

MQSeries for Windows NT**

GC33-1871-00

Quick Beginnings

Version 5.0



MQSeries for Windows NT**

GC33-1871-00

Quick Beginnings

Version 5.0

Note

Before using this document, read the general information under Appendix E, "Notices" on page 117.

First Edition (September 1997)

This edition applies to Version 5.0 of *MQSeries for Windows NT* 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 1995,1997. 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 Windows NT	vii
Road Map	vii
Conventions	viii

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 Windows NT with Other Software Servers	8
Chapter 2. More about MQSeries for Windows NT	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 Windows NT 17

Chapter 3. Planning to Install MQSeries for Windows NT	19
Hardware Requirements	19
Software Requirements	19
MQSeries for Windows NT Components	21
README file	22
Chapter 4. Installing MQSeries for Windows NT	23
Installing the Server	23
Installing the Server and Client on the Same Machine	28
Setting Up Communications	28

Contents

Verifying the Installation of MQSeries for Windows NT	39
MQSeries Utilities	45
Chapter 5. Planning to Install MQSeries Clients	47
AIX Client: Hardware and Software Required	47
Components for AIX	49
DOS Client: Hardware and Software Required	51
Components for DOS	52
HP-UX Client: Hardware and Software Required	53
Components for HP-UX	54
OS/2 Warp Client: Hardware and Software Required	55
Components for OS/2 Warp	56
Sun Solaris Client: Hardware and Software Required	58
Components for Sun Solaris	59
Windows 3.1 Client: Hardware and Software Required	60
Components for Windows 3.1	61
Windows 95 Client: Hardware and Software Required	62
Components for Windows 95	63
Windows NT Client: Hardware and Software Required	64
Components for Windows NT	65
Chapter 6. Installing MQSeries Clients	67
Installing on AIX	67
Installing on DOS	69
Installing on HP-UX	70
Installing on OS/2 Warp	71
Installing on Sun Solaris	74
Installing on Windows NT or Windows 95	76
Installing on Windows 3.1	77
Configuring and Verifying a Client	79
<hr/>	
Part 3. Using MQSeries for Windows NT	81
Chapter 7. Using the MQSeries Command Sets	83
Introducing Command Sets	83
Working with Queue Managers	87
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 Windows NT Library	99
Viewing Online Information	101
Printing and Ordering Books	102
Available from the Internet	103

Figures

Part 4. Appendixes	105
Appendix A. Unattended Installation	107
Automating Installation and Maintenance	107
Appendix B. Preparing to Install From a LAN	111
Appendix C. Removing MQSeries for Windows NT	113
Appendix D. Installing the MQSeries Product License	115
Enrolling the Trial-Period License	115
Enrolling the Product License	115
Appendix E. Notices	117
Trademarks	118
Index	119

Figures

1. Select Destination Directories window	25
2. Choose MQSeries Components window	26
3. The SNA Node Configuration window	29
4. Define the Node	30
5. Define a Local LU 6.2	31
6. Define a LAN Device	32
7. Define a LAN Connection (Basic page)	33
8. Define a LAN Connection (Advanced page)	34
9. Define a LAN Connection (Security page)	35
10. Define a Partner LU 6.2	36
11. Define CPI-C Side Information	37
12. Define a Transaction Program	38

Tables

Tables

- 1. Getting Started Road Map vii
- 2. MQSeries publications – file names 102

Welcome to MQSeries for Windows NT

This book describes MQSeries for Windows NT 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 Windows NT.

Part 2 provides planning and installing information for MQSeries for Windows NT. Information is provided for both server and client components.

Part 3 explains how to use MQSeries for Windows NT.

Road Map

Use Table 1 to find the information you need to get started.

<i>Table 1. 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 MQSeries for Windows NT	Chapter 2, "More about MQSeries for Windows NT" on page 9
Learn about system requirements for MQSeries for Windows NT	Chapter 3, "Planning to Install MQSeries for Windows NT" on page 19
Install MQSeries for Windows NT	Chapter 4, "Installing MQSeries for Windows NT" on page 23
Learn about system requirements for clients	Chapter 5, "Planning to Install MQSeries Clients" on page 47
Install an MQSeries client	Chapter 6, "Installing MQSeries Clients" on page 67
Start using command sets	Chapter 7, "Using the MQSeries Command Sets" on page 83
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
Contact IBM	<i>Readers' Comment Form</i>

Conventions

Conventions

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

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

Part 1. The IBM Software Servers

This part of the book introduces IBM software servers and explains the benefits of using them. It also includes a brief overview of the major features of MQSeries for Windows NT. It includes the following chapters:

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

Chapter 1. Introducing IBM's Software Servers

When you choose the newest version of MQSeries for Windows NT, 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.

Directory and Security Server

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.

Lotus Domino

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:

- 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.

Combining Software Servers

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

Combining MQSeries for Windows NT with Other Software Servers

When you combine MQSeries for Windows NT 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 Windows NT, which provides flexible, reliable communications for your network. Take advantage of your easy-to-use Windows- and DOS-based 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 Windows NT can be installed with currently supported releases of Windows NT. (See "Software Requirements" on page 19 for a complete list of software requirements.)

To learn how to install MQSeries for Windows NT, refer to "Installing the Server" on page 23.

Chapter 2. More about MQSeries for Windows NT

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.

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

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 Windows NT

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

- Chapter 3, "Planning to Install MQSeries for Windows NT" on page 19
- Chapter 4, "Installing MQSeries for Windows NT" on page 23
- Chapter 5, "Planning to Install MQSeries Clients" on page 47
- Chapter 6, "Installing MQSeries Clients" on page 67

Chapter 3. Planning to Install MQSeries for Windows NT

This chapter is a summary of the requirements to run MQSeries for Windows NT, the network protocols and the compilers supported, the delivery media, and the various components of the product. The following information applies to the server environment only. For information about client environments, see Chapter 5, "Planning to Install MQSeries Clients" on page 47.

Hardware Requirements

The following are the hardware requirements for MQSeries for Windows NT:

- Any Intel 486 (or above) processor-based IBM PC machine or compatible
- Any communications hardware supporting SNA LU 6.2, TCP/IP, NetBIOS, or SPX

Disk Space Required

MQSeries requires a minimum of 10 megabytes (MB) of disk space for product code and data. In addition, allow a minimum of 20 MB for working space.

Software Requirements

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

- Microsoft Windows NT Version 3.51 with service pack 5 applied to it, or NT Version 4.0, including TCP/IP, NetBIOS, and SPX.
- Connectivity
 - IBM Communications Server for Windows NT, Version 5.0
 - IBM Personal Communications for Windows NT, Version 4.1
 - Attachmate Extra! Personal Client, Version 6.1 or Version 6.2
 - Microsoft SNA Server, Version 2.11 or Version 3.0
 - TCP/IP, NetBIOS, and SPX are part of the base operating system

Software Requirements

Software Supported

The following are options, not prerequisites. Minimum supported levels are shown. Later levels, if any, will be supported unless otherwise stated.

- Transaction Processing Monitors
 - IBM Transaction Server for Windows NT, Version 4.0
 - BEA TUXEDO Version 5.1 and Version 6.1
- Lotus Notes Version 4.1
- Databases
 - IBM DB2 for Windows NT and Windows 95, Version 2.11
- IBM Software Servers
 - Communications Server for Windows NT, Version 5.0
 - Database Server for Windows NT, Version 4.0
 - Directory Security Server for Windows NT
 - Internet Connection Server for Windows NT, Version 4.1
 - Internet Connection Secure Server for Windows NT, Version 4.1
 - Transaction Server for Windows NT, Version 4.0

DCE Support

The IBM Directory and Security Server for Windows NT, Version 5.0 (the variant that supports DES encryption).

MQSeries DCE security modules are provided as part of the MQSeries for Windows NT product.

Compilers Supported

The following software compilers are supported:

- C and C++ using IBM VisualAge C++ for Windows, Version 3.5
- C and C++ using Microsoft Visual C++ for Windows 95 and NT, Version 4.0
- IBM VisualAge COBOL for Windows NT, Version 2.1
- COBOL using Micro Focus COBOL for Windows NT, Version 3.3 or Version 4.0
- COBOL using Micro Focus COBOL Version 3.1J for Windows NT
- IBM PL/I for Windows Version 1.2
- IBM VisualAge for PL/I for Windows

Note: For CICS Version 2, MQSeries C applications can support only the Microsoft Visual C++ compiler, and not the IBM VisualAge C++ compiler.

MQSeries for Windows NT Components

When you install MQSeries for Windows NT you can select which components you install. The various components are:

MQSeries Base Server for Windows NT

The MQSeries for Windows NT base code.

