Client: Hardware and Software Required

Hardware

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

Programming Requirements

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

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

Connectivity

- NetBIOS
- IBM TCP/IP V2.1.1 for DOS
- SPX

Workstation clients

MQSeries client code for Windows 3.1 workstations is distributed with the server code for all MQSeries products (except MQSeries for AS/400, MQSeries for MVS/ESA, and MQSeries for Tandem NonStop Kernel). It operates under:

- Windows 3.1
- WIN-OS/2 environment under OS/2
- Windows 95 in 16-bit mode
- Windows for Workgroups

Options, not prerequisites

- TCP/IP for OS/2 V2.0. The base kit is required. The DOS access kit allows clients access to TCP/IP via programs that run under WIN-OS/2.
- TCP/IP V2.1.1 for DOS/Windows
- Novell Netware client for DOS/Win31 V1.20

Components for Windows 3.1

- Novell Netware client for OS/2 V2.1 (allows clients to access SPX via programs that run under WIN-OS/2)

Compilers for MQSeries Applications on Windows 3.1 Clients

The following COBOL compilers are supported:

- Micro Focus COBOL for Windows Version 3.3

The following C compilers are supported:

- Microsoft C/C++ Version 7.0
- Microsoft Visual C++ for Windows Version 2.0

The following C++ compilers are supported:

- Microsoft Visual C++ Version 1.5

Components for Windows 3.1

These are identical to those for DOS. See “Components for DOS” on page 46.

Windows 95 Client: Hardware and Software Required

Hardware

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

Programming Requirements

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

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

Connectivity

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

Workstation MQSeries clients

MQSeries client code for Windows 95 workstations is distributed with the server code for all MQSeries Version 5 products. It operates under:

- Windows 95

Options, not prerequisites

- IBM DCE for Windows 95: V1.1 and later compatible versions. (This will become IBM Directory and Security Server for Windows 95 in 1998).

You cannot use the supplied DCE security exit (described in the book *MQSeries Intercommunication*) from an MQSeries client on Windows 95 connected to an MQSeries for HP-UX server or an MQSeries for Sun Solaris server.

Compilers for MQSeries Applications on Windows 95 Clients

The following COBOL compilers are supported:

- Micro Focus COBOL Workbench Version 4.0

The following C compilers are supported:

- IBM VisualAge C++ for Windows Version 3.5 (33H4979)
- Microsoft Visual C++ for Windows 95/NT Version 4.0

The following C++ compilers are supported:

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

Components for Windows 95

MQSeries Client

The MQSeries client code for Windows 95.

MQSeries Toolkit

This includes:

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

Windows NT Client: Hardware and Software Required

Hardware

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

Programming Requirements

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

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

Connectivity

- IBM Communications Server for Windows NT, V5.0 (4231747)
- IBM Personal Communications for Windows NT, V4.1
- Attachmate Extra! Personal Client, V6.1 & V6.2
- Microsoft SNA Server, V2.11 or V3
- TCP/IP, NetBIOS, and SPX are part of the base operating system
- OnNet SDK for Windows
- FTP Software

Workstation MQSeries clients

MQSeries client code for Windows NT workstations is distributed with the server code for all MQSeries Version 5 products. It operates under:

- Microsoft Windows NT V3.5.1 with service pack 5 applied to it, or Windows NT V4, to include TCP/IP, NetBIOS, and SPX.

Compilers for MQSeries Applications on Windows NT Clients

The following COBOL compilers are supported:

- Micro Focus COBOL for Windows NT Version 3.1J
- Micro Focus COBOL for Windows NT Version 3.3

The following C compilers are supported:

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

The following C++ compilers are supported:

- IBM VisualAge C++ Version 3.5 (33H4979)
- Microsoft Visual C++ Version 4.0

The following PL/I compilers are supported:

- IBM PL/I for Windows V1.2,
- IBM VisualAge for PL/I for Windows

Components for Windows NT

These are identical to those for OS/2 Warp. See "Components for OS/2 Warp" on page 50.

Components for Windows NT

Chapter 6. Installing MQSeries Clients

This chapter describes the following:

- “Installing on AIX”
- “Installing on DOS” on page 63
- “Installing on HP-UX” on page 64
- “Installing on OS/2 Warp” on page 66
- “Installing on Sun Solaris” on page 74
- “Installing on Windows NT or Windows 95” on page 76
- “Installing on Windows 3.1” on page 80

Installing on AIX

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

Note: If you plan to install an MQSeries client and server on the same machine, see “Installing the Server and Client on the Same Machine” on page 32. If you have a previous version of the MQSeries client installed on your AIX system, you must uninstall the old version before installing the Version 5 MQSeries client.

Before Installation

Before you can install the MQSeries client on your AIX system, you:

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

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

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

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

Installing on AIX

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

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

Creating the mqm User ID and Group

You can 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:

```
Security & Users
  Passwords
```

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

```
Security & Users
  Users
    Change / Show Characteristics of a User
```

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

Note: 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 applications that use the queue manager only do not need mqm group authority.

Installation

1. Logon as root.
2. Insert the MQSeries Client CD-ROM into the CD-ROM drive.
3. Type `xinstallm -ez`
The MQSeries Welcome window is displayed.
4. **Make sure you are installing the correct client** for your system, as displayed in the Welcome window.
A window is then displayed where you can make some choices.
5. Choose the software source: **CD-ROM**.
6. For **Which bundle of software would you like to install?** choose: **Media-defined**.
7. Click on **Install/Update**.
A bundle of software products is created:
`mqm.Client`
See "Components for AIX" on page 44 for details of the components.
8. Choose the `mqm.Client` bundle and click on **Install/Update** again.
A work in progress window gives information as the installation proceeds.
9. At the end of installation you can click on the **View log** button and scroll to the bottom of the log to see the filesets that have been installed successfully.

Installing on DOS

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

Using Setup

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

Installing on HP-UX

This results in a self-exploding file being run to generate a tree of sub-directories containing the DOS client. See “Components for DOS” on page 46 for details.

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

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

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

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

Installing on HP-UX

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

Note: If you plan to install an MQSeries client and server on the same machine, see “Installing the Server and Client on the Same Machine” on page 32.

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

Before Installation

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

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

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

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

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

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

Installing on HP-UX

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

Installation

Use the HP-UX **swinstall** program, or use SAM, after mounting the CD-ROM. For further details see the *HP-UX Administration Guide*. See also “Components for AIX” on page 44.

