

WebSphere MQ



Clients

Note!

Before using this information and the product it supports, be sure to read the general information under “Notices” on page 179.

Second edition (October 2002)

This edition applies to the following WebSphere MQ V5.3 products:

- WebSphere® MQ for AIX®
- WebSphere MQ for HP-UX
- WebSphere MQ for iSeries™
- WebSphere MQ for Linux for Intel
- WebSphere MQ for Linux for zSeries™
- WebSphere MQ for Solaris
- WebSphere MQ for Windows®
- WebSphere MQ for z/OS™

Unless otherwise stated, the information also applies to these products:

- MQSeries® for AT&T GIS UNIX® V2.2.1
- MQSeries for Compaq NonStop Kernel V5.1
- MQSeries for Compaq OpenVMS Alpha V5.1
- MQSeries for Compaq OpenVMS VAX V2.2.1
- MQSeries for Compaq Tru64 UNIX, V5.1
- MQSeries for OS/2® Warp V5.1
- MQSeries for SINIX and DC/OSx V2.2.1
- MQSeries for Sun Solaris, Intel Platform Edition, V5.1
- MQSeries for VSE/ESA™ V2.1

© Copyright International Business Machines Corporation 1994, 2002. All rights reserved.

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

Contents

Figures	vii
--------------------------	------------

Tables	ix
-------------------------	-----------

About this book	xi
What you need to know to understand this book	xi
Terms used in this book	xi
How to use this book	xii

Summary of changes	xiii
Changes for this release	xiii

Part 1. Introduction to WebSphere MQ clients 1

Chapter 1. Overview of WebSphere MQ clients	3
What is a WebSphere MQ client?	3
How the client connects to the server	4
Why use WebSphere MQ clients?	5
What applications run on a WebSphere MQ client?	5
How do I set up a WebSphere MQ client?	6

Part 2. Installing WebSphere MQ clients 7

Chapter 2. Preparing for installation	11
Platform support for WebSphere MQ clients	11
Applications on Version 5 clients	12
WebSphere MQ clients on other platforms	12
Communications	12
Performance considerations	13
Hardware and software requirements.	14
AIX client: hardware and software required	15
AT&T GIS UNIX (NCR UNIX) client: hardware and software required	17
Compaq Tru64 UNIX client: hardware and software required	18
Compaq OpenVMS Alpha client: hardware and software required	19
DOS client: hardware and software required	20
HP-UX client: hardware and software required	21
Linux client: hardware and software required	22
OS/2 Warp client: hardware and software required	24
SINIX and DC/OSx client: hardware and software required	26
Solaris client: hardware and software required.	28
Sun Solaris, Intel Platform Edition client: hardware and software required	29
VM/ESA client: hardware and software required	30
Windows client: hardware and software required	31

Windows 3.1 client: hardware and software required	33
Windows 95 client: hardware and software required	34

Chapter 3. Installing client components from WebSphere MQ products and Version 5 MQSeries products (not z/OS) 35

Installing a WebSphere MQ or MQSeries Version 5 client and server system	35
Installing WebSphere MQ clients on the same machine as the server	37
Uninstalling WebSphere MQ clients	37
Installing on AIX	38
Components for AIX	38
Before installing	38
Easy installation.	40
Custom installation.	41
Migrating from an earlier version of MQSeries for AIX.	42
Changing the national language	43
Translated messages	44
Removing a WebSphere MQ client from AIX	44
Migrating to and from the WebSphere MQ SSL support.	44
Installing on Compaq Tru64 UNIX.	45
Before installation	45
Installation	45
Translated messages	46
Uninstalling an MQSeries client from Compaq Tru64 UNIX	46
Installing on HP-UX	48
Components for HP-UX	48
Before installation	48
Installation	49
Translated messages	49
Removing a WebSphere MQ client from HP-UX	50
Migrating to and from the WebSphere MQ SSL support.	50
Installing on Linux	51
Components for Linux.	51
Before installation	52
Installation	53
Translated messages	54
Removing the WebSphere MQ client from Linux	54
Migrating to and from the WebSphere MQ SSL support.	55
Installing on OS/2 Warp	56
Components for OS/2 Warp.	56
Installation	56
Unattended installation on OS/2 Warp	58
Installation and maintenance parameters	60
Installation response files.	61
Removing an MQSeries client from OS/2 Warp	63

Installing on Solaris	64
Components for Solaris	64
Before installation	64
Installation	65
Translated messages	66
Removing a WebSphere MQ client from Solaris	66
Migrating to and from the WebSphere MQ SSL	
support.	66
Installing on Sun Solaris, Intel Platform Edition	67
Before Installation	67
Installation	67
Uninstalling the Sun Solaris, Intel Platform	
Edition client.	68
Kernel Configuration	68
Translated Messages	68
Installing on Windows.	69
Preparing to install the WebSphere MQ client	69
Installing the WebSphere MQ client	69
Other methods of installing the WebSphere MQ	
client	72
Uninstalling a WebSphere MQ client	76

Chapter 4. Installing MQSeries clients with other MQSeries products. 79

Obtaining MQSeries clients from the MQSeries products	80
MQSeries client and server on the same platform	80
MQSeries client and server on different platforms	80
Obtaining WebSphere MQ clients from IBM Transaction Processing SupportPacs	81
Installing the MQSeries server	81
MQSeries client files on the server.	81
Installing MQSeries clients from MQSeries for Compaq OpenVMS Alpha	82
Installing an MQSeries client on Compaq OpenVMS Alpha from MQSeries for Compaq OpenVMS Alpha	82
Installing an MQSeries client on OS/2 Warp from MQSeries for Compaq OpenVMS Alpha.	83
Installing an MQSeries client on DOS from MQSeries for Compaq OpenVMS Alpha.	84
Installing an MQSeries client on Windows 3.1 from MQSeries for Compaq OpenVMS Alpha	84
Installing MQSeries clients from MQSeries on UNIX systems.	85
Installing an MQSeries client on a UNIX system from a UNIX system	85
Installing an MQSeries client on OS/2 Warp from a UNIX system	86
Installing an MQSeries client on DOS from a UNIX system.	87
Installing an MQSeries client on Windows 3.1 from a UNIX system	87
MQSeries client for VM/ESA	88
Changing the config.sys and autoexec.bat files.	88
Changing the OS/2 Warp config.sys file.	88
Changing the autoexec.bat file for DOS and Windows 3.1	89

Chapter 5. Configuring communication links. 91

Deciding which communication type to use	91
Defining a TCP/IP connection	93
TCP/IP connection limits.	93
Defining a TCP/IP connection on a WebSphere MQ client (any platform).	94
Defining a TCP/IP connection on an OS/2 Warp server	94
Defining a TCP/IP connection on a Windows server	95
Defining a TCP/IP connection on a UNIX system server	96
Defining a TCP/IP connection on an OS/400 server	97
Defining a TCP/IP connection on a z/OS server	97
Defining a TCP/IP connection on a Compaq OpenVMS Alpha server	97
Defining a TCP/IP connection on a Compaq NonStop Kernel server	98
Defining a TCP/IP connection on a VSE/ESA server	98
Defining an LU 6.2 connection	99
Defining an LU 6.2 Connection on an OS/2 Warp client	99
Defining an LU 6.2 Connection on an OS/2 Warp server.	100
Defining an LU 6.2 Connection on a Windows WebSphere MQ client	101
Defining an LU 6.2 Connection on a Windows server	101
Defining an LU 6.2 Connection on a UNIX WebSphere MQ client	102
Defining an LU 6.2 Connection on a UNIX server	102
Defining an LU 6.2 Connection on an OS/400 server	103
Defining an LU 6.2 Connection on a Compaq NonStop Kernel server	104
Defining an LU 6.2 Connection on a z/OS server	104
Defining an LU 6.2 Connection on a Compaq OpenVMS Alpha client	105
Defining a NetBIOS connection	106
NetBIOS on a WebSphere MQ client (any suitable platform)	106
NetBIOS on an OS/2 Warp server	106
NetBIOS on a Windows server	107
Defining an SPX connection	108
Defining an SPX connection on a WebSphere MQ client (any suitable platform)	108
Defining an SPX connection on an OS/2 Warp or Windows server	108
SPX and IPX parameters.	108
Defining a DECnet connection.	111
Defining a DECnet connection on a WebSphere MQ client.	111
Defining a DECnet connection on a Compaq OpenVMS Alpha server	111

Chapter 6. Verifying the installation	113
The installation used for the example	113
What the example shows	113
Setting up the server	114
Setting up a Windows server using WebSphere MQ Explorer	115
Setting up the server (OS/400)	115
Setting up the server (z/OS)	116
Setting up the server (VSE/ESA)	116
Setting up the WebSphere MQ client.	116
Defining a client-connection channel using MQSERVER	117
Defining a client-connection channel using WebSphere MQ Explorer	117
Putting a message on the queue	117
On the WebSphere MQ client workstation (not VM/ESA or Windows 3.1)	118
On the WebSphere MQ client workstation (VM/ESA)	118
On the WebSphere MQ client workstation (Windows 3.1)	118
Getting the message from the queue.	119
On the WebSphere MQ client workstation (not VM/ESA or Windows 3.1)	119
On the WebSphere MQ client workstation (VM/ESA)	119
On the WebSphere MQ client workstation (Windows 3.1)	119
Ending verification	120
 Part 3. System administration	 121
 Chapter 7. Setting up WebSphere MQ client security	 123
Authentication	123
Environment variables	124
User IDs	124
Access control	125
 Chapter 8. Using channels.	 127
What is a channel?	127
Message Channels.	127
MQI Channels	127
Defining MQI channels	128
Automatically defined channels	128
User defined channels	129
Creating one definition on the WebSphere MQ client and the other on the server.	129
On the server	129
On the WebSphere MQ client	130
Creating both definitions on the server.	132
Defining the server-connection channel.	132
Defining the client-connection channel	133
Accessing client-connection channel definitions	133
Client channel definition table.	134
Migrating to a later release level of WebSphere MQ	135
Channel exits	136
Path to exits.	136
Connecting a client to a queue-sharing group	136
Connecting to a specific queue manager	137

Connecting to the generic interface	137
Stopping channels.	137

Chapter 9. The Secure Sockets Layer (SSL) on WebSphere MQ clients

Specifying that an MQI channel uses SSL	139
Specifying the location of LDAP servers that hold certificate revocation lists (CRLs)	139
When a WebSphere MQ client application issues an MQCONN call	140
Using a client channel definition table	140
Using Active Directory on Windows.	141

Chapter 10. Using WebSphere MQ environment variables

MQCCSID	144
MQCHLLIB	144
MQCHLTAB	145
MQDATA (DOS and Windows 3.1 only)	145
MQNAME	146
MQ_PASSWORD (DOS, OS/2 Warp, Windows 3.1, Windows 95, and Windows 98 only).	146
MQSERVER	146
TCP/IP default port	147
SPX default socket.	147
Examples of using MQSERVER	147
Canceling MQSERVER	148
MQTRACE (DOS, Windows 3.1, and VM/ESA only)	148
MQ_USER_ID (DOS, OS/2 Warp, Windows 3.1, Windows 95, and Windows 98 only).	148
MQSWORKPATH (OS/2 Warp only)	148