MQSeries Clients

MQSeries Client for Windows NT and MQSeries client for Java:

- The MQSeries Client for Windows NT 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 Java interfaces for send, receive, and security exits allowing you to provide encryption and authentication of messages sent by the MQSeries Client for Java.

MQSeries Toolkit

Sample MQSeries program source code in C and COBOL, including header files, link libraries, and source files for sample applications. These are useful when starting MQSeries application program development.

MQSeries Utilities

A set of Java-based utilities. These utilities should be installed only if you have Java 1.1.1 (or later) runtime code on your machine. Included are:

- RAS tools (to assist with serviceability and problem determination).
- MQSeries installation verification. This feature helps you to verify your installation quickly and easily.

MQSeries Online Documentation

Online versions of the books for MQSeries for Windows NT in HTML formats. Included are:

- MQSeries Documentation
- MQSeries Internet Documentation

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

MQSeries Web Administration

Allows you to administer your MQSeries installation through a web browser.

MQSeries Bindings for Java

A set of executables allowing an application programmer to use Java to communicate with MQSeries.

MQSeries Internet Gateway

Provides access to MQSeries applications via HTML and CGI.

README

README file

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

The README file contains any product and documentation updates made after this book was printed.

Chapter 4. Installing MQSeries for Windows NT

This chapter tells you how to install the MQSeries for Windows NT server.

Note: 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 D, "Installing the MQSeries Product License" on page 115 (which tells you how to enroll the product license in these situations).

For information on installing clients, see Chapter 6, "Installing MQSeries Clients" on page 67.

Installing the Server

To install MQSeries for Windows NT, you must be logged on to Windows NT Server as an administrator. You can run the installation either directly from the CD-ROM or across a LAN. This chapter describes how to install directly from the CD-ROM; for information on how to install across a LAN, see Appendix B, "Preparing to Install From a LAN" on page 111.

If you have previously installed MQSeries for Windows NT, ensure that no queue managers are running and that the IBM MQSeries Service is stopped.

A check is made for an existing MQSeries configuration file (MQS.INI). If one is found, you will be given the option of using it with the newly installed system, otherwise a new version is created. For information about this configuration file, its contents and location, see the *MQSeries System Administration* book.

The following procedure for installing MQSeries for Windows NT assumes that the CD-ROM drive is E:\ and that the product will be installed to the C drive.

Note: When an instruction tells you to *enter* a command or piece of text, type the command or text and then either press the Enter key or select the **OK** push button.

When you are instructed to *select* an object (for example, a push button or an item in a list), either highlight the object and then press the Enter key, or double-click on the object with the left mouse button.

Installing the Server

Installation Procedure

1. Insert the MQSeries for Windows NT CD-ROM into the CD-ROM drive.

The MQSeries Language Selection window is displayed, in which you choose which national language you want to install. Supported languages are:

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

Select the language of your choice, and then select the **Next** push button.

2. The MQSeries Welcome window is displayed, introducing the installation process.

Note: If the setup process has detected that the IBM MQSeries Service is currently running, it will ask you if you want to stop the service or cancel the installation. You cannot install MQSeries over an installation that is currently running. If Version 2.0 of IBM MQSeries is detected, the setup process must remove it before the installation can be started.

3. The Select Destination Directories window is displayed, which allows you to specify the directories into which the MQSeries executables and data files are to be installed. (See Figure 1 on page 25.)

The default is the C drive, but you can change this by selecting the browse button and choosing a different drive. The destination locations for executables and data files do not need to be the same.

Note: You can specify the destination directories only if this is a new installation of MQSeries for Windows NT. If MQSeries has been installed previously and you are now adding new components, the destination directories specified during the previous installation are fixed and cannot be changed during this installation procedure.

Installing the Server

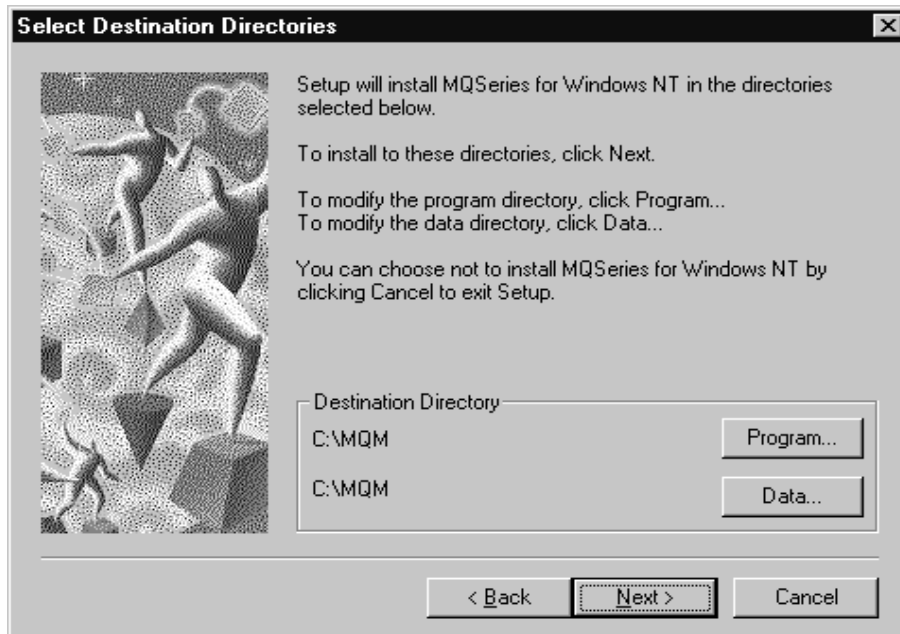


Figure 1. Select Destination Directories window

When you have selected the destination directories, select the **Next** button or press Enter.

4. The Choose MQSeries Components window is displayed, showing a list of components that can be installed. (See Figure 2 on page 26.) For a complete list of components that can be installed, see "MQSeries for Windows NT Components" on page 21.

Installing the Server

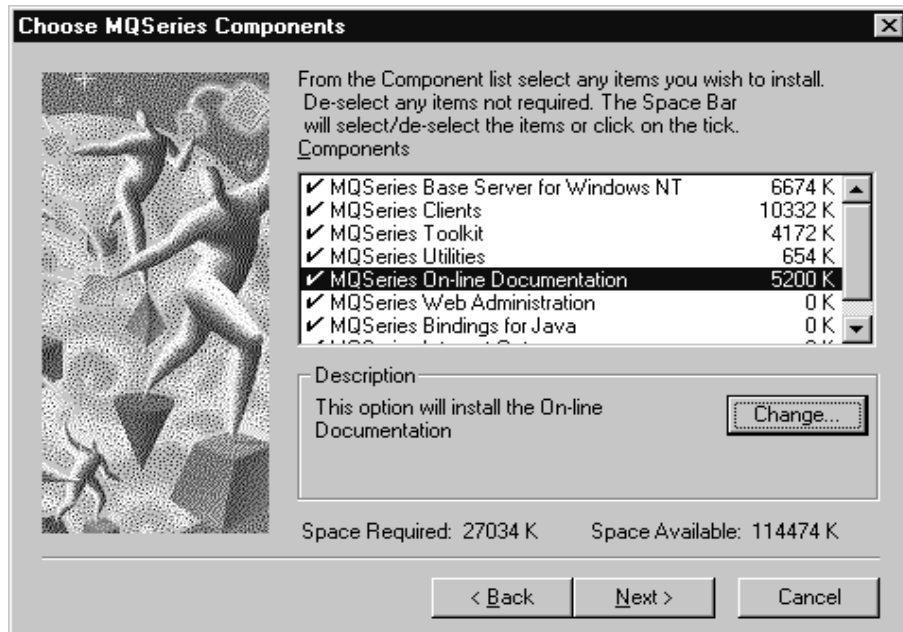


Figure 2. Choose MQSeries Components window

The MQSeries Clients and MQSeries Online Documentation components contain individually selectable subcomponents. To gain access to these, highlight the 'parent' component (either MQSeries Clients or MQSeries Online Documentation) and then select the **Change** button.

When you install MQSeries for the first time, all of the components are selected for installation. To deselect any of the components, click on the tick-mark preceding it. If you are adding new components to a previously installed version of MQSeries, the previously installed components are not selected automatically.

When you have selected the required components, select the **Next** button or press Enter.

5. The next window, Select Program Folder, prompts you for a folder name to contain MQSeries objects. The default name is IBM MQSeries for Windows NT. Choose a name and then select the **Next** button or press Enter.
6. The next window, Start Copying Files, summarizes all of the selections made so far. If any amendments are necessary, you can use the **Back** button to return to the relevant window and make any changes. Otherwise, select the **Next** button or press Enter.