The component **MQSeries Client for Java** should be installed only if you have Java 1.1.2 (or later compatible) runtime code on your machine. Also you must use HP-UX Version 10.20.

The **depot** to use is then in the HPUX10/MQS500.000 file under the mount point.

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

Kernel Configuration

See the MQSeries family homepage for a SupportPac that gives additional performance information - see “Available from the Internet” on page 103.

Translated Messages

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

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

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

Installing on OS/2 Warp

Installing on OS/2 Warp

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

Note: If you plan to install an MQSeries client and server on the same machine, see “Installing the Server and Client on the Same Machine” on page 32. If you currently have a manually installed MQSeries client on your OS/2 Warp system from a previous release of MQSeries, you must manually delete it before attempting to install the Version 5 client. You must not install the Version 5 client onto a system which currently has a Version 2 MQSeries Server installed.

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

Installation

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

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

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

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

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

Installing on OS/2 Warp

space required for installation. See "Components for OS/2 Warp" on page 50 for details.

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

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

Use the push buttons as necessary:

- To display descriptions of the selected options, select **Descriptions**.
- To select all of the options, select **Select all**.
- To deselect all of the options, select **Deselect all**.

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

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

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

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

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

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

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

11. When installation is complete, the Installation and Maintenance panel is displayed. Select **OK**. The introductory MQSeries for OS/2 panel is then displayed. Leave the installation program by selecting the **Exit** push button.

Installing on OS/2 Warp

12. When the installation process is complete, a folder is created on the OS/2 desktop, containing objects as follows:

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

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

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

Unattended Installation on OS/2 Warp

It is possible to install MQSeries clients on OS/2 Warp workstations without user interaction being required at the workstation. Unattended installation can be performed using response files. The sample response file AMQISAM2.RSP shows an example of unattended installation.

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

Attention

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

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

Starting Unattended Installation

The following steps describe how to perform an unattended installation:

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

Directory	Language
\OS2\En_Us	US English
\OS2\Pt_BR	Brazilian Portuguese

Installing on OS/2 Warp

\OS2\Fr_FR	French
\OS2\De_DE	German
\OS2\It_IT	Italian
\OS2\Ja_JP	Japanese
\OS2\Ko_KR	Korean
\OS2\Es_ES	Spanish
\OS2\Zh_CN	Simplified Chinese
\OS2\Zh_TW	Traditional Chinese

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

```
INSTALL /A:I /R:J:\INSTMQS\AMQISAM2.RSP
/S:J:\INSTMQS
/L1:J:\INSTMQS\MQM.OUT
/L2:J:\INSTMQS\MQM.HIS /X
```

Note: When you issue this command, you must type it as one continuous line. In the example above, for clarity, the command is spread over several lines.

In this example, the MQSeries client is installed from the redirected drive J: on the server, according to the options supplied in the response file AMQISAM2.RSP. The response file specifies the drive and directory in which to install the client. Errors are logged in MQM.OUT and the history log is contained in MQM.HIS. The `/X` parameter specifies that the installation is non-interactive.

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

Installation and Maintenance Parameters

The installation and maintenance parameters are as follows:

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

Installing on OS/2 Warp

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

/A:*action*

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

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

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

/G:*include path*

Specifies the drive and path of a general response file to be included by the specific response file. For more information about response files, see "Installation Response Files" on page 71.

/L1:*error log*

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

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

Example:

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

/L2:*history log*

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

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

Example:

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

/R:*response file*

Specifies the drive, path, and file name of a response file; see "Installation Response Files" on page 71.

Example:

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

Installing on OS/2 Warp

/S:*source location*

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

/T:*install target directory*

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

/TU:*update CONFIG.SYS directory*

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

/X Specifies that the installation is fully automatic.

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

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

Installation Response Files

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

In an installation response file you can specify:

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

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

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

Installing on OS/2 Warp

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

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

Keywords for Response Files

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

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

Installing on OS/2 Warp

INCLUDE	<p>Specifies which general response files to include with a specific response file. The format of this keyword is:</p> <p>INCLUDE = <i>filespec</i></p> <p>Where <i>filespec</i> specifies the general response file to be included. If the file specification contains any global characters (* or ?), the first file found that matches the specification is included. If the specification is not valid, no general response file is included.</p> <p>Note: You should not have more than five levels of included response files.</p> <p>Actions: All actions.</p>
OVERWRITE	<p>Specifies whether to overwrite files automatically during installation. Valid values for this keyword are YES and NO.</p> <p>Actions: INSTALL, UPDATE.</p>
SAVEBACKUP	<p>Specifies whether to save a backup version of MQSeries for OS/2 Warp when it is updated. Valid values for this keyword are YES and NO.</p> <p>Actions: UPDATE.</p>

Response File Structure

There are two kinds of line in a response file:

- **Comment lines**

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

- **Response lines**

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

Response lines have the following syntax:

keyword = value

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

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

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

Installing on Sun Solaris

Installing on Sun Solaris

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

Note: If you plan to install an MQSeries client and server on the same machine, see “Installing the Server and Client on the Same Machine” on page 32.

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

Before Installation

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

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

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

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

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

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

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

Installation

Carry out the following procedure:

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

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

Installing on Sun Solaris

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

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

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

2. Use the Sun Solaris **pkgadd** program, to install the MQSeries client software by carrying out the following procedure:
 - a. Type `pkgadd -d /cdrom/mqclient/solaris/mqs500.img`.
 - b. You are prompted for a list of components to be installed. Select the ones you require - if you want to install all the components, select **all**.
See "Components for AIX" on page 44 for details.
The component **MQSeries Client** for Java should be installed only if you have Java 1.1.1 (or later compatible) runtime code on your machine. Also you require version 2.5.1 of the Solaris Operating System plus the following patches: 103566-08, 103600-13, 103640-08.
 - c. Press the Enter key.

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

Kernel Configuration

See the MQSeries family homepage for a SupportPac that gives additional performance information - see "Available from the Internet" on page 103.

Translated messages

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

For example:

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

Installing on Windows NT or 95

Installing on Windows NT or Windows 95

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

Note: If you plan to install an MQSeries client and server on the same machine, see “Installing the Server and Client on the Same Machine” on page 32.

Note for Windows NT users: If you have the IBM Anti-Virus software installed on your Windows NT workstation/Server Version 4 and you install IBM MQSeries, the folders containing the MQSeries programs and sub-folders will not be created. The solution to this problem is to de-install IBM Anti-Virus software, install MQSeries and then re-instate IBM Anti-Virus software.