Part 4. Application programming

Chapter 11. Using the message queue interface (MQI)

Limiting the size of a message.	151
Choosing client or server coded character set identifier (CCSID)	151
CCSID and encoding fields - multiple puts	152
Controlling application in a Windows 3.1 environment.	152
Designing applications	152
Windows 3.1 environment	152
Using MQINQ	152
Using syncpoint coordination	153
Compaq NonStop Kernel server	153
Using MQCONN	153
Shared connection handles on MQCONN	154

Chapter 12. Building applications for WebSphere MQ clients

Running applications in the WebSphere MQ client environment.	156
Triggering in the client environment.	156
Process definition	157
Trigger monitor	157
CICS applications (non-z/OS)	158

Linking C applications with the WebSphere MQ client code	158
Running 16-bit and 32-bit Windows clients	160
Linking C++ applications with the WebSphere MQ client code	160
Linking COBOL applications with the WebSphere MQ client code	160
Linking PL/I applications with the WebSphere MQ client code	161
Linking Visual Basic applications with the WebSphere MQ client code	162

Chapter 13. Running applications on WebSphere MQ clients 163

Using environment variables	164
Using MQSERVER	164
Using MQCHLLIB and MQCHLTAB	164
Using the MQCNO structure	164
Using DEFINE CHANNEL	164
Role of the client channel definition table	165
Multiple queue managers	165
Queue-sharing groups	165
Examples of MQCONN calls	165
What the examples demonstrate	167
Example 1. Queue manager name includes an asterisk (*)	167
Example 2. Queue manager name specified	168
Example 3. Queue manager name is blank or an asterisk (*)	168

Chapter 14. Solving problems 171

WebSphere MQ client fails to make a connection	171
Stopping WebSphere MQ clients	172
Error messages with WebSphere MQ clients	172
Compaq OpenVMS Alpha, OS/2 Warp, UNIX systems, Windows, and Windows 95	172
DOS and Windows 3.1 clients	172
How to read the error log and FFDCs for DOS and Windows 3.1	172
Using trace on DOS and Windows 3.1	173
Example DOS trace data.	173
Using trace on OS/2 Warp, Windows, and Windows 95.	174
File names for trace files.	175
How to examine First Failure Support Technology™ (FFST) files	175
Using trace on AIX and AT&T GIS UNIX	175
Using trace on Compaq OpenVMS Alpha, HP-UX, SINIX, DC/OSx, and Solaris	176
File names for trace files.	177
How to examine FFSTs	177
Using trace on VM/ESA	177
Example VM/ESA trace data	177

Appendix. Notices 179

Trademarks	181
----------------------	-----

Index 183

Sending your comments to IBM 189

Figures

1. Link between a client and server	3	6. Client-connection and server-connection on	
2. WebSphere MQ server connected to clients on		an MQI channel.	128
different platforms	5	7. Simple channel definition	131
3. LU 6.2 communication setup panel - initiated		8. Defining the server-connection channel	133
end	103	9. Defining the client-connection channel	133
4. Security in a client-server connection	123	10. MQCONN example	166
5. Message channels between two queue		11. Extract from a DOS client trace	174
managers	127	12. Extract from a VM/ESA client trace	178

Tables

1.	Transmission protocols for MQI channels	12	8.	Settings on the WebSphere MQ client system for a server platform	99
2.	Sets of client components on Client CD-ROM 2	53	9.	Stop Channel: Stop Modes and Reason Codes issued	138
3.	Order for removing components	55	10.	Programming languages supported in client environments	155
4.	Features installed with each type of installation	69			
5.	Valid feature names	74			
6.	Transmission protocols - combination of WebSphere MQ client and server platforms . .	92			
7.	Maximum outstanding connection requests queued at a TCP/IP port	93			

About this book

This book contains information about the WebSphere MQ client/server environment. It describes how to install a WebSphere MQ client, how to configure the communication links and how to set up WebSphere MQ channels so that your WebSphere MQ applications can run on the client machine.

The WebSphere MQ environment variables are described and there are chapters about building and running your applications on a WebSphere MQ client.

Most of the information you need to know about WebSphere MQ clients is in this book. Some of the reference material in the other WebSphere MQ books includes information about WebSphere MQ clients. That reference material is not repeated here.

This book is intended for system administrators, for anyone who installs and configures WebSphere MQ products for the client-server environment, and for application programmers who write programs to make use of the Message Queue Interface (MQI).

What you need to know to understand this book

To understand this book, you should have:

- Experience in installing and configuring the system you use for the server. This can be Compaq OpenVMS Alpha, OS/400®, Compaq NonStop Kernel, OS/2 Warp, z/OS, one of the UNIX systems listed below, VSE/ESA, or Windows.
- Experience with any client platforms that you will be using.
- Understanding of the purpose of the Message Queue Interface (MQI).

For further information on MQI, see the following manuals:

- For MQI call constants, see the *WebSphere MQ Application Programming Reference* manual.
 - For commands and responses, see the *WebSphere MQ Programmable Command Formats and Administration Interface* manual, and for events, see the *WebSphere MQ Event Monitoring* manual.
- Experience of WebSphere MQ programs in general, or familiarity with the content of the other WebSphere MQ publications.

Terms used in this book

The term *Linux* denotes:

- Linux for Intel
- Linux for zSeries

The term *UNIX systems* denotes the following UNIX operating systems:

- AIX
- AT&T GIS UNIX (NCR UNIX)
- Compaq Tru64 UNIX
- DC/OSx
- HP-UX

About this book

- Linux for Intel
- Linux for zSeries
- SINIX
- Solaris
- Sun Solaris, Intel Platform Edition

The term *Windows* denotes the following Windows operating systems:

- Windows 98
- Windows 2000
- Windows NT[®]
- Windows XP

Where it is clear from the context that only WebSphere MQ server platforms are being referenced, the term *Windows* denotes:

- Windows 2000
- Windows NT
- Windows XP

Any references to Windows 3.1 and Windows 95 are explicit.

The term *z/OS* means any release of z/OS or OS/390[®] supported by the current version of WebSphere MQ for z/OS.

The variable *mqmtop* represents the name of the base directory where WebSphere MQ is installed on UNIX systems.

- On AIX, the actual name of the directory is **/usr/mqm**
- On other UNIX systems, the actual name of the directory is **/opt/mqm**

How to use this book

Read Chapter 1, “Overview of WebSphere MQ clients” on page 3 first as a brief introduction. “How do I set up a WebSphere MQ client?” on page 6 gives you a list of tasks that you might need to carry out and guides you through the rest of the book.

Summary of changes

This section describes changes in this edition of *WebSphere MQ Clients*. Changes since the previous edition of the book are marked by vertical lines to the left of the changes.

Changes for this release

- The book documents the WebSphere MQ Version 5.3 clients on the following platforms:
 - AIX
 - HP-UX
 - Linux
 - OS/400 (for Java applications only)
 - Solaris
 - Windows
- There is a new chapter about using the WebSphere MQ Secure Sockets Layer (SSL) support on client systems.
- There are various editorial improvements, clarifications, and corrections throughout the book.

Changes

Part 1. Introduction to WebSphere MQ clients

Chapter 1. Overview of WebSphere MQ clients	3
What is a WebSphere MQ client?	3
How the client connects to the server	4
Client and queue manager on the same machine	4
Clients on different platforms	4
Why use WebSphere MQ clients?	5
What applications run on a WebSphere MQ client?	5
How do I set up a WebSphere MQ client?	6

Introduction

Chapter 1. Overview of WebSphere MQ clients

This chapter discusses the following topics:

- “What is a WebSphere MQ client?”
- “Why use WebSphere MQ clients?” on page 5
- “How do I set up a WebSphere MQ client?” on page 6

What is a WebSphere MQ client?

A WebSphere MQ client is part of the WebSphere MQ product that can be installed on its own, on a separate machine from the base product and server. You can run a WebSphere MQ application on a WebSphere MQ client and it can interact with one or more WebSphere MQ servers and connect to their queue managers by means of a communications protocol. The servers to which the client connects might or might not be part of a cluster.

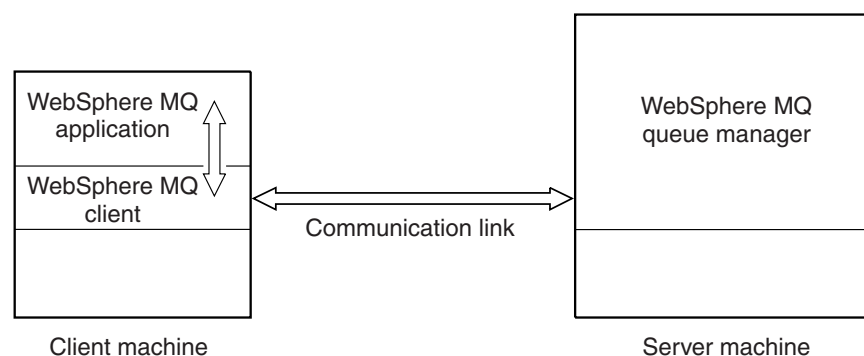


Figure 1. Link between a client and server

The following platforms can be used. The combinations depend on which WebSphere MQ product you are using and are described in “Platform support for WebSphere MQ clients” on page 11. Other WebSphere MQ clients are also available; these are described in “WebSphere MQ clients on other platforms” on page 12.

WebSphere MQ client

Compaq OpenVMS Alpha
DOS
OS/2 Warp
OS/400 (for Java™ applications only)
UNIX systems
VM/ESA
Windows

WebSphere MQ server

Compaq NonStop Kernel
Compaq OpenVMS Alpha
OS/2 Warp
OS/400
UNIX systems
VSE/ESA
Windows
z/OS

The MQI is available to applications running on the client platform; the queues and other WebSphere MQ objects are held on a queue manager that you have installed on a server machine.

An application that you want to run in the WebSphere MQ client environment must first be linked with the relevant client library. When the application issues an MQI call, the WebSphere MQ client directs the request to a queue manager, where it is processed and from where a reply is sent back to the WebSphere MQ client.

Overview of WebSphere MQ clients

The link between the application and the WebSphere MQ client is established dynamically at runtime (except in the case of DOS, when it is a static link).

The WebSphere MQ client on OS/400 supports only Java applications that use WebSphere MQ classes for Java or WebSphere MQ classes for Java Message Service (JMS). It does not support applications that use the MQI directly. For full details on how to install, configure, and use WebSphere MQ classes for Java and WebSphere MQ classes for JMS on an OS/400 client system, or on any other client platform that supports Java messaging, see *WebSphere MQ Using Java*.

How the client connects to the server

An application running in the WebSphere MQ client environment runs in synchronous mode because there must be an active connection between the client and server machines.

The connection is made by an application issuing an MQCONN or MQCONNX call. Clients and servers communicate through *MQI channels*. When the call succeeds, the MQI channel remains connected until the application issues a MQDISC call. This is the case for every queue manager that an application needs to connect to.