Note: If you are upgrading from Version 2.0 of MQSeries, the previous version of MQSeries will be removed at this point.

7. The next window, the Progress Indicator, indicates which component is currently being copied, and the amount completed so far (indicated by the Percent Complete indicator).

Installing the Server

8. If you chose to install either the MQSeries Server or the MQSeries Client for Windows NT, a panel called MQS.INI Options is displayed. This window allows you to choose whether the MQS.INI file is overwritten. If you are adding to an existing installation and do not want to lose your current queue manager settings, ensure that the **Overwrite the MQS.INI** check box is not selected. Whether you upgrade or replace an existing copy of MQS.INI, the previous version will be renamed (as MQS.ORI).

When you have finished with this window, select the **Next** button or press Enter.

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 program folder chosen earlier in the installation (the default was the IBM MQSeries for Windows NT folder). If you do choose to view the README file at this point, the installation will not continue until the Notepad application used to view the file is dismissed.

Select the **Next** button or press Enter to continue.

10. The installation of MQSeries for Windows NT is now complete, and a window is displayed giving you the option of restarting your computer now or leaving it until later. The default selection is for you to restart your computer later. If you choose to restart your computer now, it is recommended that you minimize the Setup application and save all your current work. At the same time, end any applications that you may have running.

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

11. If you are running Windows NT Version 3.5 or 3.51, an MQSeries folder is added to your desktop when the setup is complete. If you are running Windows NT Version 4.0, the folder is added to the **Programs** menu on the Taskbar.
12. If you are upgrading from MQSeries for Windows NT Version 2.0, the setup process attempts to upgrade your code page conversion table (CCSID.TBL) to the new format required by Version 5.0. If any problems are encountered during this process, a message is written to the application event log. Therefore it is advisable to check in the application event log after performing an upgrade to determine if any further actions are required.

Installation is now complete. You can now go on to set up your communications as shown in "Setting Up Communications" on page 28, and then verify the installation as shown in "Verifying the Installation of MQSeries for Windows NT" on page 39.

Registration

When you have finished installing MQSeries for Windows NT, please register your use of the product by selecting the **MQSeries Registration Tool** icon in the MQSeries for Windows NT folder, and following the instructions.

Setting up Communications

Installing the Server and Client on the Same Machine

To install an MQSeries for Windows NT 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.

Setting Up Communications

If MQSeries for Windows NT is to exchange messages with MQSeries on another machine, communications must be configured between the two. You will not be able to run a server-to-server verification process until cross-server communication has been enabled. MQSeries for Windows NT supports the TCP/IP, SNA LU 6.2, NetBIOS, and SPX protocols. The *MQSeries Intercommunication* book describes how to configure communications from Windows NT to other MQSeries platforms, and provides many example configurations. In “Establishing an LU 6.2 Connection,” there is an example SNA LU 6.2 configuration that uses IBM Communications Server for Windows NT, V5.0.

Establishing an LU 6.2 Connection

This section shows how to establish an LU 6.2 connection using IBM Communications Server for Windows NT, Version 5.0. You may use any of the supported LU 6.2 products for this platform. The panels of other products will not be identical to those shown here, but most of their content will be similar.

Setting up Communications

Some of the values you are required to supply are assigned by a network administrator. You are recommended to obtain those values before you start the SNA configuration.

Configure the Local Node

To configure the local node, follow these steps:

1. From the **Scenarios** pull-down of the Communications Server SNA Node Configuration window, select the **CPI-C, APPC or 5250 Emulation** scenario, as shown in Figure 3.

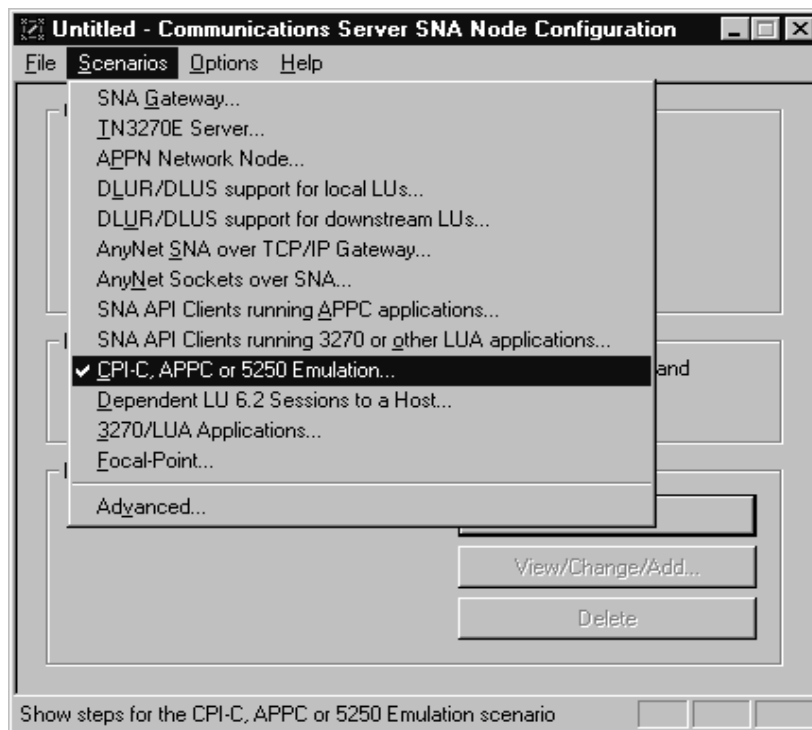


Figure 3. The SNA Node Configuration window

The CPI-C, APPC or 5250 Emulation scenario window is displayed.

2. Click on **Configure Node**, then click on **New**. The Define the Node property sheet, shown in Figure 4 on page 30, is displayed.

Setting up Communications

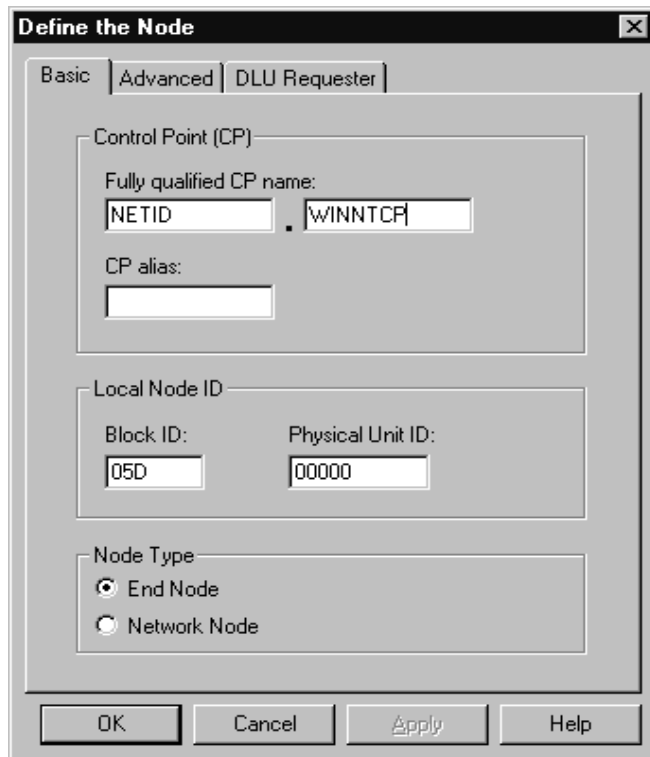


Figure 4. Define the Node

3. In the Fully qualified CP name field on the Basic page, enter the unique ID of the network to which you are connected (NETID in Figure 4) and the control point name (WINNTCP in Figure 4). Both values are typically assigned by the network administrator. Click on **OK** to continue.
4. From the SNA Node Configuration window, click on **Configure Local LU 6.2**, then click on **New**. The Define a Local LU 6.2 window, as shown in Figure 5 on page 31, is displayed.

Setting up Communications



Figure 5. Define a Local LU 6.2

5. In the Local LU name field on the Basic page, enter the name of the LU on your workstation (WINNTLU in Figure 5). This value is typically assigned by a network administrator. In the Local LU alias field, enter the name by which your local LU will be known to your applications (NTQMGR in Figure 5). You assign this value. Click on **OK** to continue.

Adding a Connection

To add a connection, follow these steps:

1. From the SNA Node Configuration window, select **Configure Devices**, select **LAN** as the DLC type, then click on **New**. The Define a LAN Device property sheet, as shown in Figure 6 on page 32, is displayed.