Installation

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

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

This window presents you with a list of the National Languages that are available.

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

The MQSeries Welcome window is displayed.

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

5. Select Destination Directory requires a destination directory into which the MQSeries files will be installed.

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

6. Choose MQSeries Components displays a list of components from which you can select the ones you want to be installed. See “Components for OS/2 Warp” on page 50 or “Components for Windows 95” on page 57 for details.

The component **MQSeries Client for Java** should be installed only if you have Java 1.1.1 (or later compatible) runtime code on your machine. Also on a Windows NT client you must use Microsoft Windows NT Version 4.

Installing on Windows NT or 95

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

Click on the **Next** button or press Enter to continue.

7. Select Program Folder prompts you for a folder name to contain the MQSeries objects. The default name is MQSeries Client for Windows NT (or 95). You can rename the default or choose an existing folder.

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

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

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

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

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

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

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

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

Unattended install on Windows 95 and Windows NT

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

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

For unattended installation, you must:

Installing on Windows NT or 95

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

In this example, the machine is being used as a server – see “Setting up the Server.”

2. Use the recording you have created to install the product on a remote machine – see “Installing on the Remote Machine” on page 79.

Setting up the Server

To set up your server, follow these steps:

1. Load the MQSeries client CD-ROM. When the *Language Selection* window appears, select **Cancel**.
Note: For an unattended installation, you will know the language you require.
2. Copy the *win95* or the *winnt* subdirectory, and all of its subdirectories, from the client CD-ROM to a shared-resource image directory.
3. Using the Microsoft Windows NT Explorer, or an interface of your choice, select the *winnt* directory from the CD-ROM or the shared resource.
4. Select one of the following language directories:
 - SetupCn – Simplified Chinese
 - SetupDe – German
 - SetupEn – English
 - SetupEs – Spanish
 - SetupIt – Italian
 - SetupJp – Japanese
 - SetupKo – Korean
 - SetupPt – Portuguese
 - SetupTw – Traditional Chinese
5. From the selected language directory, run `setup.exe`, with the `-r` option to create a response file.

This can be done by:

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

Installing on Windows NT or 95

Notes:

- a. When the installation has ended, a response file will have been automatically generated that can be used as the basis for unattended installation.
 - b. The response file, SETUP.ISS will be generated in the WINDOWS directory for Windows 95 or the WINNT35 or the WINNT directory, depending on the version of Windows NT that you are using.
6. When the response file has been generated, you can copy it to the language dependent directory in the \shared-resource image directory for your national language.

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

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

where SETUPEN is the appropriate language directory for your system.

This assumes that the Windows NT operating system was installed to the C drive in a directory WINNT35 and that the MQSeries installation image on the server resides on M:\INSTMQS.

Similarly, to copy the SETUP.ISS file from the WINDOWS directory to \IMAGE\SETUPEN, type:

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

where SETUPEN is the appropriate language directory for your system.

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

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

Installing on the Remote Machine

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

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

- Appropriate language subdirectory
- Disk subdirectories

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

1. Use the NET USE command as follows:

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

For example:

Installing on Windows 3.1

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

where:

- J: is the redirected drive containing the MQSeries client you are installing.
 - MQMSHARE is the netname for M:\INSTMQS.
2. Go to the language directory that contains the SETUP.EXE and SETUP.ISS files and run **SETUP -s**.

Notes:

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

Installing on Windows 3.1

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

Installation

1. Insert the MQSeries Client CD-ROM into the CD-ROM drive.
2. Open the directory WIN31.
3. Change to the appropriate language subdirectory for the language you wish to install:

```
setupen - English  
setupfr - French  
setupde - German  
setupes - Spanish  
setupit - Italian  
setuppt - Portuguese Brazilian  
setupjp - Japanese  
setupko - Korean  
setupcn - Simplified Chinese  
setuptw - Traditional Chinese
```

4. Run setup.exe

The MQSeries Welcome window is displayed.

Installing on Windows 3.1

5. **Make sure you are installing the correct client** for your system, Windows 3.1, as displayed in the Welcome window.

6. Select Destination Directory requires a destination directory into which the MQSeries files will be installed.

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

7. Choose MQSeries Components displays a list of components from which you can select the ones you want to be installed. See “Components for DOS” on page 46 for details.

To select a component, click in the square next to it so that a check mark appears (just highlighting the line does not select it). The space needed for each component is shown here, and the space available on the drive you have selected. Click on the **Next** button or press Enter to continue.

8. Select Program Folder prompts you for a folder name to contain the MQSeries objects. The default name is MQSeries Client for Windows 3.1. You can rename the default or choose an existing folder.

9. Start Copying Files displays the selection you have made. Click on the **Back** button if you want to return to change your choice.

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

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

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

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

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

Select Yes, I want to restart my computer now and click on the **Finish** button to complete the setup.

12. When setup is complete, the MQSeries Client folder is added to the Program Manager. Note that the MQSeries client is a set of services and it does not have to be explicitly run, so the folder does not have an object called a “client”.

Configuring a Client

Configuring and Verifying a Client

After you have installed the client, you need to configure your communications and verify the installation. For information on how to do this, see the *MQSeries Clients* manual.

Part 3. Using MQSeries for Sun Solaris

This part of the book shows you how to use MQSeries for Sun Solaris It contains the following chapters:

- Chapter 7, "Using the MQSeries Command Sets" on page 85
- Chapter 8, "Using the MQSeries World Wide Web Interface" on page 97
- Chapter 9, "Obtaining Additional Information" on page 99

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, and channels. To perform these tasks, you must select the appropriate command from one of the supplied command sets (see "Introducing Command Sets").

Introducing Command Sets

MQSeries provides three command sets for performing administration tasks:

- Control commands
- MQSC commands
- PCF commands

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

Control Commands

Control commands fall into three categories:

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

MQSeries Command Sets

Using Control Commands

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

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

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

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

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

crtmqcvx (data conversion)

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

crtmqm (create queue manager)

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

dltmqm (delete queue manager)

Deletes a specified queue manager.

dmpmqlog (dump log)

Dumps a formatted version of the MQSeries system log.

dspmqaut (display authority)

Displays the current authorizations to a specified object.

dspmqcsv (display command server)

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

dspmqfls (display MQSeries files)

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

dspmqtrc (display MQSeries formatted trace output)

Displays MQSeries formatted trace output.

dspmqtrn (display MQSeries transactions)

Displays details of in-doubt transactions.

endmqcsv (end command server)

Stops the command server on the specified queue manager.

endmqm (end queue manager)

Stops a specified local queue manager.