Client and queue manager on the same machine

You can also run an application in the WebSphere MQ client environment when your machine also has a queue manager installed. In this situation, you have the choice of linking to the queue manager libraries or the client libraries, but remember that if you link to the client libraries, you still need to define the channel connections. This can be useful during the development phase of an application. You can test your program on your own machine, with no dependency on others, and be confident that it will still work when you move it to a full WebSphere MQ environment.

Clients on different platforms

Here is another example of a WebSphere MQ client and server system. In this example, the server machine communicates with three WebSphere MQ clients on different platforms.

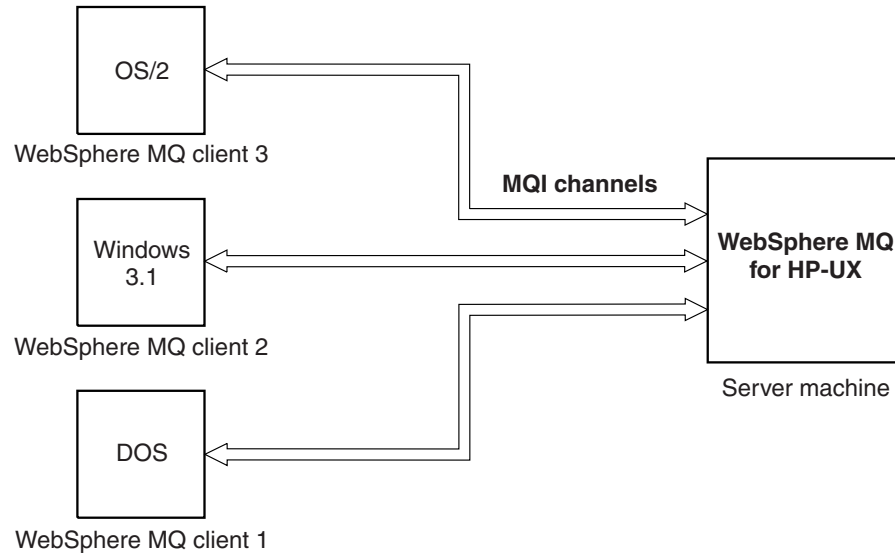


Figure 2. WebSphere MQ server connected to clients on different platforms

Other more complex environments are possible. For example, a WebSphere MQ client can connect to more than one queue manager, or any number of queue managers connected as part of a queue-sharing group.

Why use WebSphere MQ clients?

Using WebSphere MQ clients is an efficient way of implementing WebSphere MQ messaging and queuing.

You can have an application that uses the MQI running on one machine and the queue manager running on a different machine (either physical or virtual). The benefits of doing this are:

- There is no need for a full WebSphere MQ implementation on the client machine; for example, it could be a DOS, Windows 3.1, Windows 95, or Windows 98 system.
- Hardware requirements on the client system are reduced.
- System administration requirements are reduced.
- a WebSphere MQ application running on a client can connect to multiple queue managers on different systems.
- Alternative channels using different transmission protocols can be used.

What applications run on a WebSphere MQ client?

The full MQI is supported in the client environment. This enables almost any WebSphere MQ application to be relinked to run on a WebSphere MQ client by link the application on the WebSphere MQ client to the MQIC library, rather than to the MQI library. The exceptions are:

- An application that needs syncpoint coordination with other resource managers
- MQGET with signal

An application running on a WebSphere MQ client can connect to more than one queue manager concurrently, or use a queue manager name with an asterisk (*) on an MQCONN or MQCONNEX call (see the examples in Chapter 13, “Running applications on WebSphere MQ clients” on page 163).

How do I set up a WebSphere MQ client?

To set up a WebSphere MQ client you need to have a WebSphere MQ server already installed and working on a machine, to which your client will connect. The steps involved in setting up a client are:

1. Check that you have a suitable platform for a WebSphere MQ client and that the hardware and software satisfy the requirements. This is described in Chapter 2, “Preparing for installation” on page 11.
2. Decide how you are going to install WebSphere MQ on your client machine, and then follow the instructions for your particular combination of client and server platforms. This is described in Chapter 3, “Installing client components from WebSphere MQ products and Version 5 MQSeries products (not z/OS)” on page 35, and Chapter 4, “Installing MQSeries clients with other MQSeries products” on page 79.
3. Ensure that your communication links are configured and connected. This is described in Chapter 5, “Configuring communication links” on page 91.
4. Check that your installation is working correctly. This is described in Chapter 6, “Verifying the installation” on page 113.
5. When you have the verified WebSphere MQ client installation, consider whether you need to take any action on security. This is described in Chapter 7, “Setting up WebSphere MQ client security” on page 123.
6. Set up the channels between the WebSphere MQ client and server that are required by the WebSphere MQ applications you want to run on the client. This is described in Chapter 8, “Using channels” on page 127. There are some additional considerations if you are using SSL. These are described in Chapter 9, “The Secure Sockets Layer (SSL) on WebSphere MQ clients” on page 139. You might need to use WebSphere MQ *environment variables* to set up the channels. These are described in Chapter 10, “Using WebSphere MQ environment variables” on page 143.
7. WebSphere MQ applications are fully described in the *WebSphere MQ Application Programming Guide*.
8. There are some differences to consider when designing, building and running applications in the WebSphere MQ client environment. For information about these differences, see:
 - Chapter 11, “Using the message queue interface (MQI)” on page 151
 - Chapter 12, “Building applications for WebSphere MQ clients” on page 155
 - Chapter 13, “Running applications on WebSphere MQ clients” on page 163
 - Chapter 14, “Solving problems” on page 171

Part 2. Installing WebSphere MQ clients

Chapter 2. Preparing for installation	11
Platform support for WebSphere MQ clients	11
Applications on Version 5 clients	12
WebSphere MQ clients on other platforms	12
Communications	12
Performance considerations	13
Hardware and software requirements	14
AIX client: hardware and software required	15
Hardware	15
Software	15
Optional software	15
AT&T GIS UNIX (NCR UNIX) client: hardware and software required	17
Hardware	17
Software	17
Optional software	17
Compaq Tru64 UNIX client: hardware and software required	18
Hardware	18
Software	18
Optional software	18
Compaq OpenVMS Alpha client: hardware and software required	19
Hardware	19
Software	19
Optional software	19
DOS client: hardware and software required	20
Hardware	20
Software	20
Optional software	20
HP-UX client: hardware and software required	21
Hardware	21
Software	21
Optional software	21
Linux client: hardware and software required	22
Hardware	22
Software	22
Optional software	22
OS/2 Warp client: hardware and software required	24
Hardware	24
Software	24
Optional software	24
SINIX and DC/OSx client: hardware and software required	26
Hardware	26
Software	26
Optional software	26
Solaris client: hardware and software required	28
Hardware	28
Software	28
Optional software	28
Sun Solaris, Intel Platform Edition client: hardware and software required	29
Hardware	29
Software	29
Optional software	29
VM/ESA client: hardware and software required	30
Hardware	30
Software	30
Optional software	30
Windows client: hardware and software required	31
Hardware	31
Software	31
Optional software	31
Windows 3.1 client: hardware and software required	33
Hardware	33
Software	33
Optional software	33
Windows 95 client: hardware and software required	34
Hardware	34
Software	34
Optional software	34
Chapter 3. Installing client components from WebSphere MQ products and Version 5 MQSeries products (not z/OS)	35
Installing a WebSphere MQ or MQSeries Version 5 client and server system	35
Installing WebSphere MQ clients on the same machine as the server	37
Uninstalling WebSphere MQ clients	37
Installing on AIX	38
Components for AIX	38
Before installing	38
Creating another file system for the client	39
Creating the mqm user ID and group	39
Easy installation	40
Custom installation	41
Migrating from an earlier version of MQSeries for AIX	42
Changes to the installation path	42
Changing the national language	43
Translated messages	44
Removing a WebSphere MQ client from AIX	44
Migrating to and from the WebSphere MQ SSL support	44
Installing on Compaq Tru64 UNIX	45
Before installation	45
Installation	45
Translated messages	46
Uninstalling an MQSeries client from Compaq Tru64 UNIX	46
Installing on HP-UX	48
Components for HP-UX	48
Before installation	48
Installation	49
Translated messages	49
Removing a WebSphere MQ client from HP-UX	50

Installing WebSphere MQ clients

	Migrating to and from the WebSphere MQ SSL support.	50
	Installing on Linux	51
	Components for Linux.	51
	Before installation	52
	Creating the file systems	52
	Creating the mqm user ID and mqm group.	53
	Installation	53
	Translated messages	54
	Removing the WebSphere MQ client from Linux	54
	Migrating to and from the WebSphere MQ SSL support.	55
	Installing on OS/2 Warp	56
	Components for OS/2 Warp.	56
	Installation	56
	Unattended installation on OS/2 Warp	58
	Starting unattended installation.	59
	Installation and maintenance parameters	60
	Installation response files.	61
	Response file structure.	62
	Keywords for response files	62
	Removing an MQSeries client from OS/2 Warp	63
	Installing on Solaris	64
	Components for Solaris	64
	Before installation	64
	Installation	65
	Translated messages	66
	Removing a WebSphere MQ client from Solaris	66
	Migrating to and from the WebSphere MQ SSL support.	66
	Installing on Sun Solaris, Intel Platform Edition	67
	Before Installation	67
	Installation	67
	Uninstalling the Sun Solaris, Intel Platform Edition client	68
	Kernel Configuration	68
	Translated Messages	68
	Installing on Windows.	69
	Preparing to install the WebSphere MQ client	69
	Installing the WebSphere MQ client	69
	Typical client installation	69
	Compact client installation	71
	Custom client installation.	71
	Modifying the client installation	71
	Modifying the client installation using Add/Remove Programs	72
	Other methods of installing the WebSphere MQ client	72
	Installing from a LAN.	72
	Unattended (silent) installation	73
	Advanced installation methods.	75
	Using Microsoft System Management Server	75
	Uninstalling a WebSphere MQ client	76
	Uninstalling WebSphere MQ client from Windows	76

Chapter 4. Installing MQSeries clients with other MQSeries products	79
Obtaining MQSeries clients from the MQSeries products	80
MQSeries client and server on the same platform	80

MQSeries client and server on different platforms	80
Obtaining WebSphere MQ clients from IBM Transaction Processing SupportPacs	81
Installing the MQSeries server	81
MQSeries client files on the server.	81
Compaq OpenVMS Alpha and UNIX systems z/OS, Compaq NonStop Kernel, and VSE/ESA	81
Installing MQSeries clients from MQSeries for Compaq OpenVMS Alpha	82
Installing an MQSeries client on Compaq OpenVMS Alpha from MQSeries for Compaq OpenVMS Alpha	82
Installing an MQSeries client on OS/2 Warp from MQSeries for Compaq OpenVMS Alpha.	83
Installing an MQSeries client on DOS from MQSeries for Compaq OpenVMS Alpha.	84
Installing an MQSeries client on Windows 3.1 from MQSeries for Compaq OpenVMS Alpha	84
Installing MQSeries clients from MQSeries on UNIX systems.	85
Installing an MQSeries client on a UNIX system from a UNIX system	85
Installing an MQSeries client on OS/2 Warp from a UNIX system	86
Installing an MQSeries client on DOS from a UNIX system.	87
Installing an MQSeries client on Windows 3.1 from a UNIX system	87
MQSeries client for VM/ESA	88
Changing the config.sys and autoexec.bat files.	88
Changing the OS/2 Warp config.sys file.	88
Changing the autoexec.bat file for DOS and Windows 3.1	89