Setting up Communications

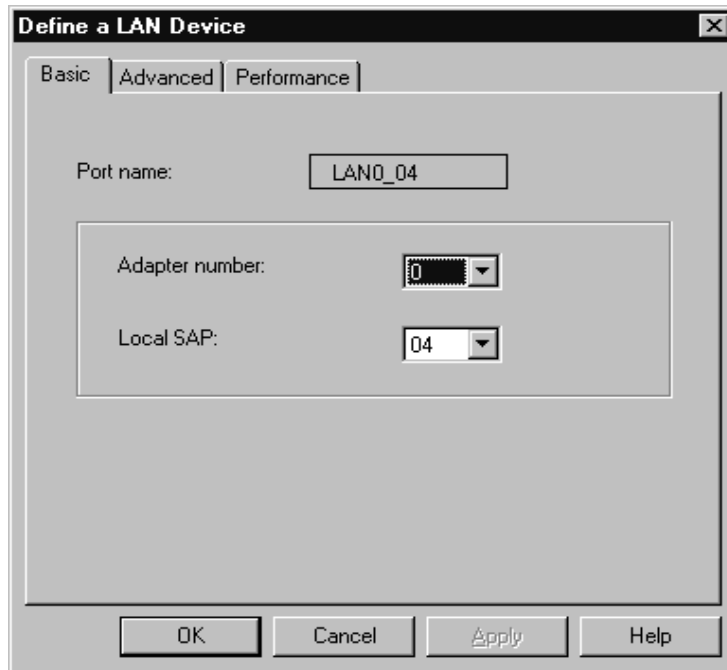


Figure 6. Define a LAN Device

2. If you have the LLC2 protocol installed with Communications Server for Windows NT, the Adapter number list box lists the available LAN adapters. See the help file INLLC40.HLP (Windows NT 4.0) or INLLC35.HLP (Windows NT 3.51) in the Communications Server installation directory for LLC2 installation instructions.
3. The default values displayed on the Define a LAN Device Basic page may be accepted. Click on **OK** to continue.
4. From the SNA Node Configuration window, select **Configure Connections**, select **LAN** as the DLC type, then click on **New**. The Define a LAN Connection property sheet, as shown in Figure 7 on page 33, is displayed.

Setting up Communications



Figure 7. Define a LAN Connection (Basic page)

5. In the Destination address field on the Basic page, enter the LAN address of the system to which you are connecting. Select the Advanced page.

Setting up Communications

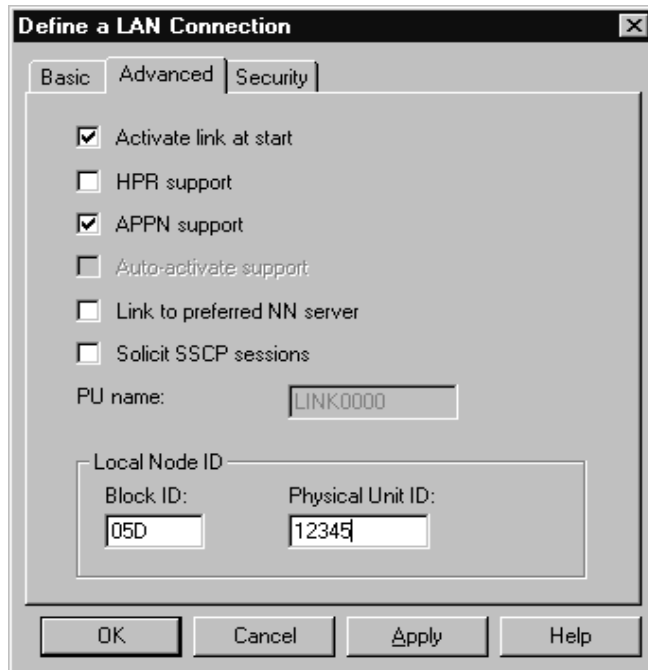


Figure 8. Define a LAN Connection (Advanced page)

6. In the Block ID field on the Advanced page, enter the local node ID (hex). This value is assigned by the network administrator. Select the Security page.

Setting up Communications

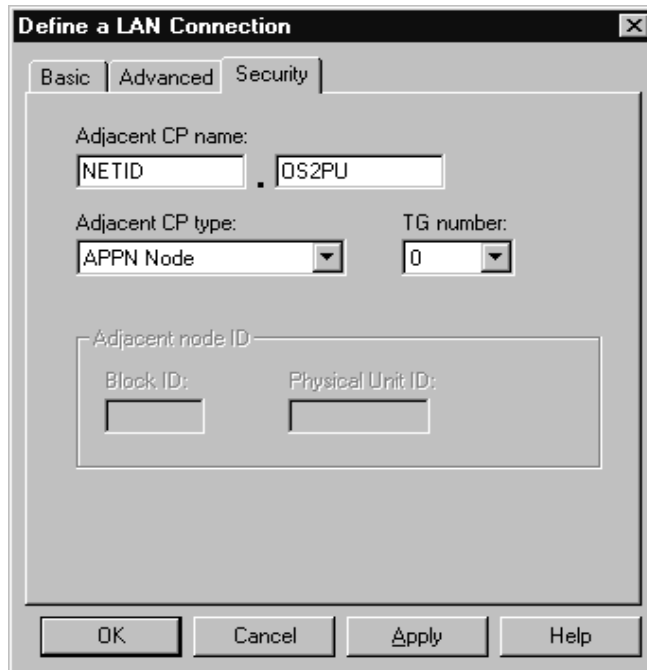


Figure 9. Define a LAN Connection (Security page)

7. In the Adjacent CP name field on the Security page, enter the network name and control point name of the remote node (NETID and OS2PU in Figure 9). In the Adjacent CP type field, enter APPN Node. You do not need to complete the Adjacent node ID field for a peer-to-peer connection. Click on **OK** to continue. Take note of the default link name used to identify this new definition (for example, LINK0000).

Setting up Communications

Adding a Partner

To add a partner LU definition, follow these steps:

1. From the SNA Node Configuration window, select **Configure Partner LU 6.2**, then click on **New**. The Define a Partner LU 6.2 property sheet, as shown in Figure 10, is displayed.

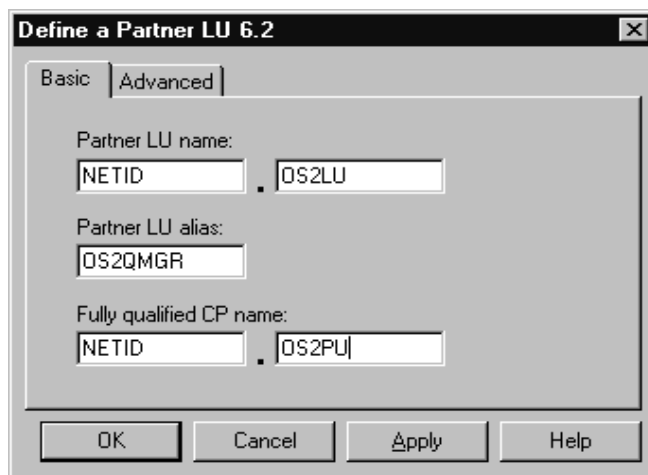


Figure 10. Define a Partner LU 6.2

2. In the Partner LU name field on the Basic page, enter the network name and LU name of the remote system (NETID and OS2LU in Figure 10). In the Partner LU alias field, enter the remote LU alias (OS2QMGR in Figure 10). In the Fully qualified CP name fields, enter the network name and control point name of the remote system (NETID and OS2PU in Figure 10). Click on **OK** to continue.

Setting up Communications

Adding a CPI-C Entry

To add a CPI-C Side information entry, follow these steps:

1. From the SNA Node Configuration window, select **Configure CPI-C Side Information**, then click on **New**. The Define a CPI-C Side Information property sheet, as shown in Figure 11, is displayed.