MQSeries Command Sets

endmqtrc (end MQSeries trace)

Ends tracing for the specified entity or all entities.

rcdmqimg (record media image)

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

rcrmqobj (recreate object)

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

rsvmqtrn (resolve MQSeries transactions)

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

runmqchi (run channel initiator)

Runs a channel initiator process.

runmqchl (run channel)

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

runmqdlq (run dead-letter queue handler)

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

runmqsc (run MQSeries commands)

Issues MQSC commands to a queue manager.

runmqtrm (start trigger monitor)

Invokes a trigger monitor.

setmqaut (set/reset authority)

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

setmqprd (enroll production license)

Enrolls the MQSeries production License.

setmqtry (start trial period)

Starts the MQSeries trial period.

strmqcsv (start command server)

Starts the command server for the specified queue manager.

strmqm (start queue manager)

Starts a local queue manager.

strmqtrc (start MQSeries trace)

Enables tracing.

For more information about the syntax and purpose of control commands, see the *MQSeries System Administration* manual.

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

Working With Queue Managers

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 Command Reference*.

Running MQSC Commands

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

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

For more information about using MQSC commands, see the *MQSeries System Administration* manual.

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* manual.

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* manual.

Starting a Queue Manager

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

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

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

Stopping a Queue Manager

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

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

Working With Queue Managers

Quiesced Shutdown

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

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

Immediate Shutdown

For an *immediate shutdown* any current MQI calls are allowed to complete, but any new calls fail. This type of shutdown does not wait for applications to disconnect from the queue manager.

Use this as the normal way to stop the queue manager, optionally after a quiesce period. For an immediate shutdown, the command is:

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

Preemptive Shutdown

Preemptive shutdown

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

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

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

This stops all queue manager code immediately.

Restarting a Queue Manager

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

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

Deleting a Queue Manager

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

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

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

Working with MQSeries Objects

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

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

For more information about using MQSC commands, see the *MQSeries System Administration* manual. For a complete description of the MQSC commands, see the *MQSeries Command Reference*.

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 89.

Using the MQSC Facility Interactively

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

```
runmqsc
```

In this command, a queue manager name has not been specified, therefore the MQSC commands will be processed by the default queue manager. Now you can type in any MQSC commands, as required. For example, try this one:

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

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

Working with Objects

- 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 Command Reference* manual for the correct syntax.

Ending Interactive Input to MQSC

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

```
END
```

Alternatively, you can use the EOF character CTRL+D

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

Defining a Local Queue

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

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

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

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

The following MQSC command does this:

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

Notes:

1. Most of these attributes are the defaults as supplied with the product. However, they are shown here for purposes of illustration. You can omit them if you are sure that the defaults are what you want or have not been changed. See also "Displaying Default Object Attributes" on page 94.
2. USAGE (NORMAL) indicates that this queue is not 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 95.

Working with Objects

Displaying Default Object Attributes

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

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

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

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

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

This command displays the three specified attributes as follows:

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

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

Copying a Local Queue Definition

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

For example:

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

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

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

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

Working with Objects

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

Notes:

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

Changing Local Queue Attributes

You can change queue attributes in two ways, using either the `ALTER QLOCAL` command or the `DEFINE QLOCAL` command with the `REPLACE` attribute. In “Defining a Local Queue” on page 93, 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`.

Working with Objects

- An application currently has the queue open.

Deleting a Local Queue

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

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

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

Browsing Queues

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

The default file names and paths are:

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

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

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

There are no defaults; both parameters are required.

Chapter 8. Using the MQSeries World Wide Web Interface

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

Overview of MQSeries Internet Gateway

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

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

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

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

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

Obtaining More Information

The MQSeries product family Web site is at:

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

The following documentation is accessible from this Web site:

Using the Web Interface

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

Obtaining MQSeries Internet Gateway

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

The following Gateways are available:

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

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

Chapter 9. Obtaining Additional Information

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

The MQSeries for Sun Solaris Library

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

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

MQSeries Library

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

Viewing Online Information

This section describes viewing:

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

Viewing Online Help

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

Viewing Online Books

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

The MQSeries online documentation is in PostScript format on the CD-ROM, in the `/BOOKS/PS` directory.

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

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

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

BookManager Books

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

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

Printing and Ordering Books

HTML Books on the World Wide Web

The books listed in “The MQSeries for Sun Solaris Library” on page 99 are supplied in HTML format on the MQSeries Server CD-ROM, and on the World Wide Web. For the latest information on using the HTML books, see “Available from the Internet” on page 103.

Printing and Ordering Books

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

Printing PostScript Books

The MQSeries for Sun Solaris books are supplied in PostScript format on the MQSeries for Sun Solaris Server CD-ROM.

Table 3 on page 103 shows the file names used for the PostScript files.

Note: The fifth character in the name represents the National Language of the book.
For example:

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

Internet Availability

Table 3. MQSeries publications – file names

Book	File Name
<i>MQSeries Planning Guide</i>	CSQZAB00
<i>MQSeries for Sun Solaris Quick Beginnings</i>	AMQDAC00
<i>MQSeries Intercommunication</i>	CSQZAE00
<i>MQSeries Clients</i>	CSQZAF00
<i>MQSeries System Administration</i>	AMQZAG00
<i>MQSeries Programmable System Management</i>	CSQZAI00
<i>MQSeries Command Reference</i>	CSQZAJ00
<i>MQSeries Application Programming Reference</i>	CSQZAK00
<i>MQSeries Application Programming Guide</i>	CSQZAL00
<i>MQSeries Application Programming Reference Summary</i>	CSQZAM00
<i>MQSeries Using C++</i>	AMQZAN00
<i>MQSeries Messages</i>	AMQZA000

Available from the Internet

The MQSeries product family Web site is at:

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

By following links from this Web site you can:

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

Internet Availability

Part 4. Appendixes

Appendix A. Installing the MQSeries Product License

This appendix tells you how to enroll the MQSeries product license if you:

- Are using the trial product
- Have previously been using the trial version of the product

Enrolling the Trial-Period License

The trial-period license is **not** enrolled automatically.

When you have installed the product and issue a **crtmqm** or **strmqm** command, you receive a message informing you that the product license needs to be installed.

You **must** use the **setmqtry** command to start the trial license period. Before the trial period begins you have to accept a license agreement to enroll the product license.

Subsequently, whenever you issue a **crtmqm** or **strmqm** command, you are told how many days remain in the trial period. These commands fail when the trial period ends.