Chapter 5. Configuring communication links	91
Deciding which communication type to use	91
Defining a TCP/IP connection	93
TCP/IP connection limits.	93
Defining a TCP/IP connection on a WebSphere MQ client (any platform).	94
Defining a TCP/IP connection on an OS/2 Warp server	94
Using inetd	94
Using the Run Listener (RUNMQLSR) command	95
Defining a TCP/IP connection on a Windows server	95
Using the Run Listener (RUNMQLSR) command	95
Defining a TCP/IP connection on a UNIX system server	96
Using the Run Listener (RUNMQLSR) command	97
Defining a TCP/IP connection on an OS/400 server	97
Defining a TCP/IP connection on a z/OS server	97
Defining a TCP/IP connection on a Compaq OpenVMS Alpha server	97
Defining a TCP/IP connection on a Compaq NonStop Kernel server	98

Installing WebSphere MQ clients

Defining a TCP/IP connection on a VSE/ESA server	98	What the example shows	113
Defining an LU 6.2 connection	99	Security	113
Defining an LU 6.2 Connection on an OS/2 Warp client	99	Setting up the server	114
Configuring SNA	99	Setting up a Windows server using WebSphere MQ Explorer	115
Setting the Transaction Program name	99	Setting up the server (OS/400)	115
Establishing a session	99	Setting up the server (z/OS)	116
Defining an LU 6.2 Connection on an OS/2 Warp server	100	Setting up the server (VSE/ESA)	116
Using the RUNMQLSR command	100	Setting up the WebSphere MQ client.	116
Using Communications Manager/2	100	Defining a client-connection channel using MQSERVER	117
Defining an LU 6.2 Connection on a Windows WebSphere MQ client	101	Defining a client-connection channel using WebSphere MQ Explorer	117
Defining an LU 6.2 Connection on a Windows server	101	Putting a message on the queue	117
Using the RUNMQLSR command	101	On the WebSphere MQ client workstation (not VM/ESA or Windows 3.1)	118
Using an SNA server	102	On the WebSphere MQ client workstation (VM/ESA)	118
Defining an LU 6.2 Connection on a UNIX WebSphere MQ client	102	On the WebSphere MQ client workstation (Windows 3.1)	118
Defining an LU 6.2 Connection on a UNIX server	102	Getting the message from the queue.	119
Defining an LU 6.2 Connection on an OS/400 server	103	On the WebSphere MQ client workstation (not VM/ESA or Windows 3.1)	119
Defining an LU 6.2 Connection on a Compaq NonStop Kernel server	104	On the WebSphere MQ client workstation (VM/ESA)	119
Defining an LU 6.2 Connection on a z/OS server	104	On the WebSphere MQ client workstation (Windows 3.1)	119
Defining an LU 6.2 Connection on a Compaq OpenVMS Alpha client	105	Ending verification	120
Defining a NetBIOS connection	106		
NetBIOS on a WebSphere MQ client (any suitable platform)	106		
NetBIOS on an OS/2 Warp server	106		
Start a listener program	107		
NetBIOS on a Windows server	107		
Start a listener program	107		
Defining an SPX connection	108		
Defining an SPX connection on a WebSphere MQ client (any suitable platform)	108		
Defining an SPX connection on an OS/2 Warp or Windows server	108		
Using the Run Listener (RUNMQLSR) command	108		
SPX and IPX parameters.	108		
SPX parameters on an OS/2 Warp client	108		
SPX parameters on a DOS or Windows 3.1 client	109		
SPX parameters on a Windows client	109		
SPX parameters on a WebSphere MQ client for Windows 95	109		
Defining a DECnet connection.	111		
Defining a DECnet connection on a WebSphere MQ client.	111		
Defining a DECnet connection on a Compaq OpenVMS Alpha server	111		
Receiving on DECnet Phase IV	111		
Receiving on DECnet OSI	111		
Chapter 6. Verifying the installation	113		
The installation used for the example	113		

Installing WebSphere MQ clients

Chapter 2. Preparing for installation

This chapter discusses the following topics:

- “Platform support for WebSphere MQ clients”
- “Communications” on page 12
- “Hardware and software requirements” on page 14

Platform support for WebSphere MQ clients

The platform support for WebSphere MQ clients and servers is as follows. Any of the products listed is installed as a *Base product and Server (Base product and Client Attachment feature* on WebSphere MQ for z/OS). These WebSphere MQ products can accept connections from the WebSphere MQ clients on the platforms listed, subject to differences in coded character set identifier (CCSID) and communications protocol.

If you are using previous versions of MQSeries products, make sure that code conversion from the CCSID of your client is supported by the server. See the language support tables in the *WebSphere MQ Application Programming Reference* book for more information.

The following products:

WebSphere MQ Version 5.3 products

- WebSphere MQ for AIX
- WebSphere MQ for HP-UX
- WebSphere MQ for iSeries
- WebSphere MQ for Linux for Intel
- WebSphere MQ for Linux for zSeries
- WebSphere MQ for Solaris
- WebSphere MQ for Windows
- WebSphere MQ for z/OS

MQSeries Version 5 products

- MQSeries for Compaq NonStop Kernel
- MQSeries for Compaq OpenVMS Alpha
- MQSeries for Compaq Tru64 UNIX
- MQSeries for OS/2 Warp
- MQSeries for Sun Solaris, Intel Platform Edition

MQSeries non-Version 5 products

- MQSeries for AT&T GIS UNIX
- MQSeries for SINIX and DC/OSx
- MQSeries for VSE/ESA

can accept connections from WebSphere MQ clients on the following platforms:

- Compaq OpenVMS Alpha
- DOS
- OS/2 Warp

Platform support for WebSphere MQ clients

- OS/400 (for Java applications only)
- UNIX systems
- VM/ESA®
- Windows
- Windows 3.1
- Windows 95

Applications on Version 5 clients

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

WebSphere MQ clients on other platforms

Most WebSphere MQ products supply files for clients on the same platform as the server and for clients on other platforms. The following WebSphere MQ products supply files for clients on other platforms only:

- WebSphere MQ for iSeries
- WebSphere MQ for z/OS
- MQSeries for Compaq Tru64 UNIX
- MQSeries for VSE/ESA

For details see “Compaq OpenVMS Alpha and UNIX systems” on page 81 and Chapter 3, “Installing client components from WebSphere MQ products and Version 5 MQSeries products (not z/OS)” on page 35.

Communications

WebSphere MQ clients use MQI channels to communicate with the server. A channel definition must be created at both the WebSphere MQ client and server ends of the connection. How to do this is explained in “Defining MQI channels” on page 128.

The transmission protocols possible are shown in the following table:

Table 1. Transmission protocols for MQI channels

Client platform	LU 6.2	TCP/IP	NetBIOS	SPX	DECnet
Compaq OpenVMS Alpha	✓	✓			✓
DOS		✓	✓	✓	
OS/2 Warp	✓	✓	✓	✓	
OS/400 ¹		✓			
UNIX systems	✓ ²	✓			
VM/ESA	✓	✓			
Windows	✓ ³	✓	✓	✓	
Windows 3.1		✓	✓	✓	
Windows 95		✓	✓	✓	

Table 1. Transmission protocols for MQI channels (continued)

Client platform	LU 6.2	TCP/IP	NetBIOS	SPX	DECnet
Notes: 1. For Java applications only 2. Except on Linux for zSeries and Sun Solaris, Intel Platform Edition 3. Except on Windows 98					

Table 6 on page 92 shows the possible combinations of WebSphere MQ client and server platforms, using these transmission protocols.

A WebSphere MQ application on a WebSphere MQ client can use all the MQI calls in the same way as when the queue manager is local. MQCONN or MQCONNX associates the WebSphere MQ application with the selected queue manager, creating a *connection handle*. Other calls using that connection handle are then processed by the connected queue manager. WebSphere MQ client communication is synchronous, in contrast to communication between queue managers, which is connection-independent and time-independent.

The transmission protocol is specified via the channel definition and does not affect the application. For example, a Windows application can connect to one queue manager over TCP/IP and to another queue manager over NetBIOS.

Performance considerations

The transmission protocol you use might affect the performance of the WebSphere MQ client and server system.

For dial-up support over a slow telephone line, it might be advisable to use channel exits to compress the data transmitted.

Hardware and software requirements

The following table shows where you can find hardware and software requirements for the client platforms.

Client platform	Page
AIX	15
AT&T GIS UNIX (NCR UNIX)	17
Compaq OpenVMS Alpha	19
Compaq Tru64 UNIX	18
DOS	20
HP-UX	21
Linux	22
OS/2 Warp	24
SINIX and DC/OSx	26
VM/ESA	30
Solaris	28
Sun Solaris, Intel Platform Edition	29
Windows	31
Windows 3.1	33
Windows 95	34

For your server platform hardware and software requirements, see the manual that describes installation for your platform.

AIX client: hardware and software required

This section outlines the hardware and software requirements for the WebSphere MQ client for AIX.

Hardware

The WebSphere MQ client for AIX can run on any machine that supports the AIX V4.3.3 PowerPC[®] 32-bit, or AIX V5.1 Power 32-bit only operating system, whether from IBM[®] or other vendors. For example:

- IBM RS/6000[®] POWERserver[®]
- IBM RS/6000 POWERstation
- IBM Scalable POWERparallel[®] systems
- Bull DPX/20 (RISC)
- Bull ESCALA (SMP)

There must be enough random access memory (RAM) and disk storage for the programming prerequisites specified later in this section, the WebSphere MQ client, the access methods, and the application programs.

Software

The following are the software prerequisites for the WebSphere MQ client for AIX. Minimum supported levels are shown. Later levels, if any, are supported unless otherwise stated.

Operating system:

- AIX V4.3.3, with PTF U472177 (PowerPC 32-bit).
Later levels of some listed products might be required for AIX Version 4.2, SMP, and SP[™]. Later levels of operating system might be required to support corequisite products.
- AIX V5.1 (Power 32 bit only) with PTF U47687 (trace update)

Use the **oslevel** command to determine the level of the operating system you are running.

Connectivity:

- TCP/IP (as part of the base operating system)
- IBM eNetwork[™] Communications Server for AIX, Version 5.0 (for SNA LU 6.2 connectivity)

Optional software

DCE:

- IBM DCE, Version 3.1
- IBM DCE, Version 3.2

Notes:

1. DCE names and security modules for WebSphere MQ are provided as part of the WebSphere MQ for AIX product.
2. If you want to use the DCE send and receive exits supplied with WebSphere MQ, you must use a version of DCE that supports the Data Encryption Standard (DES).
3. A WebSphere MQ client application that uses DCE threads cannot connect to a server queue manager over an SSL channel. Because TXSeries uses DCE threads, this applies to a TXSeries application that runs as a WebSphere MQ client application.