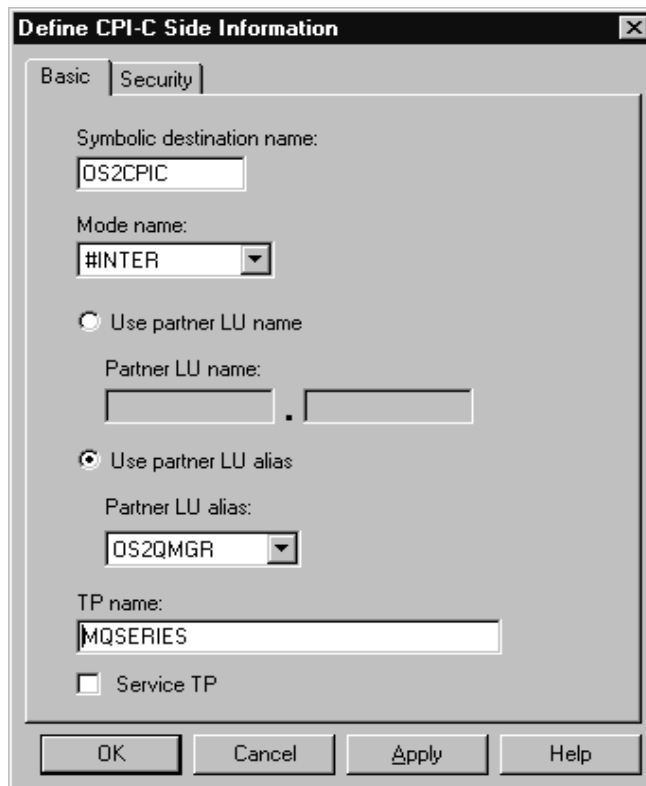


Figure 11. Define CPI-C Side Information

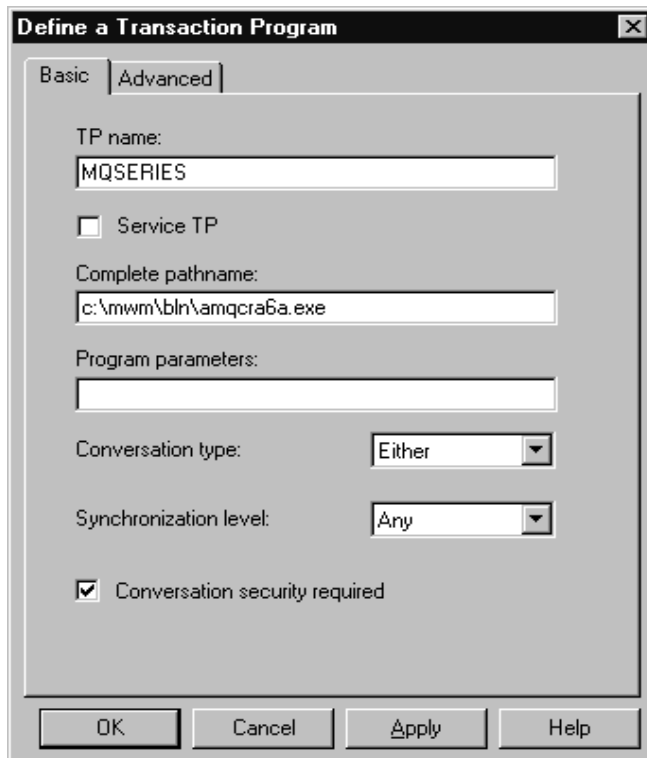
2. In the Symbolic destination name field of the Basic page, enter the CPI-C name (OS2CPIC in Figure 11). In the Mode name field, enter the mode value (#INTER in Figure 11). The mode value is assigned by the network administrator. Enter either a fully qualified partner LU name (network name plus LU name) or a partner LU alias (OS2QMGR in Figure 11). In the TP name field, enter the partner TP name (MQSERIES in Figure 11). Click on **OK** to continue.

Setting up Communications

Configuring an Invokable TP

To add a Transaction Program (TP) definition, follow these steps:

1. From the SNA Node Configuration window, select **Configure Transaction Programs**, then click on **New**. The Define a Transaction Program property sheet, as shown in Figure 12, is displayed.



The image shows a dialog box titled "Define a Transaction Program" with a close button (X) in the top right corner. It has two tabs: "Basic" (selected) and "Advanced". The "Basic" tab contains the following fields and controls:

- TP name: A text box containing "MQSERIES".
- Service TP: An unchecked checkbox.
- Complete pathname: A text box containing "c:\mwm\bin\amqcra6a.exe".
- Program parameters: An empty text box.
- Conversation type: A dropdown menu set to "Either".
- Synchronization level: A dropdown menu set to "Any".
- Conversation security required: A checked checkbox.

At the bottom of the dialog box are four buttons: "OK", "Cancel", "Apply", and "Help".

Figure 12. Define a Transaction Program

2. In the TP name field on the Basic page, enter the transaction program name (MQSERIES in Figure 12). In the Complete pathname field, enter the actual path and name of the the program that will be run when a conversation is initiated with your workstation. Click on **OK** to continue.

What Next?

The SNA configuration task is complete. From the **File** pull-down, select **Save** and specify a file name under which to save your SNA configuration information. When prompted, select this configuration as the default.

Verifying the Installation

From the SNA Node Operations application, start the node by clicking the **Start node** button on the toolbar. Specify the file name of the configuration you just saved. (It should appear in the file-name box by default, because you identified it as your default configuration.) When the node startup is complete, ensure that your link to the remote node has been established by selecting the **Connections** button on the toolbar, then find the link name you configured (for example, LINK0000). The link should be active if the remote node is active waiting for the link to be established.

A complementary SNA setup process is required on the node to which you are connecting before you can attempt MQSeries server-to-server message transmissions.

You are now ready to proceed with MQSeries for Windows NT verification.

Verifying the Installation of MQSeries for Windows NT

This section describes how to verify that MQSeries for Windows NT has been correctly installed and configured. You can do the verification in two ways:

- By following the steps outlined in “Verification Procedure” on page 40
- If you have Java 1.1.1 or later, you can run the Installation Verification wizard, described in “Using the Installation Verification Wizard” on page 45

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

The following protocols are supported:

- LU 6.2
- NetBIOS
- SPX
- TCP/IP

Note: The following examples assume that you will be using a SNA (LU 6.2) connection; for information about using other protocols, see the *MQSeries Intercommunication* manual. An example of how to set up communications using SNA is shown in “Setting Up Communications” on page 28.

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

Verifying the Installation

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 are described in “Verifying a Local Installation” below, and in “Verifying a Server-to-Server Installation” on page 42. 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 Windows NT on the workstation (include the Base Server component as a minimum)
2. Open a command prompt window
3. 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.
4. Start the default queue manager:
 - Type the following and then press Enter:

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

Verifying the Installation

5. Enable MQSC 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.

6. 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* and a queue called *ORANGE.QUEUE*.

7. Stop MQSC by pressing Ctrl-Z, 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.
```

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

```
amqsput ORANGE.QUEUE
```

2. The following message is displayed:

```
Sample AMQSPUT0 start  
target queue is ORANGE.QUEUE
```

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

4. The following message is displayed:

```
Sample AMQSPUT0 end
```

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

Verifying the Installation

5. To get the message from the queue, using the sample AMQSGET, 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)
```

Note: Any text entered in MQSC in lowercase is converted automatically to uppercase unless you enclose it in single quotation marks.

- 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) trptype (lu62) +  
conname (remote_sna_hostname) xmitq (transmit1.queue)
```

where *remote_sna_hostname* is the SNA address of the receiver workstation (note that this example is LU6.2 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-Z, or typing end, and pressing Enter.

Now set up the receiver workstation.

Receiver Workstation

1. Open a command prompt window
2. 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.

Verifying the Installation

3. Start the queue manager:

- Type the following and then press Enter:

```
strmqm
```

- A message tells you when the queue manager has started

4. 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.

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.

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

6. Create a receiver channel:

```
define channel (first.channel) chltype (rcvr) trptype (lu62)
```

You have now defined a default queue manager called *venus.queue.manager*, a queue called *ORANGE.QUEUE*, and a receiver channel called *FIRST.CHANNEL*.

7. Stop MQSC by pressing Ctrl-Z 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 **Receiver** workstation start the listener program:

```
start runmqlsr -t lu62 -m venus.queue.manager
```

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

```
start runmqchl -c first.channel -m saturn.queue.manager
```

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

4. 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.

5. Type the text of the message and press Enter twice.

MQSeries Utilities

6. On the **Receiver** workstation, use the AMQSGET sample program to get the message from the queue:

```
AMQSGET ORANGE.QUEUE
```
7. The message is displayed.
The verification is complete.

MQSeries Utilities

The IBM MQSeries Utilities should be installed only if you have the Java 1.1.1 (or later) runtime code. The utilities include an installation verification wizard and a set of RAS tools.

Using the Installation Verification Wizard

If you have Java 1.1.1 (or later) runtime code, you can use the installation verification wizard, which is installed as part of the MQSeries Utilities component.

The wizard gives you the option of verifying any or all of the following installations:

Local	A standalone (server) installation with no communications links to other MQSeries machines
Server-Server	An MQSeries server that will communicate with other MQSeries servers
Client-Server	An MQSeries server that will communicate with an MQSeries client
Client	An MQSeries client that will communicate with an MQSeries server

Running the Installation Verification Wizard

To start the wizard, select the MQSeries Utilities folder from the MQSeries folder, and then double-click on the Installation Verification wizard icon. Then follow the instructions presented by the wizard.

MQSeries Utilities

Running the Service Tools

This is a set of tools for IBM reliability, availability, and serviceability. These are provided to assist with serviceability and problem determination.

To start the tool, select the MQSeries Utilities folder from the MQSeries folder, and then double-click on the individual icon. Then follow the instructions presented by the wizard.

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 the above 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.

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.

Hardware and Software, AIX

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:

- 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

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.

Components for AIX

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) runtime code on your machine. 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 Utilities

A set of Java-based utilities. To use the utilities you need to have Java 1.1.1 (or later) runtime code on your machine. For information about Java runtime see "Available from the Internet" on page 103.

Included are:

- RAS tools. Tools to assist with serviceability and problem determination.
- MQSeries installation verification. This feature helps you to verify your installation quickly and easily.

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.

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 OS/400 and MQSeries for MVS/ESA). 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

Components for DOS

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

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.

Components for HP-UX

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.

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 49.

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

MQI 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.

Components for OS/2 Warp

- If used as a DCE server this software is known to run adequately in the following environment:
 - On a Pentium** processor running 90 MHz or faster.
 - On a machine with 64 MB or more of memory.
 - Using OS/2 Warp Server V4.0 or later.
- MQSeries DCE names and security modules are provided as part of the MQSeries for OS/2 Warp product.

Compilers for MQSeries Applications on OS/2 Clients

The following 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

Components for OS/2 Warp

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) runtime code on your machine. 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 Utilities

A set of Java-based utilities. To use the utilities you need to have Java 1.1.1 (or later) runtime code on your machine. For information about Java runtime see "Available from the Internet" on page 103.

Included are:

- RAS tools. Tools to assist with serviceability and problem determination.
- MQSeries installation verification. This feature helps you to verify your installation quickly and easily.

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. An additional 25 MB of disk space is required if DynaText is to be installed.

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 49.

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 OS/400 and MQSeries for MVS/ESA). It operates under:

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

Components for Windows 3.1

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.
- 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 52.

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

For TCP/IP connectivity:

- TCP/IP (in the operating system)

For SPX connectivity:

- SPX (in the operating system)

For NetBIOS connectivity:

- 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).

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

MQSeries Utilities

A set of Java-based utilities. To use the utilities you need to have Java 1.1.1 (or later) runtime code on your machine. For information about Java runtime see "Available from the Internet" on page 103.

Included are:

- RAS tools. Tools to assist with serviceability and problem determination.
- MQSeries installation verification. This feature helps you to verify your installation quickly and easily.

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 56.

Components for Windows NT

Chapter 6. Installing MQSeries Clients

This chapter describes the following:

- “Installing on AIX”
- “Installing on DOS” on page 69
- “Installing on HP-UX” on page 70
- “Installing on OS/2 Warp” on page 71
- “Installing on Sun Solaris” on page 74
- “Installing on Windows NT or Windows 95” on page 76
- “Installing on Windows 3.1” on page 77

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 28.

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`
- `/var/mqm/objects`

You can choose to NFS mount the `/var/mqm/errors` and `/var/mqm/trace` directories 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 49 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 Install

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 drive where you want to install the MQSeries client, for example C:\.

Installing on HP-UX

4. Change to C:\. and type the command:

```
setup
```

All the files required for the MQSeries client are placed in the directory c:\mqmdos. See "Components for DOS" on page 52 for details.

5. Open the file README.TXT in the c:\mqmdos directory and follow the instructions in the file to complete your installation.

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.

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**
- **/var/mqm/objects**

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

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

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

Installation

Use the HP-UX **swinstall** program, or use SAM, after mounting the CD-ROM. For further details see the *HP-UX Administration Guide*. See also “Components for AIX” on page 49.

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

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 28.

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.

Installing on OS/2 Warp

Installation

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

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

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

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

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

See "Components for OS/2 Warp" on page 56 for details.

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).

Installing on OS/2 Warp

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

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

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

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

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

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

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

```
READ.ME
MQSeries Installation and Maintenance
MQSeries Utilities
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.)

Installing on Sun Solaris

Unattended Installation

It is possible to install MQSeries clients using an unattended installation method. For details of this see the book *MQSeries for OS/2 Warp V5.0 Quick Beginnings*.

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 28.

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`
- `/var/mqm/objects`

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
```

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 Sun Solaris **pkgadd** program, to install the MQSeries client software by carrying out the following procedure:
 - a. Type `pkgadd -d /cdrom/solaris/mqm/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 49 for details.
 - 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 28.

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 56 or “Components for Windows 95” on page 63 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.
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.

Installing on Windows 3.1

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".

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. Run setup.exe
4. 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.

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

Installing on Windows 3.1

6. The next window, 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. The next window, 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 52 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. The next window, 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. The next window, 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 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.

Configuring a Client

Part 3. Using MQSeries for Windows NT

This part of the book shows you how to use MQSeries for Windows NT. It contains the following chapters:

- Chapter 7, "Using the MQSeries Command Sets" on page 83
- 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 the MQSeries for Windows NT environment, you enter control commands at a command prompt. In this environment, control commands and their flags are not case sensitive, but arguments to those commands (such as queue names and queue-manager names) are case sensitive. For example, in the command:

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

- The command name can be entered in uppercase or lowercase, or a mixture of the two. These are all valid: `crtmqm`, `CRTMQM`, and `CRTmqm`.
- The flag can be entered as `-u`, `-U`, `/u`, or `/U`.
- The arguments `SYSTEM.DEAD.LETTER.QUEUE` and `jupiter.queue.manager` must be entered exactly as shown.

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.

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.

endmqtrc (end MQSeries trace)

Ends tracing for the specified entity or all entities.

MQSeries Command Sets

rcdmqing (record media image)

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

rcrmqobj (recreate object)

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

rsvmqtrn (resolve MQSeries transactions)

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

runmqchi (run channel initiator)

Runs a channel initiator process.

runmqchl (run channel)

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

runmqdlq (run dead-letter queue handler)

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

runmqlsr (run listener)

Runs a listener process.

runmqsc (run MQSeries commands)

Issues MQSC commands to a queue manager.

runmqtrm (start trigger monitor)

Invokes a trigger monitor.

scmmqm (add the queue manager to the Windows NT Service Control Manager)

Adds a queue manager to, or deletes a queue manager from, the list of those that will be started automatically (by the IBM MQSeries service) when the system starts.

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 Command Sets

MQSeries (MQSC) Commands

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

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

For detailed information about each MQSC command, see the *MQSeries 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; and specifies the names of both a default transmission queue and a dead-letter queue. (For more information about these attributes, see the *MQSeries System Administration* manual.)

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

where:

<code>-q</code>	Indicates that this queue manager is the default queue manager.
<code>-d MY.DEFAULT.XMIT.QUEUE</code>	Is the name of the default transmission queue.
<code>-u SYSTEM.DEAD.LETTER.QUEUE</code>	Is the name of the dead-letter queue.
<code>saturn.queue.manager</code>	Is the name of this queue manager. This must be the last parameter specified on the crtmqm command.

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.

Working with Queue Managers

Starting a Queue Manager Automatically

In the Windows NT environment, a queue manager can be invoked automatically when the system starts. To request automatic startup of a queue manager, enter:

```
scmmqm -a -s c:\mqm\startup.cmd saturn.queue.manager
```

where:

-a	Specifies that the queue manager is to be added to the list of those that will be automatically started.
-s	References a command file, of any name and location, that describes what actions are taken when this queue manager starts.
saturn.queue.manager	Is the name of this queue manager. This parameter must be the final parameter specified on the scmmqm command.

The command file, referenced -s, is a text file containing a series of commands. The commands are executed in order, and must start in column one. Valid commands are:

- **runmqchi**
- **runmqchl**
- **runmqlsr**
- **runmqtmc**
- **runmqtrm**
- **strmqcsv**
- **strmqm**
- **tpstart** (SNA command)

The commands must be followed by their correct parameters and each queue manager included in the automatic startup list needs its own command file.

For example, the following command file

```
strmqm saturn.queue.manager  
runmqlsr -t TCP -m saturn.queue.manager -n server
```

starts a queue manager, and starts a listener for that queue manager.

All queue managers in the automatic startup list are started and stopped by one service named IBM MQSeries. This service can be modified from the Control Panel or from the DOS prompt and it accepts two commands only: start and stop.

You can also prevent a queue manager from starting automatically. See "Removing a Queue Manager From the Automatic Startup List" on page 89 for more information.

Removing a Queue Manager From the Automatic Startup List

In order to prevent Windows NT from starting a queue manager automatically, use the following command:

```
scmmqm -d saturn.queue.manager
```

See the *MQSeries System Administration* manual for more information about the **scmmqm** command.

Stopping a Queue Manager

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

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

Quiesced Shutdown

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

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

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
```

Working with Queue Managers

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. In the Windows NT environment, the **dltmqm** command also removes a queue manager from the automatic startup list (described in “Starting a Queue Manager Automatically” on page 88).