Enrolling the Product License

You can install the product license:

- Locally – see “Enrolling the Product License Locally” on page 108
- Remotely – see “Enrolling the Product License Remotely” on page 108

Note to users

You do not have to reinstall the product to enroll the production license.

Enrolling the License

Enrolling the Product License Locally

You enroll the production license by issuing the following command:

```
setmqprd /dev/cdrom/licenses/amqppswd.lic
```

where /dev/cdrom is your primary CD-ROM device.

Enrolling the Product License Remotely

If you installed the product remotely using the trial version of the product, you will have to use **setmqprd** at each remote location, using the following command – `setmqprd /dev/cdrom/licenses/amqppswd.lic`

where /dev/cdrom is your primary CD-ROM device, or reinstall MQSeries.

Appendix B. Migrating from an Earlier Version of MQSeries

This appendix tells you how to install the current version of the product over an earlier version.

Stop MQSeries...

You must halt all MQSeries activity on the target machine before installing the new release.

You do not have to recreate your MQSeries objects.

Carry out the following procedure:

1. Install the new version of the product as described in "Installing the MQSeries for Sun Solaris Server" on page 29.
2. Issue a **crtmqm** or **strmqm** command.

Migration

Appendix C. Sample MQI Programs and MQSC Files

MQSeries for Sun Solaris provides a set of short sample MQI programs and MQSC command files. You can use these directly or modify them for experimental purposes.

MQSC Command File Samples

Table 4 lists the MQSC command file samples. These are simply ASCII text files containing MQSC commands. You can invoke the **runmqsc** command against each file in turn to create the objects specified in the file.

By default, these files are located in directory `/opt/mqm/samp`

File name	Purpose
amqscic0.tst	Defines objects for use in the sample CICS transaction.
amqscos0.tst	Creates a set of MQI objects for use with the C and COBOL program samples.
amqmdefs.tst	Defines objects for the administration application sample.

C and COBOL Program Samples

Table 5 lists the sample MQI source files. By default, the source files are `/opt/mqm/samp`, and the compiled versions in directory `/opt/mqm/samp/bin`. To find out more about what the programs do and how to use them, see the *MQSeries Application Programming Guide*.

C	COBOL	Purpose
amqsbcg0.c	–	Reads and then outputs both the message descriptor and message context fields of all the messages on a specified queue.
amqsecha.c	amqmechx.cbl	Echoes a message from a message queue to the reply-to queue. Can be run as a triggered application program.
amqsgbr0.c	amq0gbr0.cbl	Writes messages from a queue to stdout, leaving the messages on the queue. Uses MQGET with the browse option.

Samples

<i>Table 5 (Page 2 of 2). Sample programs - source files</i>		
C	COBOL	Purpose
amqsget0.c	amq0get0.cbl	Removes the messages from the named queue (using MQGET) and writes them to stdout.
amqsinqa.c	amqminqx.cbl	Reads the triggered queue; each request read as a queue name; responds with information about that queue.
amqsput0.c	amq0put0.cbl	Copies stdin to a message and then puts this message on a specified queue.
amqsreq0.c	amq0req0.cbl	Puts request messages on a specified queue and then displays the reply messages.
amqsseta.c	amqmsetx.cbl	Inhibits puts on a named queue and responds with a statement of the result. Runs as a triggered application.
amqstrg0.c	–	A trigger monitor that reads a named initiation queue and then starts the program associated with each trigger message. Provides a subset of the full triggering function of the supplied runmqtrm command.
amqsvfcx.c	–	A sample C skeleton of a Data Conversion exit routine.
amqsptl0.c	–	Putting messages to a distribution list.
amqsprma.c	–	Putting reference messages to a queue.
amqsgrma.c	–	Getting reference messages from a queue.
amqsxrma.c	–	Reference message channel exit.
Note: You can create the objects required by these samples using the MQSC command file amqscos0.tst.		

Supporting CICS and Encina for Transaction Processing

The samples include a CICS transaction and some associated headers and initialization programs.

<i>Table 6. Samples for transaction processing with CICS and Encina</i>	
File name	Purpose
amqzscix.c	CICS initialization program.
amqscic0.ccs	Sample CICS program
amqzscgx.c	GLUE program for CICS for Solaris
amqscih0.h	Header file for CICS transaction sample amqscic0.
amqsxae0.c	Encina transaction
Note: Objects to support transaction processing can be created using the MQSC command file amqscic0.tst.	

Supporting Tuxedo for Transaction Processing

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

<i>Table 7. Samples for transaction processing with Tuxedo</i>	
File name	Purpose
amqstxsx.c	Sample server
amqstxgx.c	Sample GET client application
amqstxpx.c	Sample PUT client application
amqstxvx.flds	Field definition
ubbstxcx.cfg	Configuration file

Samples

Supporting Lotus Notes and MQSeries Link

<i>Table 8. Samples for Lotus Notes</i>	
File name	Purpose
amqslnk0.c	MQSeries link

Supporting Databases

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

<i>Table 9. Sample programs - databases</i>		
C	COBOL	Purpose
amqsxas0.c	amqsxas0.cbl	Updates a single database within an MQSeries unit of work.
amqsxag0.c	amqsxag0.cbl	amqsxag0.c together with amqsxab0.sqc and amqsxaf0.sqc, or amqsxag0.cbl together with amqsxab0.sqb and amqsxaf0.sqb, update two databases within an MQSeries unit of work.

Miscellaneous Tools

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

<i>Table 10. Miscellaneous files</i>		
File name	Location	Purpose
amqtrc.fmt	<code>/opt/mqm/lib</code>	Defines MQSeries trace formats.
ccsid.tbl	<code>/var/mqm/conv/table</code>	Edit this file to add any newly supported CSSID values to your MQSeries system.

Appendix D. Applying maintenance to MQSeries for Sun Solaris

This appendix tells you how to apply maintenance to MQSeries for Sun Solaris.

Maintenance updates in the form of a Program Temporary Fix (PTF) are supplied on CD-ROM. They can also be downloaded from :

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

Attention

Do not have any queue managers operating during installation of maintenance on MQSeries for Sun Solaris.

To end all running queue managers:

1. End the queue manager by issuing the command:

```
endmqm -i QMgrName
```

2. Check that the queue manager has ended.

Use the command:

```
endmqm -p QMgrName
```

and check that the message queue manager is not running.

Alternatively, use the command:

```
ps -ef | grep amq
```

where | is the pipe symbol. Check that there are no amq***** files listed, ignoring any commencing with amqi.