Hardware and software, AIX

Compilers for WebSphere MQ client applications on an AIX system: The following compilers are supported:

- IBM VisualAge® Java Enterprise Edition for AIX, Version 2.0
- IBM Visual Age C for AIX, V5.0
- IBM Visual Age C++ Professional for AIX, V5.0
- Micro Focus Server Express V2.0.10

AT&T GIS UNIX (NCR UNIX) client: hardware and software required

This section outlines the hardware and software requirements for the MQSeries client for AT&T GIS UNIX (NCR UNIX).

Hardware

The MQSeries client can run on the following:

- Any AT&T GIS 34XX, 35XX, or 36XX system with minimum system disk space of 20 MB
- Any LAN adapter
- Any communications hardware supporting SNA LU 6.2 or TCP/IP

Software

The following are the software prerequisites for the MQSeries client for AT&T GIS UNIX (NCR UNIX). Minimum supported levels are shown. Later levels, if any, are supported unless otherwise stated.

Operating system:

- AT&T GIS UNIX SVR4 MP-RAS, Version 3.0, including TCP/IP (this platform has become NCR UNIX SVR4 MP-RAS, R3.0)

Connectivity:

- TCP/IP (as part of the base operating system)
- AT&T GIS SNA Services, Version 2.06 or later Version 2 (for SNA LU 6.2 connectivity)

Optional software

Compiler for WebSphere MQ client applications on an AT&T GIS UNIX (NCR UNIX) system: The following compiler is supported:

- AT&T GIS High Performance C, Version 1.0b

Compaq Tru64 UNIX client: hardware and software required

This section outlines the hardware and software requirements for the MQSeries client for Compaq Tru64 UNIX.

Hardware

The MQSeries client can run only on:

- A desktop or server system capable of running Compaq Tru64 UNIX with a minimum system disk space of 25 MB
- Any communications hardware supporting TCP/IP

Software

The following are the software prerequisites for the MQSeries client for Compaq Tru64 UNIX. Minimum supported levels are shown. Later levels, if any, are supported unless otherwise stated.

Operating system:

- Compaq Tru64 UNIX Version 4.0 or Version 5.0

Connectivity:

- TCP/IP as part of the base operating system

For information about SNA support, see the README file supplied with MQSeries for Compaq Tru64 UNIX.

Optional software

Compilers for WebSphere MQ client applications on a Compaq Tru64 UNIX system: The following compilers are supported:

- Compaq C for Tru64 UNIX (provided as part of the base operating system)
- Compaq C++ for Tru64 UNIX Version 6.2
- Micro Focus COBOL for UNIX Version 4.1B¹ or Version 4.1.00G²
- Java Development Kit for Compaq Tru64 UNIX, Version 1.1.8

1. Supplied by Compaq

2. Supplied by Micro Focus

Compaq OpenVMS Alpha client: hardware and software required

This section outlines the hardware and software requirements for the MQSeries client for Compaq OpenVMS Alpha.

Hardware

The MQSeries client can run on Compaq OpenVMS Alpha systems with minimum system disk space of 700 blks (350 KB) and minimum memory of 8 MB.

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

- Digital SNA Domain Gateway for Synchronous or Channel Transport
- Digital SNA Peer Server
- Any communications hardware supporting TCP/IP or DECnet

Software

The following are the software prerequisites for the MQSeries client for Compaq OpenVMS Alpha. Minimum supported levels are shown. Later levels, if any, are supported unless otherwise stated.

Operating system:

- OpenVMS, Version 6.2

Connectivity:

- Digital SNA APPC LU 6.2 Programming Interface, Version 2.3
- Digital DECnet SNA Gateway software, Version 1.2A
- Process Software TCPWare, Version 5.2-3
- VAX/AXP: DECnet SNA APPC/LU 6.2, Version 2.2
- VAX/AXP: CISCO (formerly TGV) MultiNet, Version 3.5 for OpenVMS
- AXP: TCP/IP Services for OpenVMS AXP, Version 4.0
- Digital TCP/IP Services for OpenVMS (UCX), Version 4.1
- VAX: TCP/IP Services for OpenVMS VAX, Version 3.3
- Attachmate Pathway for OpenVMS, Version 2.5.1

Optional software

- Distributed Computing Environment for OpenVMS, Version 1.3b

Compilers for WebSphere MQ client applications on a Compaq OpenVMS

Alpha system: The following compilers are supported:

- AXP/VAX: DEC C, Version 5.2
- AXP: DEC C++, Version 5.2
- VAX: DEC C++, Version 5.0
- VAX: VAX COBOL, Version 5.3
- AXP: DEC COBOL, Version 2.2

Hardware and software, DOS

DOS client: hardware and software required

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

Hardware

The 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 specified later in this section, the MQSeries client, the access methods, and the application programs.

Software

The following are the software prerequisites for the MQSeries client for DOS. Minimum supported levels are shown. Later levels, if any, are supported unless otherwise stated.

Operating system:

- DOS, Version 5.0

Connectivity:

- TCP/IP (as part of the base operating system)
- SPX

Optional software

- Novell Netware Client for DOS/Win, Version 1.20 and Version 2.5
- Novell LAN Workplace, Version 5.1
- FTP PC/TCP for DOS, Version 5.0

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

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

Compilers for WebSphere MQ client applications on a DOS system: The following compiler is supported:

- Microsoft[®] Visual C++, Version 1.5

HP-UX client: hardware and software required

This section outlines the hardware and software requirements for the WebSphere MQ client for HP-UX.

Hardware

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

Software

The following are the software prerequisites for the WebSphere MQ client for HP-UX. Minimum supported levels are shown. Later levels, if any, are supported unless otherwise stated.

Operating system:

- HP-UX, Version 11
- HP-UX, Version 11i

Connectivity:

- TCP/IP (as part of the base operating system)
- HP SNAplusII (for SNA LU 6.2 connectivity)

Optional software

DCE:

- The version of HP DCE/9000 appropriate for the level of the HP-UX operating system in use

Notes:

1. DCE names and security modules for WebSphere MQ are provided as part of the WebSphere MQ for HP-UX product.
2. A WebSphere MQ client application that uses DCE threads cannot connect to a server queue manager over an SSL channel. Because TXSeries uses DCE threads, this applies to a TXSeries application that runs as a WebSphere MQ client application.

Compilers for WebSphere MQ client applications on an HP-UX system: The following compilers are supported:

- Micro Focus Server Express V2.0.10
- HP-UX ANSI C Compiler
- C Softbench, Version 7.0
- Fully patched HP DCE/9000 application development tools

Linux client: hardware and software required

This section outlines the hardware and software requirements for the WebSphere MQ client for Linux.

Hardware

The WebSphere MQ client for Linux can run on any machine that supports a Linux for Intel or Linux for zSeries operating system. Typical disk space requirements are 45 MB without the WebSphere MQ SSL support and 115 MB with the WebSphere MQ SSL support.

For connectivity, you can use any communications hardware that supports SNA LU 6.2 (Linux for Intel only) or TCP/IP.

Software

The following are the software prerequisites for the WebSphere MQ client for Linux. Minimum supported levels are shown. Later levels, if any, are supported unless otherwise stated.

Operating system: The WebSphere MQ client for Linux can be installed on any distribution of Linux for Intel or Linux for zSeries that supports:

- Linux kernel, Version 2.4
- glibc, Version 2.2
- pthreads, Version 0.7
- libstdc++, Version 3.0, for C++ programming
- Red Hat Package Manager (RPM), for installation

The WebSphere MQ client for Linux has been tested with the following distributions of Linux for Intel:

- Red Hat Linux, Version 7.2 and Version 7.3
- Caldera OpenLinux
- SuSE Linux, Version 7.3
- TurboLinux, Version 7.0

The WebSphere MQ client for Linux has been tested with the following distributions of Linux for zSeries:

- Red Hat Linux, Version 7
- SuSE Linux Enterprise Server, Version 7

Connectivity:

- SNA LU 6.2 (Linux for Intel only), IBM Communications Server for Linux, Version 6.0
- TCP/IP, as part of the base operating system

Optional software

Secure Sockets Layer (SSL): If you want to use the WebSphere MQ SSL support, you need the IBM Global Security Kit (GSKit), Version 6, which is supplied with WebSphere MQ.

Java messaging: If you want to use WebSphere MQ classes for Java or WebSphere MQ classes for Java Message Service (JMS), you need a Java Runtime Environment (JRE), Version 1.3 or later.

Compilers for WebSphere MQ client applications on a Linux system: The following compilers are supported:

- GNU C Compiler
- GNU C++ Compiler:
 - Version 2.95.2 or Version 3, for Linux for Intel
 - Version 2.95.2, for Linux for zSeries
- IBM Developer Kit for Linux, Java 2 Technology Edition, Version 1.3.1

OS/2 Warp client: hardware and software required

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

Hardware

The 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 specified later in this section, the MQSeries client, the access methods, and the application programs.

The system unit must have a CD-ROM device.

Software

The following are the software prerequisites for the MQSeries client for OS/2 Warp. Minimum supported levels are shown. Later levels, if any, are supported unless otherwise stated.

Operating system:

- OS/2 Warp, Version 4.0 (84H1426) (5622-851)
- OS/2 Warp Server, Version 4.0 (25H8002)
- OS/2 Warp Server Advanced SMP feature, Version 4.0
- OS/2 Workspace-on-Demand

Connectivity:

- IBM eNetwork Communications Server for OS/2 Warp, Version 5.0
- Novell Netware Client for OS/2 Warp, Version 1.20 (for direct IPX/SPX support)
- IntraNetWare Client for OS/2 Warp, Version 2.12
- TCP/IP for OS/2 Warp, Version 2.0 base kit plus NetBIOS kit, Version 3.5 (base kit is essential)
- NetWare for OS/2 Warp, Version 4.11

Optional software

- IBM Directory and Security Server for OS/2 Warp, Version 4 or later compatible versions. This must be a DCE product which supports DES data encryption if the user wishes to run the WebSphere MQ-supplied DCE send, receive, or message exits.
- If used as a DCE server this software is known to run adequately in the following environment:
 - On a Pentium[®] processor running 90 MHz or faster
 - On a machine with 64 MB or more of memory
 - Using OS/2 Warp Server, Version 4.0 or later
- WebSphere MQ DCE names and security modules are provided as part of the MQSeries for OS/2 Warp product.

Compilers for WebSphere MQ client applications on an OS/2 Warp system: The following compilers are supported:

- IBM VisualAge COBOL for OS/2 Warp, Version 1.1 and Version 2.2
- Micro Focus COBOL compiler, Version 4.0 (32 bit)
- IBM VisualAge C++ for OS/2 Warp, Version 3.0
- Borland C++ compiler, Version 2.0 and Version 5.02 (C bindings only)
- IBM C and C++ compilers, Version 3.6

Hardware and software, OS/2 Warp

- IBM PL/I for OS/2 Warp, Version 1.2
- IBM VisualAge for PL/I for OS/2 Warp
- IBM VisualAge Java Enterprise Edition for OS/2 Warp, Version 2.0
- IBM VisualAge Java Professional Edition for OS/2 Warp, Version 2.0

SINIX and DC/OSx client: hardware and software required

This section outlines the hardware and software requirements for the MQSeries client for SINIX and DC/OSx.