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 87.

Using the MQSC Facility Interactively

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

- 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.)

Working with Objects

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 or CTRL+Z and then press Enter.

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.

Working with Objects

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

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

The following MQSC command does this:

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

Notes:

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

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.

Working with Objects

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);
```

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 92, we defined the queue ORANGE.LOCAL.QUEUE. Suppose, for example, you wanted to increase the maximum message length on this queue to 10 000 bytes.

- Using the ALTER command:

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

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

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

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

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

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

Clearing a Local Queue

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

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

You cannot clear a queue if:

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

Working with Objects

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	c:\mqm\tools\c\samples\amqsbcg0.c
Executable	c:\mqm\tools\c\samples\bin\amqsbcg.exe

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/400, OS/2, Sun Solaris, HP-UX, or Windows NT, and supports the CGI, ICAPI, ISAPI, and NSAPI, Web server interfaces.

Obtaining More Information

The following documentation is accessible on the World Wide Web:

- *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. It is also available from the MQSeries World Wide Web site (see “Available from the Internet” on page 103).

The following components can be installed from the MQSeries CD-ROMs, and they are also currently available for downloading from the web site (see “Available from the Internet” on page 103):

- MQSeries Internet Gateway for OS/2
- MQSeries Internet Gateway for MVS OpenEdition
- MQSeries Internet Gateway for OS/400
- MQSeries Internet Gateway for Windows NT
- MQSeries Internet Gateway for AIX
- MQSeries Internet Gateway for HP-UX

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 Windows NT or related products.

The MQSeries for Windows NT Library

The MQSeries for Windows NT Server CD-ROM contains the following documentation for MQSeries for Windows NT.

Order Number	Title	Description
MQSeries Books		
GC33-1871	<i>Quick Beginnings</i>	Gives a brief overview of MQSeries for Windows NT, 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

To view the online help for a specific topic, select the Help Index and click on the item you require.

Viewing Online Books

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

BookManager Books

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

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

HTML Books on the World Wide Web

The books listed in "The MQSeries for Windows NT 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

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 Windows NT books are supplied in PostScript format on the MQSeries for Windows NT Server CD-ROM.

Table 2 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

Table 2. MQSeries publications – file names

Book	File Name
<i>MQSeries Planning Guide</i>	CSQZAB00
<i>MQSeries for Windows NT Quick Beginnings</i>	AMQEAC00
<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 Software Server home page is located at:

`http://www.software.ibm.com/is/sw-servers/mqseries/`

By following links from the MQSeries Software Server home page, you can:

- Learn more about the MQSeries Software Servers, including MQSeries for Windows NT**.
- Obtain useful information about MQSeries and latest technology.
- Access the MQSeries books in HTML format.

The MQSeries family home page is located at:

`http://www.software.ibm.com/mqseries/`

From the MQSeries family home page, you can obtain latest information about the whole MQSeries product family. MQSeries SupportPacs can be downloaded from this Web site.

Internet Availability

Part 4. Appendixes

Appendix A. Unattended Installation

This appendix describes how to perform unattended installation and maintenance of MQSeries for Windows NT.

Automating Installation and Maintenance

You can install or maintain MQSeries for Windows NT without interaction. This process is known as *unattended installation or maintenance* and uses response files. Response files are described in "Installation Response Files" on page 109.

For unattended installation and maintenance, you must first set up the server (see "Setting up the Server") and then install on the remote machine (see "Installing on the Remote Machine" on page 109).

Setting up the Server

To set up your server, follow these steps:

1. Copy the contents of the MQSeries for Windows NT CD-ROM to a local or LAN drive, for example:

```
XCOPY E:\*.* /E
```

2. Edit the supplied response file SETUP.ISS or create a new one.

To create a response file that can be used to install certain configurations, run the SETUP.EXE file with the -r switch and proceed to carry out an installation that will be copied by the unattended installation.

For example,

```
SETUP -r
```

Notes:

- a. When installation has ended, a response file will have been automatically generated for you that can be used as the basis for unattended installation.
- b. The response file, SETUP.ISS will be generated in the WINNT35 or WINNT directories, depending on the Version of Windows NT you are using.
- c. The SETUP command must be entered on a single line.

Unattended Installation

- d. You can enter the parameters in any order.
 - e. Values can be uppercase or lowercase.
 - f. If you enter the same parameter more than once, the value of the last entry in the parameter list applies.
3. Once the response file has been generated, you can copy it to the language dependent directories in the \IMAGE directory for your national language.

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

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

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

4. To run the Setup program in unattended mode from the command prompt, type the following:
- a. M: and press Enter
 - b. CD INSTMQS and press Enter
 - c. SETUP -s -l <language selection>

where language selection is:

- u – English
- d – German
- e – Spanish
- f – French
- i – Italian
- j – Japanese
- t – Traditional Chinese
- k – Korean
- p – Portuguese (Brazilian)
- c – Simplified Chinese

If you do not want to take a copy of the response file, SETUP.ISS from the Windows NT directory, then you can refer to the response file by using an additional flag, -f1.

The -s parameter indicates that the installation is to be carried out in silent mode. For example:

```
SETUP -s -f1C:\WINNT35\SETUP.ISS -l <language selection>
```

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 MQSeries for Windows NT over a network because you can do it from a redirected drive on a LAN server.

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

1. Use the NET USE command as follows:

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

For example:

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

where J: is either the redirected drive containing the MQSeries for Windows NT system you are installing, or if you are applying maintenance, the PTF media that is to be applied.

2. From the root of the directory, type SETUP.

Attention: When applying maintenance upgrades you should use the Setup program supplied with the maintenance media. Also, you **must** ensure that no part of the MQSeries system is running when applying a maintenance upgrade.

Installation Response Files

An installation response file is an ASCII text file containing values for the options that you select when you install or maintain an MQSeries for Windows NT system. This allows installation and maintenance to be performed automatically, without user interaction.

In an installation response file you can specify:

- The path for the program and data files
- Whether the Registry should be updated automatically
- The MQSeries for Windows NT components to be installed
- Whether the MQS.INI file should be overwritten
- The name of the folder where the MQSeries program items are to reside
- Whether you want to view the README file
- Whether the machine should be restarted automatically

Unattended Installation

The installation response file supplied with MQSeries for Windows NT can be found as a sample file on the installation media. Initially, you have only one response file, specified by the `-fl` parameter of the installation program.

To perform other actions you can:

- Copy the sample response file and prepare your own installation response file, using a suitable editor.
- Allow the setup program to generate a response file automatically.

Appendix B. Preparing to Install From a LAN

There are two ways to put the MQSeries installation files on a LAN server for easier access: you can copy the CD-ROM to the hard drive on the LAN server, or make the CD-ROM directory shareable. The following example assumes that you are copying the CD-ROM onto a hard drive labelled M:

1. Create a directory on the LAN server to store the MQSeries installation files. For example:

```
MD M:\INSTMQS
```

2. Copy the contents of the CD-ROM into this directory. You can do this using the XCOPY command from the command prompt. For example:

```
XCOPY E:\*.* M:\INSTMQS /E
```

3. Give all licensed users access to the directory that now contains the CD-ROM image (in this example, the M drive).
4. From the command prompt on the target machine, connect to the appropriate drive and directory for your enterprise using the NET USE command as follows:

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

For example:

```
NET USE X: \\MQMNT\INSTMQS
```

where X: is the redirected drive containing the MQSeries for Windows NT software to be installed. Alternatively, use the File Manager or Explorer to map the shared resource to a drive letter.

5. Change to the installation directory (in this example, the X drive).
6. Type SETUP and press Enter.
7. Follow the on-screen prompts.

Installing from a LAN

Appendix C. Removing MQSeries for Windows NT

This section describes how to remove MQSeries for Windows NT server files from your machine. You can do this in either of the following ways:

- By starting the MQ Uninstaller application from the MQSeries folder:
 1. Open the MQSeries program folder created during the MQSeries installation (by default this folder is called **IBM MQSeries for Windows NT**).
 2. Select the **MQ Uninstaller** icon to run the uninstall application.
- If you are using Windows NT Version 4.0, you can start the MQ Uninstaller application by selecting **MQSeries for Windows NT** from the Add-Remove programs applet in the Control Panel. When the Add-Remove programs applet is running, choose the components to be removed from the list (by default none are selected).

When you remove the MQSeries server component, you can leave your queue manager definitions on the machine by selecting the **Remove Executables Only** radio button. If you want to remove these definitions together with the server, select the **Remove Executables and Data** radio button.

When you have made all the required selections, start the uninstall process by clicking the **Remove** button.

Note: If you choose to remove all currently installed MQSeries components, you must restart the computer before you add any new ones.

Removing MQSeries for Windows NT

Appendix D. 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 116
- Remotely – see “Enrolling the Product License Remotely” on page 116

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 G:\LICENSES\AMPCERT.LIC
```

where G is the drive on which the CD is placed.