Space Requirements

A PTF requires hard disk space for installation. In addition, the installation process requires an identical amount of disk space to save the previous level. For example, a 16 MB PTF requires 32 MB of space.

Restoring Levels

This allows a PTF to be removed, and the previous level to be automatically restored. If disk space is limited, the backup can be suppressed by creating an empty flag file called MQPTF_NOSAVE in the directory `/var/sadm/pkg`.

Note that if this option is used, the previous level will not be restored if a PTF is removed. The only way to restore a previous level in this instance is to reinstall the product and then to reapply a previous PTF image.

Applying the maintenance information

1. Mount the CD by typing the following commands:

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

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

2. Install the software as follows:

- a. Type `pkgadd -d /cdrom/mq_solaris/mqm/patchname`.
- b. Press the Enter key.

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

Restoring the Previous Service Level

To restore the previous service level:

1. Log in as root, or use the command `su`.
2. Use the `pkgrm` command to remove the latest PTF from the system.

For example, to remove a service update, issue the following command:

```
pkgrm mqm-upd0x
```

where 'x' is a number.

Note: Any error messages displayed of the form `<shared pathname not removed>` can be ignored.

Details of the `pkgrm` command can be found in the Solaris documentation, or by using the `man pkgrm` command.

3. If you have installed an MQI client, and the client was updated after installing the PTF that is being removed, then you *must* specifically update your MQI client installation again, after the PTF has been removed.

Appendix E. Support for different code sets on MQSeries for Sun Solaris

MQSeries for Sun Solaris supports most of the code sets used by the locales – that is, the subsets of the user's environment which define the conventions for a specific culture – that are provided as standard on MQSeries for Sun Solaris.

If the locale is not set, the value of the LANG environment variable is used. If neither the locale nor LANG environment variable is set the CCSID used is 819 - the ISO 8859-1 code set.

Note: Not all the locales listed below are supported by all versions of Solaris.

The CCSID (Coded Character Set Identifier) used in MQSeries to identify the code set used for the message and message header data is obtained by analyzing the LC_CTYPE environment variable.

Table 11 shows the locales and the CCSIDs that are registered for the code set used by the locale.

Locale	Language	code set	CCSID
C	English	ISO 8859-1	819
ar	Arabic	ISO 8859-6	1089
ar_AA	Arabic	ISO 8859-6	1089
bu	Bulgarian	ISO 8859-5	915
bu_BG	Bulgarian	ISO 8859-5	915
cs	Czech	ISO 8859-2	912
cs_CZ	Czech	ISO 8859-2	912
da	Danish	ISO 8859-1	819
da_DK	Danish	ISO 8859-1	819
de	German	ISO 8859-1	819
de_DE	German	ISO 8859-1	819
de_AT	German - Austria	ISO 8859-1	819
de_CH	German - Switzerland	ISO 8859-1	819
el	Greek	ISO 8859-7	813
el_GR	Greek	ISO 8859-7	813
en	English - United Kingdom	ISO 8859-1	819
en_GB	English - United Kingdom	ISO 8859-1	819
en_UK	English - United Kingdom	ISO 8859-1	819
en_AU	English - Australia	ISO 8859-1	819

Supported code sets

Table 11 (Page 2 of 3). Locales and CCSIDs

Locale	Language	code set	CCSID
en_CA	English - Canada	ISO 8859-1	819
en_US	English - USA	ISO 8859-1	819
es	Spanish	ISO 8859-1	819
es_ES	Spanish	ISO 8859-1	819
fi	Finnish	ISO 8859-1	819
fi_FI	Finnish	ISO 8859-1	819
fr	French - France	ISO 8859-1	819
fr_FR	French - France	ISO 8859-1	819
fr_BE	French - Belgium	ISO 8859-1	819
fr_CA	French - Canada	ISO 8859-1	819
fr_CH	French - Switzerland	ISO 8859-1	819
hr	Croatian	ISO 8859-2	912
hr_HR	Croatian	ISO 8859-2	912
hu	Hungarian	ISO 8859-2	912
hu_HR	Hungarian	ISO 8859-2	912
is	Icelandic	ISO 8859-1	819
is_IS	Icelandic	ISO 8859-1	819
it	Italian - Italy	ISO 8859-1	819
it_IT	Italian - Italy	ISO 8859-1	819
it_CH	Italian - Switzerland	ISO 8859-1	819
iw	Hebrew	ISO 8859-8	916
iw_IL	Hebrew	ISO 8859-8	916
ja	Japanese	eucJP	5050
ja_JP	Japanese	eucJP	5050
ko	Korean	eucKR	970
ko_KR	Korean	eucKR	970
mk	Macedonian	ISO 8859-5	915
mk_MK	Macedonian	ISO 8859-5	915
nl	Dutch - Netherlands	ISO 8859-1	819
nl_NL	Dutch - Netherlands	ISO 8859-1	819
nl_BE	Dutch - Belgium	ISO 8859-1	819
no	Norwegian	ISO 8859-1	819
no_NO	Norwegian	ISO 8859-1	819
pl	Polish	ISO 8859-2	912
pl_PL	Polish	ISO 8859-2	912
POSIX	English	ISO 8859-1	819
pt	Portuguese	ISO 8859-1	819
pt_PT	Portuguese	ISO 8859-1	819
ro	Romanian	ISO 8859-2	912
ro_RO	Romanian	ISO 8859-2	912

Supported code sets

Table 11 (Page 3 of 3). Locales and CCSIDs

Locale	Language	code set	CCSID
ru	Russian	ISO 8859-5	915
ru_RU	Russian	ISO 8859-5	915
ru_SU	Russian	ISO 8859-5	915
sh	Serbocroatian	ISO 8859-2	912
sh_SP	Serbocroatian	ISO 8859-2	912
sh_YU	Serbocroatian	ISO 8859-2	912
sl	Slovene	ISO 8859-2	912
sl_SL	Slovene	ISO 8859-2	912
sk	Slovak	ISO 8859-2	912
sk_SK	Slovak	ISO 8859-2	912
sr	Serbian Cyrillic	ISO 8859-5	915
sr_SP	Serbian Cyrillic	ISO 8859-5	915
sv	Swedish	ISO 8859-1	819
sv_SE	Swedish	ISO 8859-1	819
tr	Turkish	ISO 8859-9	920
tr_TR	Turkish	ISO 8859-9	920
zh	Simplified Chinese	eucCN	1383
zh_TW	Traditional Chinese	eucTW	964

For further information listing inter-platform support for these locales, see the *MQSeries Application Programming Reference*.