Hardware

The MQSeries client can run on:

- SINIX: RM200, RM300, RM400, RM600 systems with minimum system disk space of 30 MB
- DC/OSx: MIServer, Nile systems with minimum system disk space of 30 MB
- Any communications hardware supporting SNA LU 6.2 or TCP/IP

Software

The following are the software prerequisites for the MQSeries client for OS/2 Warp. Minimum supported levels are shown. Later levels, if any, are supported unless otherwise stated.

Operating system:

- SINIX operating system: SINIX-N, Version 5.42C10 (for RM200, RM300, RM400) or SINIX-Y, Version 5.42A40 (for RM600)
- DC/OSx operating system, Version 1.1-cd079 or later

Connectivity:

SINIX

- SNA:
 - TRANSIT-SERVER 3.4 (SNA Communication Server Version)
 - TRANSIT-CLIENT 3.4 (SNA Comm. Client / Local Functions)
 - TRANSIT-CPIC 3.4 (SNA LU 6.2 Communication and CPI-C)
- OpenNet TCP/IP
- DCE, Version 1.03A00 or later

DC/OSx

- SNA requires LU 6.2 SW, Version 1.3 and:
 - To support the ISC-2 (Intelligent Synchronous Controller) serial line:
 - Comm Services, Version 1.2
 - ISC with SNA engine, Version 1.3
 - To support the ILC-T (Intelligent LAN Controller, Token ring) interface:
 - Comm Services, Version 1.2
 - Token Ring Mac interface, Version 1.3
 - To support the SNA on the ESCON[®] IBM Channel link:
 - XVI/ESCON Driver 1.0
- TCP/IP, Version 1.0

Optional software

Compilers for WebSphere MQ client applications on a SINIX or DC/OSx system: The following compilers are supported:

SINIX

- C compiler (C-DS, MIPS), Version 1.1
- Micro Focus COBOL, Version 3.2

DC/OSx

- C4.0 compiler, Version 4.0.1
- Micro Focus COBOL, Version 3.2

Solaris client: hardware and software required

This section outlines the hardware and software requirements for the WebSphere MQ client for Solaris.

Hardware

The WebSphere MQ client can run on a:

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

with a minimum system disk space of 25 MB.

Note: Solaris systems from other manufacturers are not supported.

For connectivity, you can use any communications hardware that supports SNA LU 6.2 or TCP/IP.

Software

The following are the software prerequisites for the WebSphere MQ client for Solaris. Minimum supported levels are shown. Later levels, if any, are supported unless otherwise stated.

Operating system:

- Sun Solaris, Version 2.7
- Sun Solaris, Version 2.8

Connectivity:

- TCP/IP (as part of the base operating system)
- SunLink SNA Peer-to-Peer, Version 9.1 (for SNA LU 6.2 connectivity)
- If token-ring is used: SunLink Token Ring Interface /SBus, Version 3.0.2. This requires patch 102463 or Sun TRI/P Adapter, Version 1.0

Optional software

DCE:

- IBM DCE, Version 3.1
- IBM DCE, Version 3.2

Notes:

1. DCE names and security modules for WebSphere MQ are provided as part of the WebSphere MQ for Solaris product.
2. If you want to use the DCE send and receive exits supplied with WebSphere MQ, you must use a version of DCE that supports the Data Encryption Standard (DES).
3. A WebSphere MQ client application that uses DCE threads cannot connect to a server queue manager over an SSL channel. Because TXSeries uses DCE threads, this applies to a TXSeries application that runs as a WebSphere MQ client application.

Compilers for WebSphere MQ client applications on a Solaris system: The following compilers are supported:

- SunWorkShop Compiler C, Version 5.0
- SunWorkShop Compiler C++, Version 5.0
- Forte C 6 (Sun Workshop 6 C)
- Forte C++ 6 (Sun Workshop 6 C++).

Sun Solaris, Intel Platform Edition client: hardware and software required

This section outlines the hardware and software requirements for the MQSeries client for Sun Solaris, Intel Platform Edition.

Hardware

The MQSeries client can run only on:

- Sun Intel desktop or server

with a minimum system disk space of 25 MB.

Note: Solaris systems from other manufacturers are not supported.

For connectivity, you can use any communications hardware that supports SNA LU 6.2 or TCP/IP.

Software

The following are the software prerequisites for the MQSeries client for Sun Solaris, Intel Platform Edition. Minimum supported levels are shown. Later levels, if any, are supported unless otherwise stated.

Operating system:

- Sun Solaris 7 Intel Platform Edition Hardware 11/99, with patches 106981-10, 106542-10, 107637-03, and 107172-05

Connectivity:

- TCP/IP (as part of the base operating system)

Optional software

Compilers for WebSphere MQ client applications on a Sun Solaris, Intel Platform Edition system: The following compilers are supported:

- Sun WorkShop Compiler C, Version 5.0, with patches 107296-01, 107830-02, 107290-03, and 107361-04
- Sun WorkShop Compiler C++, Version 5.0, with patches 106328-08 and 107312-09

Hardware and software, VM/ESA

VM/ESA client: hardware and software required

This section outlines the hardware and software requirements for a MQSeries client for VM/ESA.

Hardware

The MQSeries client can run on any CMS system that supports the programming prerequisites specified later in this section.

Software

The following are the software prerequisites for the MQSeries client for VM/ESA. Minimum supported levels are shown. Later levels, if any, are supported unless otherwise stated.

Operating system:

- VM/ESA, Version 2 Release 3, with LE/370, Release 1.6

Connectivity:

- TCP/IP, Release 2.4
- VTAM® LU 6.2

Optional software

Compilers for WebSphere MQ client applications on a VM/ESA system: The following compilers are supported:

- IBM Assembler
- IBM VS COBOL II
- IBM C for VM, Release 3.1
- IBM OS/PL/I, Release 2.3
- IBM VM/ESA Rexx/VM

Windows client: hardware and software required

This section outlines the hardware and software requirements for the WebSphere MQ client for Windows.

Hardware

The WebSphere MQ client can run on any IBM PC machine, or equivalent, that is based on a 32-bit Intel processor, certified as Windows compatible, and capable of running Windows 98, Windows 2000, Windows NT, or Windows XP. There must be enough random access memory (RAM) and disk storage for the programming prerequisites specified later in this section, the WebSphere MQ client, the access methods, and the application programs.

Software

The following are the software prerequisites for the WebSphere MQ client for Windows. Minimum supported levels are shown. Later levels, if any, are supported unless otherwise stated.

Operating system:

- Microsoft Windows 98.
- Microsoft Windows 2000 with Service Pack 2. This can be any of the following products:
 - Microsoft Windows 2000 Professional
 - Microsoft Windows 2000 Server
 - Microsoft Windows 2000 Advanced Server
- Microsoft Windows NT, Version 4.0 with Service Pack 6a.
- Microsoft Windows XP Professional.

For Windows 98 and Windows NT, you need Microsoft Installer (MSI) 2.0 or later. This is provided on the WebSphere MQ for Windows Server CD-ROM.

Connectivity:

- IBM Communications Server for Windows NT, Version 5.0
- IBM Communications Server for Windows NT and Windows 2000, Version 6.1.1
- Attachmate Extra! Personal Client, Version 6.7
- Attachmate Extra! Enterprise 2000
- Microsoft SNA Server, Version 4.0
- Microsoft Host Integrated Server 2000
- TCP/IP, NetBIOS, and SPX, as part of the base operating system

Optional software

DCE:

- IBM Distributed Computing Environment (DCE), Version 1.1

Notes:

1. DCE names and security modules for WebSphere MQ are provided as part of the WebSphere MQ for Windows product.
2. If you want to use the DCE send and receive exits supplied with WebSphere MQ, you must use a version of DCE that supports the Data Encryption Standard (DES).
3. DCE support is not available for Windows 2000 or Windows XP.

Hardware and software, Windows

Secure Sockets Layer (SSL): If you want to use the WebSphere MQ SSL support on Windows NT, you need Microsoft Internet Explorer 4.0.1 with Service Pack 1, or later.

The following are recommended if you use the WebSphere MQ SSL support:

- A Java Secure Sockets Extension (JSSE) implementation. This is not necessary if you have a supported Java Runtime Environment (JRE), Version 1.4 or later.
- 128-bit cipher strength.

The WebSphere MQ SSL support is not available on Windows 98.

Active Directory: If you want to use Active Directory on Windows NT, you need Microsoft Active Directory Client Extensions (ADCE) for Windows NT. This is provided on the WebSphere MQ for Windows Server CD-ROM.

Java messaging: If you want to use WebSphere MQ classes for Java or WebSphere MQ classes for Java Message Service (JMS), you need a supported Java Runtime Environment (JRE), Version 1.3 or later.

Compilers for WebSphere MQ client applications on a Windows system: The following compilers are supported:

- Microsoft Visual C++, Version 6.0
- IBM VisualAge COBOL Enterprise, Version 2.2
- Micro Focus Net Express, Version 3.0 and Version 3.1
- IBM VisualAge PL/I Enterprise, Version 2.1
- Microsoft Visual Basic, Version 6.0
- IBM Developer Kit for Windows, Java 2 Technology Edition, Version 1.3.0 or later

Other:

- Microsoft Windows Terminal Server feature

Windows 3.1 client: hardware and software required

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

Hardware

The 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 specified later in this section, the MQSeries client, the access methods, and the application programs.

Software

The following are the software prerequisites for the MQSeries client for Windows 3.1. Minimum supported levels are shown. Later levels, if any, are supported unless otherwise stated.

Operating system:

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

Connectivity:

- TCP/IP
- SPX

Optional software

- Novell Netware client for DOS/Win31, Version 1.20
- SunPC NSF, Version 5.1
- OnNet SDK for Windows

Compilers for WebSphere MQ client applications on a Windows 3.1 system:

The following compiler is supported:

- Microsoft Visual C++, Version 1.5

Windows 95 client: hardware and software required

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

Hardware

The 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 specified later in this section, the MQSeries client, the access methods, and the application programs.

Software

The following are the software prerequisites for the MQSeries client for Windows 95. Minimum supported levels are shown. Later levels, if any, are supported unless otherwise stated.

Operating system:

- Windows 95

Connectivity:

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

Optional software

- IBM DCE for Windows 95, Version 1.1

A DCE security module is provided as part of the MQSeries client for Windows 95.

Compilers for WebSphere MQ client applications on a Windows 95 system: The following compilers are supported:

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

Chapter 3. Installing client components from WebSphere MQ products and Version 5 MQSeries products (not z/OS)

This chapter discusses how to install the client components for the following WebSphere MQ Version 5.3 products:

- WebSphere MQ for AIX
- WebSphere MQ for HP-UX
- WebSphere MQ for Linux for Intel
- WebSphere MQ for Linux for zSeries
- WebSphere MQ for Solaris
- WebSphere MQ for Windows

and the following MQSeries Version 5 products:

- MQSeries for Compaq Tru64 UNIX
- MQSeries for OS/2 Warp
- WebSphere MQ for Solaris, Intel Platform Edition

For information on installing WebSphere MQ client components from WebSphere MQ for z/OS, see Chapter 4, “Installing MQSeries clients with other MQSeries products” on page 79.