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 G:\LICENSES\AMPCERT.LIC
```

where G is the drive on which the CD is placed, or reinstall MQSeries.

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

PC Direct is a trademark of Ziff Communications Company and is used by IBM Corporation under license.

UNIX is a registered trademark in the United States and other countries licensed exclusively through X/Open Company Limited.

C-bus is a trademark of Corollary, Inc.

Microsoft, Windows, and the Windows 95 logo are trademarks or registered trademarks of Microsoft Corporation.

Java and HotJava are trademarks of Sun Microsystems, Inc.

Other company, product, and service names, which may be denoted by a double asterisk (**), may be trademarks or service marks of others.

Index

A

- administration
 - command sets
 - control commands 83
 - MQSeries commands (MQSC) 86
 - programmable command format commands (PCF) 86
- AIX client
 - hardware and software 47
 - installing 69
- application
 - data 10
 - time-independent 10
- attributes
 - ALL attribute 93
 - changing 95
 - default 93

B

- bibliography 99
- BookManager 101
- browsing queues 96

C

- case-sensitive control commands 84
- CD-ROM, installing from 24
- changing queue attributes 95
- channel
 - events 15
 - message 13
 - MQI 13
- clearing a local queue 95
- clients 13
 - installing on DOS 69
 - installing on HP-UX 71
 - installing on OS/2 Warp 71
 - installing on Sun Solaris 75
 - installing on Windows 3.1 77
 - installing on Windows 95 76
 - installing on Windows NT 76
- command set administration 83

- commands
 - control 83
 - MQSC
 - ALTER QLOCAL 95
 - DEFINE QLOCAL 94
 - DEFINE QLOCAL LIKE 94
 - DEFINE QLOCAL REPLACE 95
 - DELETE QLOCAL 96
 - using 86
 - programmable command format (PCF) 86
 - runmqsc 91
- Communications Server for Windows NT 28
- components of MQSeries for Windows NT
 - options 21
- configuration, kernel 71, 75
- control commands
 - case-sensitive 84
 - runmqsc 91
- controlled shutdown 89
- creating
 - groups 67, 70, 74
 - queue manager 87
 - user ID 67, 70, 74
- current queue depth 94

D

- default
 - attributes of objects 93
 - queue manager commands processed 91
- deleting 96
 - local queue 96
 - queue manager 90
- disk space
 - for installation 19
- DOS client
 - hardware and software 51
 - installing 69

E

- ending
 - queue manager 89
- ending interactive MQSC commands

Index

ending interactive MQSC commands (*continued*)
 interactive MQSC commands 92
endmqm command 89
error messages 92
event-driven processing 10
events
 channel 15
 instrumentation 15
 types of 15

F

feedback from MQSC commands 92

G

groups, creating 67, 70, 74

H

hardware requirements
 AIX client 47
 DOS client 51
 HP-UX client 53
 OS/2 client 55
 Sun Solaris client 58
 Windows 3.1 client 60
 Windows 95 client 62
 Windows NT client 64
HP-UX client
 hardware and software 53
 installing 71
HTML (Hypertext Markup Language) 101
Hypertext Markup Language (HTML) 101

I

IBM Communications Server for Windows NT 28
information, ordering publications 102
installable components 21
installation 23
 preparation 67
 product license 115
installation directory 74
installation response file, specification for 109
installing
 AIX client 69
 clients on DOS 69
 clients on HP-UX 71
 clients on OS/2 Warp 71, 72

installing (*continued*)
 clients on Sun Solaris 75
 clients on Windows 3.1 77
 clients on Windows 95 76
 clients on Windows NT 76
installing from a LAN 111
instrumentation events
 description 15
 types of 15
interactive MQSC
 ending 92
 feedback from 92
 using 91
introduction to MQSeries 9

K

kernel configuration 71, 75

L

LAN, installing from 111
LIKE attribute 94
linking to Lotus Notes 14
local queues
 clearing 95
 copying definitions 94
 defining one 92
 deleting 96
Lotus Notes
 mapping 14
 with MQSeries 14

M

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 71
monitoring queue managers 15
MQI
 channel 13

Index

MQI (*continued*)
description 9
MQSC 86
MQSC commands
ALTER QLOCAL 95
DEFINE QLOCAL 94
DEFINE QLOCAL LIKE 94
DEFINE QLOCAL REPLACE 95
DELETE QLOCAL 96
ending interactive input 92
issuing interactively 91
using 86
MQSeries objects 11

N
national language support 71
NLSPATH environment variable 71

O
objects
default attributes 93
process definition 12
queue manager 11
online books 101
ordering publications 102
OS/2 Warp client
hardware and software 55
installing 71

P
performance events 15
PostScript Format 102
preemptive queue manager shutdown 90
process definitions, description of 12
processing, event-driven 10
product license installation 115
programmable command format (PCF)
administration with 86
description 86
publications 99

Q
queue depth
current 94
determining 94

queue manager 11
creating 87
deleting 90
description 11
events 15
immediate shutdown 89
monitoring 15
objects 11
preemptive shutdown 89, 90
restart 90
shutdown
controlled 89
immediate 89
preemptive 89
quiesced 89
starting 87
automatically 88
stopping 89
queues
attributes 12
browsing 96
changing attributes 95
defining 12
description 10
local
clearing 95
copying 94
defining 92
deleting 96
quiesced shutdown 89

R
READ ME file 22
required software
MQSeries for Windows NT 19
response files
installation 109
restart queue manager 90
runmqsc
ending 92
feedback 92
using interactively 91

S
scmmqm command 88
shell commands for MQSeries 83
shutdown queue manager
controlled 89

Index

- shutdown queue manager (*continued*)
 - immediate 89
 - preemptive 89
 - quiesced 89
- softcopy information 101
- software required
 - MQSeries for Windows NT 19
- software requirements
 - AIX client 47
 - DOS client 51
 - HP-UX client 53
 - OS/2 client 55
 - Sun Solaris client 58
 - Windows 3.1 client 60
 - Windows 95 client 62
 - Windows NT client 64
- starting
 - a queue manager automatically 88
- starting a queue manager 87
- stopping queue manager 89
- storage requirements 19
- Sun Solaris client
 - hardware and software 58
 - installing 75
- syntax error, in MQSC commands 92
- System Management Interface Tool (SMIT)
 - using to create IDs 68

T

- time-independent applications 10
- translated messages 71, 75
- trigger monitor 12
- types of event 15

U

- user ID, creating 67, 70, 74

W

- Windows 3.1 client
 - hardware and software 60
 - installing 77
- Windows 95 client
 - hardware and software 62
 - installing 76
- Windows NT client
 - hardware and software 64
 - installing 76

Sending your comments to IBM

MQSeries for Windows NT**

Quick Beginnings

GC33-1871-00

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 Windows NT**

Quick Beginnings

GC33-1871-00

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 Windows NT**

MQSeries for Windows NT Quick Beginnings GC33-1871-00



You can send your comments POST FREE on this form from any one of these countries:

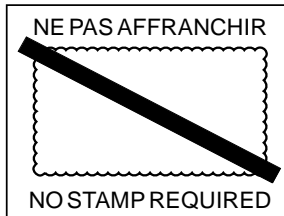
Australia	Finland	Iceland	Netherlands	Singapore	United States
Belgium	France	Israel	New Zealand	Spain	of America
Bermuda	Germany	Italy	Norway	Sweden	
Cyprus	Greece	Luxembourg	Portugal	Switzerland	
Denmark	Hong Kong	Monaco	Republic of Ireland	United Arab Emirates	

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 U.S.A.

GC33-1871-00