Supported code sets

Appendix F. Notices

References in this publication to IBM products, programs, or services do not imply that IBM intends to make them available in all countries in which IBM operates. 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. Subject to IBM's valid intellectual property or other legally protectable rights, any functionally equivalent product, program, or service may be used instead of the IBM product, program, or service. The evaluation and verification of operation in conjunction with other products, except those expressly designated by IBM, are the responsibility of the user.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
500 Columbus Avenue
Thornwood, NY 10594
USA

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:

Site Counsel
IBM Corporation
P.O. Box 12195
3039 Cornwallis Road
Research Triangle Park, NC 27709-2195
USA

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

Notices

Trademarks

The following terms are trademarks of the IBM Corporation in the United States or other countries or both:

AIX	AS/400
BookManager	CICS
DB2	FFST
First Failure Support Technology	IBM
IBMLink	MQ
MVS	MVS/ESA
OpenEdition	OS/2
OS/400	RISC System/6000
S/390	SP
SystemView	VisualAge
WIN-OS/2	Workplace
400	

Lotus, Lotus Notes, 1-2-3, LotusScript and Domino are trademarks or registered trademarks of Lotus Development Corporation in the United States, or other countries, or both.

Tivoli and TME 10 are trademarks or registered trademarks of Tivoli Systems Inc. in the United States, or other countries, or both.

Encina is a registered trademark of Transarc Corporation in the United States, or other countries, or both.

Java and HotJava are trademarks of Sun Microsystems, Inc.

Microsoft, Windows, Windows NT, and the Windows logo are registered trademarks of Microsoft Corporation.

UNIX is a registered trademark in the United States and 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 85
 MQSeries commands (MQSC) 87
 programmable command format commands (PCF) 88

AIX client
 hardware and software 42
 installing 63

application
 data 10
 time-independent 10

applications on MQSeries clients
 non-version 5 41
 Version 5 41

attributes
 ALL attribute 94
 changing 95
 default 94

B

bibliography 99
 BookManager 101
 browsing queues 96

C

case-sensitive control commands 86
 CCSID 117
 CCSID, setting 38
 changing queue attributes 95
 channel
 events 15
 message 13
 MQI 13

clearing a local queue 95
 client definitions, migrating 42
 clients 13
 installing on DOS 63
 installing on HP-UX 65
 installing on OS/2 Warp 66
 installing on Sun Solaris 74

clients (*continued*)

installing on Windows 3.1 80
 installing on Windows 95 76
 installing on Windows NT 76
 silent install on Windows 95 68
 Unattended install on OS/2 Warp 68
 unattended install on Windows 95 77
 unattended install on Windows NT 77

code set 117
 coded character set
 identifier 117
 command set administration 85
 commands
 control 85
 MQSC
 ALTER QLOCAL 95
 DEFINE QLOCAL 94
 DEFINE QLOCAL LIKE 94
 DEFINE QLOCAL REPLACE 95
 DELETE QLOCAL 96
 using 88
 programmable command format (PCF) 88
 runmqsc 91

configuration
 kernel 30
 configuration, kernel 65, 75
 control commands
 case-sensitive 86
 runmqsc 91
 controlled shutdown 90
 creating
 file system for product code 28
 groups 27, 61, 64, 74
 queue manager 89
 user ID 61, 64, 74
 users 27
 current queue depth 94

D

default
 attributes of objects 94
 queue manager commands processed 91
 deleting 96

Index

deleting (*continued*)
 local queue 96
 queue manager 91
disk requirements for installation 19
DOS client
 hardware and software 45
 installing 63

E

earlier versions
 migrating 42
ending
 interactive MQSC commands 92
 queue manager 90
endmqm command 89
environment variable
 LANG 32
 NLSPATH 32
error log
 for automatic installation 70
error messages 92
event-driven processing 10
events
 channel 15
 instrumentation
 types of 15

F

feedback from MQSC commands 92
file system, creating for product code 28

G

groups, creating 27, 61, 64, 74

H

hard disk requirements 19
hardware requirements
 AIX client 42
 DOS client 45
 HP-UX client 47
 OS/2 client 49
 Sun Solaris client 52
 Windows 3.1 client 54
 Windows 95 client 56
 Windows NT client 58

history log 70
HP-UX client
 hardware and software 47
 installing 65
Hypertext Markup Language (HTML) 102

I

information, ordering publications 102
installation
 kernel configuration 30
 migration 109
 preparation 27, 61
 procedure 29
 product license 107
 response file, specification for 79
installation directory 27, 74
installation parameters 69
installation response file
 parameter 70
 specification 71
installation, unattended 68
installing
 AIX client 63
 clients on DOS 63
 clients on HP-UX 65
 clients on OS/2 Warp 66
 clients on Sun Solaris 74
 clients on the server 32
 clients on Windows 3.1 80
 clients on Windows 95 76
 clients on Windows NT 76
 error log parameters 70
 history log parameter 70
 maintenance updates 115
 response file parameter 70
instrumentation events
 description 15
 types of 15
interactive MQSC
 ending 92
 feedback from 92
 using 91
introduction to MQSeries 9

K

kernel configuration 30, 65, 75
keywords for response files 72

L

- LANG environment variable 32
- LIKE attribute 94
- linking to Lotus Notes 14
- linking user exits 38
- local queues
 - clearing 95
 - copying definitions 94
 - defining one 93
 - deleting 96
- locale 117
- Lotus Notes
 - mapping 14
 - with MQSeries 14

M

- maintenance 115
- maintenance of MQSeries for Solaris
 - space requirements 115
- maintenance of MQSeries for Sun Solaris
 - installing updates 115
- maintenance parameters 69
- mapping Lotus Notes 14
- message
 - channels 13
 - description 10
 - descriptor 10
 - queuing 9
 - translated 75
- message length, decreasing 95
- message queue interface (MQI) 9
- message queuing 9
- message-driven processing 9
- message, translated 32, 65
- migrating client definitions 42
- migrating from Version 2 MQSeries products 42
- monitoring queue managers 15
- MQI
 - channel 13
 - description 9
- MQSC 87
- MQSC commands
 - ALTER QLOCAL 95
 - DEFINE QLOCAL 94
 - DEFINE QLOCAL LIKE 94
 - DEFINE QLOCAL REPLACE 95
 - DELETE QLOCAL 96
 - ending interactive input 92

- MQSC commands (*continued*)

- issuing interactively 91
 - using 88

- MQSC sample files 111

- MQSeries for Sun Solaris

- applying maintenance 115

- components 22

- hardware requirements 19

- overview of 19

- restoring previous service level 115

- software requirements 19

- MQSeries objects 11

- multiple threading 31

N

- national language support 32, 65

- NLSPATH environment variable 32, 65

- non-version 5 clients 41

O

- objects

- default attributes 94

- process definition 12

- queue manager 11

- online books 101

- ordering publications 102

- OS/2 Warp client

- hardware and software 49

- installing 66

- unattended install 68

- overview of MQSeries for Sun Solaris 19

P

- parameters for unattended installation 69

- performance events 15

- PostScript Format 102

- preemptive queue manager shutdown 90

- process definitions, description of 12

- processing, event-driven 10

- product license installation 107

- product migration 109

- programmable command format (PCF)

- administration with 88

- programs, using samples 111

- publications 99

Index

Q

- queue depth
 - current 94
 - determining 94
- queue manager
 - creating 89
 - deleting 91
 - description 11
 - events 15
 - immediate shutdown 90
 - monitoring 15
 - objects 11
 - preemptive shutdown 90
 - restart 90
 - shutdown
 - controlled 90
 - immediate 90
 - preemptive 90
 - quiesced 90
 - starting 89
 - stopping 89
- queues
 - attributes 12
 - browsing 96
 - changing attributes 95
 - defining 12
 - description 10
 - local
 - clearing 95
 - copying 94
 - defining 93
 - deleting 96
- quiesced shutdown 90

R

- README file 25
- requirements
 - disk storage 19
 - hardware 19
 - software 19
- response files
 - installation 71
 - keywords 72
 - structure 73
- response files, installation 79
- restart queue manager 90
- restoring
 - MQSeries for Solaris 116

- restoring (*continued*)
 - service level 116
- runmqsc
 - ending 92
 - feedback 92
 - using interactively 91

S

- sample
 - MQSC files 111
 - programs, using 111
- setting the CCSID 38
- shell commands for MQSeries 85
- shutdown queue manager
 - controlled 90
 - immediate 90
 - preemptive 90
 - quiesced 90
- silent install
 - clients on Windows 95 77
 - clients on Windows NT 77
- softcopy information 101
- software requirements
 - AIX client 42
 - DOS client 45
 - HP-UX client 47
 - OS/2 client 49
 - Sun Solaris client 52
 - Windows 3.1 client 54
 - Windows 95 client 56
 - Windows NT client 58
- starting
 - queue manager 89
 - stopping queue manager 89
- structure of response files 73
- Sun Solaris client
 - hardware and software 52
 - installing 74
- syntax error, in MQSC commands 92
- System Management Interface Tool (SMIT)
 - using to create IDs 62

T

- time-independent applications 10
- translated messages 32, 65, 75
- trigger monitor
- types of event 15

Index

U

- Unattended install
 - clients on OS/2 68
 - clients on Windows 95 77
 - clients on Windows NT 77
- unattended installation 68
- updating MQSeries for Sun Solaris 115
- user exits, linking 38
- user ID, creating 61, 64, 74
- users, creating 27

V

- Version 2 MQSeries products
 - migrating 42
- Version 5 clients 41

W

- Windows 3.1 client
 - hardware and software 54
 - installing 80
- Windows 95 client
 - hardware and software 56
 - installing 76
- Windows NT client
 - hardware and software 58
 - installing 76

Sending your comments to IBM

MQSeries for Sun Solaris

Quick Beginnings

GC33-1870-01

If you especially like or dislike anything about this book, please use one of the methods listed below to send your comments to IBM.

Feel free to comment on what you regard as specific errors or omissions, and on the accuracy, organization, subject matter, or completeness of this book. Please limit your comments to the information in this book and the way in which the information is presented.

To request additional publications, or to ask questions or make comments about the functions of IBM products or systems, you should talk to your IBM representative or to your IBM authorized remarketer.

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

You can send your comments to IBM in any of the following ways:

- By mail, use the Readers' Comment Form (RCF)
- By fax:
 - From outside the U.K., after your international access code use 44 1962 870229
 - From within the U.K., use 01962 870229
- Electronically, use the appropriate network ID:
 - IBM Mail Exchange: GBIBM2Q9 at IBMMAIL
 - IBMLink: WINVMD(IDRCF)
 - Internet: idrcf@winvmd.vnet.ibm.com

Whichever you use, ensure that you include:

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

Readers' Comments

MQSeries for Sun Solaris

Quick Beginnings

GC33-1870-01

Use this form to tell us what you think about this manual. If you have found errors in it, or if you want to express your opinion about it (such as organization, subject matter, appearance) or make suggestions for improvement, this is the form to use.

To request additional publications, or to ask questions or make comments about the functions of IBM products or systems, you should talk to your IBM representative or to your IBM authorized remarketer. This form is provided for comments about the information in this manual and the way it is presented.

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

Be sure to print your name and address below if you would like a reply.

Name

Address

Company or Organization

Telephone

Email

MQSeries for Sun Solaris

MQSeries for Sun Solaris V5.0 Quick Beginnings GC33-1870-01



You can send your comments POST FREE on this form from any one of these countries:

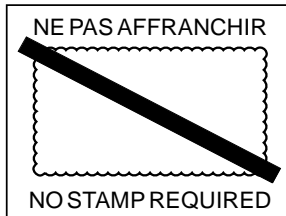
Australia	Finland	Iceland	Netherlands	Singapore	United States
Belgium	France	Israel	New Zealand	Spain	of America
Bermuda	Germany	Italy	Norway	Sweden	
Cyprus	Greece	Luxembourg	Portugal	Switzerland	
Denmark	Hong Kong	Monaco	Republic of Ireland	United Arab Emirates	

If your country is not listed here, your local IBM representative will be pleased to forward your comments to us. Or you can pay the postage and send the form direct to IBM (this includes mailing in the U.K.).

2 Fold along this line

By air mail
Par avion

IBRS/CCRI NUMBER: PHQ - D/1348/SO



REPONSE PAYEE
GRANDE-BRETAGNE

IBM United Kingdom Laboratories
Information Development Department (MP095)
Hursley Park,
WINCHESTER, Hants
SO21 2ZZ United Kingdom

3 Fold along this line

From: Name _____
Company or Organization _____
Address _____

EMAIL _____
Telephone _____

4 Fasten here with adhesive tape

1 Cut along this line

1 Cut along this line





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

GC33-1870-01