The products listed above and discussed in this chapter include an easy installation feature that helps you install clients quickly. If you are using another or an earlier MQSeries product, see Chapter 4, “Installing MQSeries clients with other MQSeries products” on page 79.

Each of the products listed above supplies software, including the easy installation feature, for clients on the following platforms:

- AIX
- HP-UX
- OS/2 Warp
- Linux
- Solaris
- Windows

Installing a WebSphere MQ or MQSeries Version 5 client and server system

The following CD-ROMs are supplied with WebSphere MQ Version 5.3:

WebSphere MQ Server CD-ROM for each server platform

Each CD-ROM contains the components that can be installed on a server machine. These components include the WebSphere MQ client for the same platform as the server.

WebSphere MQ Client CD-ROM 1

This CD-ROM contains the components that can be installed on a client machine for each of the following platforms:

- AIX

Installing clients (Version 5)

- HP-UX
- Solaris
- Windows

For each platform except Windows, the CD-ROM contains, in two separate directories, the following sets of client components:

- The client components without the WebSphere MQ SSL support
- The client components with the WebSphere MQ SSL support

Only one set of client components is supplied for Windows, and this set contains all the client components that are needed to use the Windows client with or without the WebSphere MQ SSL support.

WebSphere MQ Client CD-ROM 2

This CD-ROM contains the components that can be installed on a client machine for each of the following platforms:

- Linux for Intel
- Linux for zSeries

For each platform, the CD-ROM contains, in two separate directories, the following sets of client components:

- The client components without the WebSphere MQ SSL support
- The client components with the WebSphere MQ SSL support

The following CD-ROMs are supplied with each of the MQSeries Version 5 products:

MQSeries Server CD-ROM

This CD-ROM contains the components that can be installed on a server machine. These components include the MQSeries client for the same platform as the server.

MQSeries Client CD-ROM

This CD-ROM contains the components that can be installed on a client machine for a number of different platforms, including the platform of the server.

To install WebSphere MQ on a server machine, use the Server CD-ROM for your platform. For installation instructions, see the *Quick Beginnings* book for your platform.

To install WebSphere MQ on a client machine, use the Client CD-ROM that contains the client components for your platform. For installation instructions, see:

- "Installing on AIX" on page 38
- "Installing on Compaq Tru64 UNIX" on page 45
- "Installing on HP-UX" on page 48
- "Installing on Linux" on page 51
- "Installing on OS/2 Warp" on page 56
- "Installing on Solaris" on page 64
- "Installing on Sun Solaris, Intel Platform Edition" on page 67
- "Installing on Windows" on page 69

It is possible to install a WebSphere MQ client on the same machine as a WebSphere MQ server by using the appropriate Server CD-ROM, as explained in "Installing WebSphere MQ clients on the same machine as the server" on page 37.

Installing WebSphere MQ clients on the same machine as the server

To install a WebSphere MQ client on a WebSphere MQ server machine, use the appropriate Server CD-ROM. Use a Client CD-ROM only to install a WebSphere MQ client on a machine that is not a WebSphere MQ server.

You can install a WebSphere MQ client from a Client CD-ROM and later decide to install the WebSphere MQ server on the same machine. Before you can do this, you must remove all the client components from the machine. Then use the appropriate Server CD-ROM to install the server and client components. You cannot install the WebSphere MQ server on a machine that already has client components installed from a Client CD-ROM.

Remember that, even if your client and server reside on the same machine, you still need to define the MQI channel between them. See Chapter 8, “Using channels” on page 127 for details.

Uninstalling WebSphere MQ clients

If you want to remove the WebSphere MQ client files from your system, use the process provided to do this efficiently. Details are given after the installation instructions for each platform.

Installing on AIX

To install the WebSphere MQ client on an AIX system, use WebSphere MQ Client CD-ROM 1, which is supplied with WebSphere MQ Version 5.3.

The WebSphere MQ client is installed into the `/usr/mqm` directory. This *cannot* be changed. However, if you do not have enough space in the `/usr/mqm` file system, follow the procedure given in “Creating another file system for the client” on page 39.

If you have a previous version of the WebSphere MQ for AIX client installed on your system, or if a file system remains from a previous AIX client installation, see “Migrating from an earlier version of MQSeries for AIX” on page 42.

If you plan to install a WebSphere MQ client and server on the same machine, see “Installing WebSphere MQ clients on the same machine as the server” on page 37.

Components for AIX

The components you can install on AIX systems are:

WebSphere MQ client

The WebSphere MQ client code for your UNIX platform.

Samples

Sample application programs.

Support for DCE in samples

The DCE samples support. This should be installed 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.

WebSphere MQ Client for Java

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

AIX Java version 1.1.1

For information about Java runtime see the *WebSphere MQ Using Java* book.

Before installing

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

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

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

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

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

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

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

Creating another file system for the client

If you do not have enough space in the `/usr/mqm` file system, you can do either of the following things:

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

or

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

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

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

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

Creating the `mqm` user ID and group

If you want to create the required IDs yourself, for instance, if you want to set up all security groups before installing client components, follow this procedure before you install the client. You must create both user ID and group as `mqm`.

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

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

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

Set the name field to `mqm`

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

Installing on AIX

```
Smit
  Security & Users
    Users
      Add a User
```

Set the name field to mqm

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:

```
Smit
  Security & Users
    Change a user's Password
```

If you are entering a new user mqm, the old password is mqm

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

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

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

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

Easy installation

This section describes the 'Easy Installation' procedure using WebSphere MQ Client CD-ROM 1. By default, the procedure installs the client components for the WebSphere MQ SSL support. If you do not want to install the client components for the WebSphere MQ SSL support, see the README file supplied with WebSphere MQ for AIX.

Note: If you have a previous version of the WebSphere MQ for AIX client installed on your system, or if a file system remains from a previous AIX client installation, see "Migrating from an earlier version of MQSeries for AIX" on page 42.

1. Log on as root
2. Insert WebSphere MQ Client CD-ROM 1 into the CD-ROM drive
3. Type `cd /cdrom` to change to the cdrom directory
4. Type `xinstallm -ez`

Note: `xinstallm` is part of VSM (Visual System Management), which may not be installed on your system. It is part of the `x11.vsm.rte` fileset. You can use SMIT or the underlying `installp` command as alternatives

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
```

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.

Now go to Chapter 6, “Verifying the installation” on page 113.

Custom installation

This section describes custom installation using the System Management Interface Tool (SMIT). By default, the procedure installs the client components for the WebSphere MQ SSL support. If you do not want to install the client components for the WebSphere MQ SSL support, see the README file supplied with WebSphere MQ for AIX.

Note: If you have a previous version of the WebSphere MQ for AIX client installed on your system, or if a file system remains from a previous AIX client installation, see “Migrating from an earlier version of MQSeries for AIX” on page 42.

You can use SMIT for a custom installation as follows:

1. Log on as root.
2. Go into SMIT and from the shell, type:
`smit`
3. Select the device appropriate for your installation, using this sequence of windows:

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

Alternatively, you can use the fastpath command:

```
smitty install_latest
```

4. Press the **List** button to display the **Single Select List** window.
5. Select **/dev/cd0 (CD-ROM Drive)**
6. Press **Do** to display the parameters for **Install Latest Level**.
7. Press **F4** to get a list of filesets to install.
8. Follow the SMIT instructions to select the components you want to install.
9. Press **Enter**.
10. If you have a previous version of the product on your machine, change the **Automatically install requisite software** to **No** and **Overwrite same or newer versions** to **Yes**.
11. On AIX V4.3.3:
 - Press **OK** to install.

On AIX V5.1

- a. Change “Preview new LICENSE agreements?” to **Yes**, and click **OK** to view the license agreements.
- b. Change “ACCEPT new license agreements?” to **Yes**, and click **OK** to accept the license agreements and install WebSphere MQ.

Installing on AIX

Now go to Chapter 6, “Verifying the installation” on page 113.

Migrating from an earlier version of MQSeries for AIX

If you want to migrate from an MQSeries for AIX Version 5 client to a WebSphere MQ for AIX, V5.3 client, you must first end all MQSeries activity on the target machine, and remove any shared resources that are used by MQSeries. Do this either by shutting down the system and restarting it, or by issuing the **icprn** command to remove the shared resources.

The migration procedure described in this section applies only to migration from MQSeries for AIX Version 5 clients to WebSphere MQ for AIX, V5.3 clients. If you are migrating from an earlier version of MQSeries for AIX, you are advised to uninstall your current version before installing the WebSphere MQ for AIX, V5.3 client.

Migration from MQSeries for AIX Version 5 involves updating any currently installed filesets, and installing any new filesets that might be required.

To update currently installed filesets:

1. Go into SMIT for root authority. From the shell, enter:
`smit`
2. Select the device appropriate for your installation using the following sequence of windows:
 Software Installation and Maintenance
 Install and Update Software
 Update Installed Software to Latest Level (Update All)

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

```
smitty update_latest
```

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

Once all previously installed filesets have been updated to the latest level, you can install any additional filesets. See “Custom installation” on page 41 for more information.

Changes to the installation path

Changes in AIX LPP Version 4 packaging mean that the WebSphere MQ for AIX, V5.3 and MQSeries Version 5 client installs into directory **/usr/mqm**. Previous versions of the product installed into directory **/usr/lpp/mqm**.

Installation of the WebSphere MQ for AIX, V5.3 client fails if a file system mounted as **/usr/lpp/mqm** is detected. If you are migrating from an earlier version and a file system exists for this directory, you will need to do one of the following things before installing the WebSphere MQ for AIX, V5.3 client. Either:

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

or

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

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

If you uninstall your existing client and either delete or move your existing file system, you can then install the WebSphere MQ for AIX, V5.3 client as described in “Custom installation” on page 41 or “Easy installation” on page 40.

However, if you move the old file system to the new installation path, you should then perform the migration installation described in “Migrating from an earlier version of MQSeries for AIX” on page 42.

Note: If you have already symbolically linked a file system to **/usr/lpp/mqm**, installation of the WebSphere MQ for AIX, V5.3 client will destroy the contents of the file system and the symbolic link, leaving an empty file system. If this happens, you are advised to uninstall your existing WebSphere MQ client and either delete the file system or relink it to the new install path of **/usr/mqm**, before installing the WebSphere MQ for AIX, V5.3 client.

The installation process for the WebSphere MQ for AIX, V5.3 client creates a symbolic link from the old install path (**/usr/lpp/mqm**) to the new install path (**/usr/mqm**). Therefore any existing scripts or makefiles that reference the old path are still valid.

Changing the national language

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

First, check the initial locale setting for your machine by typing:

```
smitty mle_cc_cust_hdr
```

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

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

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

Installing on AIX

Translated messages

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

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

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

Removing a WebSphere MQ client from AIX

Use SMIT as follows to remove all the WebSphere MQ client files that were installed.

1. Type `smit`
2. Follow this sequence of windows:
 - Software Installation and Maintenance
 - Software Maintenance and Utilities
 - Remove Installed Software
3. Press the **List** button to display the installed software and select the filesets to remove. The WebSphere MQ client filesets have titles starting `mqm*`.

Migrating to and from the WebSphere MQ SSL support

To upgrade a WebSphere MQ client without the SSL support to one with the SSL support, install the two additional file sets, `gskak.rte` and `mqm.keyman.rte`, from the directory on WebSphere MQ Client CD-ROM 1 that contains the set of client components with the WebSphere MQ SSL support. To downgrade a WebSphere MQ client with the SSL support to one without the SSL support, simply remove these two file sets.

Installing on Compaq Tru64 UNIX

If you plan to install an MQSeries client and server on the same machine, you should use the appropriate MQSeries Server CD-ROM, see “Installing WebSphere MQ clients on the same machine as the server” on page 37.

MQSeries clients for other platforms can be installed from the MQSeries Client CD-ROM. The MQSeries Client CD-ROM also contains other MQSeries components that you might need to install on a workstation other than the server.

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 Compaq Tru64 UNIX 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` and `/var/mqm/errors` file systems.

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

- `/var/mqm`

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. Mount the appropriate CD-ROM by typing the following commands:

- a. For Compaq Tru64 UNIX Version 4.0:

```
mount -r -t cdfs -o noversion,rrip /dev/rz18c /cdrom
```

substituting the name of your CD-ROM device for `rz18c`.

- b. For Compaq Tru64 UNIX Version 5.0:

```
mount -r -t cdfs -o noversion,rrip /dev/disk/cdrom0C /cdrom
```

substituting the number of your CD-ROM device for `0C`.

2. Use the Compaq Tru64 UNIX program `setld` to install the software by carrying out the following procedure:

- a. Type `setld -l /cdrom`
 - b. Press the Enter key.

Installing

- c. You are prompted for a list of components to install. Select the ones you require, including MQSeries Client. If you want to install all the components, select **all**.

The components you can select are:

```
IBM MQSeries Base subset
IBM MQSeries Client subset
IBM MQSeries Java Base subset
IBM MQSeries Java Client subset
IBM MQSeries Manual pages
IBM MQSeries Runtime subset
IBM MQSeries Sample subset
(The language catalog or catalogs of your choice.)
```

The component **MQSeries Client** for Java should be installed only if you have Java 1.1.8 (or later compatible) runtime code on your machine.

For further information on using **setld** to install software packages, see the Compaq Tru64 UNIX documentation, or use the **man setld** command.

Translated messages

Messages in U.S. English are always available. If you require another of the languages that is supported by MQSeries for Compaq Tru64 UNIX, or your message catalogues are in a nonstandard directory, you *must* ensure that your NLSPATH environment variable includes the appropriate directory.

Ensure that your LANG environment variable is set correctly. For example:

```
export LANG=fr_FR.ISO8859-1
```

Uninstalling an MQSeries client from Compaq Tru64 UNIX

If you have previously installed an MQSeries client on your system, you can remove the product using the Compaq Tru64 UNIX commands:

```
setld -i
```

and

```
setld -d
```

List the MQSeries components using the Compaq Tru64 UNIX command:

```
setld -i | grep MQS
```

to list the *component names*. You will see a list like this:

```
MQS_BASE      installed  IBM MQSeries Base subset
MQS_CLIENT    installed  IBM MQSeries Client subset
MQS_JAVABASE   installed  IBM MQSeries Java Base subset
MQS_JAVABINDING installed  IBM MQSeries Java Bindings subset
MQS_JAVACLIENT installed  IBM MQSeries Java Client subset
MQS_LANG_DE_DE installed  IBM MQSeries Language-German catalog
MQS_LANG_ES_ES installed  IBM MQSeries Language-Spanish catalog
MQS_LANG_FR_FR installed  IBM MQSeries Language-French catalog
MQS_LANG_IT_IT installed  IBM MQSeries Language-Italian catalog
MQS_LANG_JA_JP installed  IBM MQSeries Language-Japanese catalog
MQS_LANG_KO_KR installed  IBM MQSeries Language-Korean catalog
MQS_LANG_PT_BR installed  IBM MQSeries Language-Brazilian Portuguese catalog
MQS_LANG_ZH_CN installed  IBM MQSeries Language-Simplified Chinese catalog
MQS_LANG_ZH_TW installed  IBM MQSeries Language-Traditional Chinese catalog
MQS_MAN        installed  IBM MQSeries Manual pages
MQS_RUNTIME    installed  IBM MQSeries Runtime subset
MQS_SAMPLES    installed  IBM MQSeries Sample subset
MQS_SERVER     installed  IBM MQSeries Server subset
```

| Issue the following command to remove a component:

| `setld -d component name`

| Remove the MQS_BASE component *after* the other components have been
| uninstalled.

| If the product installation had been manually altered, it may not completely
| uninstall, and you will need to manually delete the files and directories contained
| in /opt/mqm.

Installing on HP-UX

To install the WebSphere MQ client on an HP-UX system, use WebSphere MQ Client CD-ROM 1, which is supplied with WebSphere MQ Version 5.3.

Note: If you plan to install a WebSphere MQ client and server on the same machine, see “Installing WebSphere MQ clients on the same machine as the server” on page 37.

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

Components for HP-UX

The components you can install on HP-UX systems are:

WebSphere MQ Client

The WebSphere MQ client code for your UNIX platform.

Samples

Sample application programs.

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.

WebSphere MQ Client for Java

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

HP-UX

Java version 1.1.2

For information about Java runtime, see the *WebSphere MQ Using Java* book.

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

Before installation

Before you can install a WebSphere MQ client on your HP-UX system you:

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

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

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

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

After installation, this user ID (mqm) owns the directories and files that contain the resources associated with the product. This group and user must be defined for any machine on which the WebSphere MQ 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 appropriate HP-UX documentation.

1. Log in as root.
2. Insert WebSphere MQ Client CD-ROM 1 into the CD-ROM drive.
3. Mount the CD-ROM drive
 - a. Change directory to /usr/\$bin
 - b. Type `pfs_mountd &`
 - c. Type `pfsd4 &`
 - d. Type `pfs_mount -o xlat=unix /<path to CD-ROM device>/<localdir>`
4. Accept the licence:
 - a. Change directory to the location of the mounted CD-ROM (for example, /cdrom).
 - b. Run the mqlicense.sh script by typing one of the following commands:
 - For the WebSphere MQ client without the WebSphere MQ SSL support:
`./hpux11/MQClient/mqlicense.sh`
 - For the WebSphere MQ client with the WebSphere MQ SSL support:
`./hpux11/MQClientwithSSL/mqlicense.sh`

The license is displayed. If you accept the license, you can continue the installation. If you decline, the message: **The license agreement has not been accepted ...** is displayed, and the installation will fail.

If you are performing a silent or remote install, you can run the installation with the -accept option, so that the license is accepted without being displayed.

5. Type one of the following commands to start the installation procedure:
 - For the WebSphere MQ client without the WebSphere MQ SSL support:
`swinstall -s /cdrom/<localdir>/hpux11/MQClient/mqs530.v11`
 - For the WebSphere MQ client with the WebSphere MQ SSL support:
`swinstall -s /cdrom/<localdir>/hpux11/MQClientwithSSL/mqs530.v11`

Translated messages

Messages in U.S. English are always available. If you require one of the other languages that is supported by WebSphere MQ 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:

Installing on HP-UX

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

Removing a WebSphere MQ client from HP-UX

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

Migrating to and from the WebSphere MQ SSL support

To upgrade a WebSphere MQ client without the SSL support to one with the SSL support, install the two additional file sets, gsk6bas and MQSERIES.MQM-KEYMAN, from the directory on WebSphere MQ Client CD-ROM 1 that contains the set of client components with the WebSphere MQ SSL support. To downgrade a WebSphere MQ client with the SSL support to one without the SSL support, simply remove these two file sets.

Installing on Linux

To install the WebSphere MQ client on a Linux system, use WebSphere MQ Client CD-ROM 2, which is supplied with WebSphere MQ Version 5.3.

The WebSphere MQ client is installed in the /opt/mqm directory. You *cannot* change this directory. If you do not have enough space in the /opt/mqm file system, follow the procedure in “Creating a file system for the components” on page 52.

If you plan to install the WebSphere MQ client and on the same machine as the WebSphere MQ server, see “Installing WebSphere MQ clients on the same machine as the server” on page 37.

Components for Linux

You can install the following components on a Linux client system:

Client The WebSphere MQ client code for the Linux platform.

Runtime

This component provides support for applications. You must install this component before you install the Client component.

SDK This component is needed for application development.

Sample programs

Sample application programs that are needed if you want to verify that the WebSphere MQ client has been installed correctly.

Java messaging

This component provides WebSphere MQ classes for Java and WebSphere MQ classes for Java Message Service (JMS). To use this component, you do not need to install any other component.

Message catalogs

A message catalog in US English is installed automatically. Message catalogs are also available for the following national languages:

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

IBM Global Security Kit (GSKit) V6

This component is needed for the WebSphere MQ SSL support. GSKit contains the key management tool, iKeyman.

WebSphere MQ support for iKeyman

This component is needed for the WebSphere MQ SSL support. You must install this component after you have installed the IBM Global Security Kit (GSKit) V6.

Installing on Linux

Before installation

Before you install the WebSphere MQ client, you must do the following:

1. Create the file systems that are used to store the components and working data.
2. Create the mqm user ID and mqm group.

Creating the file systems

The components are installed in `/opt/mqm` and working data is stored in `/var/mqm`. You cannot change these directories.

Creating a file system for the components: The disk space required for the components depends on how many components you install. Typical disk space requirements are as follows:

Linux for Intel

- Without the SSL support: 27 MB
- With the SSL support: 100 MB

Linux for zSeries

- Without the SSL support: 28 MB
- With the SSL support: 93 MB

If you do not have enough space to store the components in the `/opt/mqm` file system, you can do either of the following:

- Create a new file system and mount it as `/opt/mqm`.
- Create a new directory anywhere on your machine, and create a symbolic link from `/opt/mqm` to this new directory. For example:

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

Whichever of these options you choose, you must do it *before* you install the WebSphere MQ client.

The file system for the components can be a remote network device, using NFS, for example. However, the mount options defined on that device must allow **setuid** programs, including root access, to run.

Creating a file system for the working data: Before you install the WebSphere MQ client, create and mount a file system called `/var/mqm`. In this way, other system activity is not affected if a large amount of working data builds up.

To determine the size of the `/var/mqm` file system for a client, consider:

- The size of the error log files in the `/var/mqm/errors` directory
- The amount of trace data that is written to the `/var/mqm/trace` directory

Typically, allow 15 MB of disk space in the `/var/mqm` file system.

Creating a separate file system for the error log files: You can also create a separate file system, `/var/mqm/errors`, for the error log files. This file system can be NFS mounted but, if you choose to do this, you might lose the error logs if the network fails. Typically, allow 4 MB of disk space for the error log files.

Creating the mqm user ID and mqm group

The mqm user ID, with a primary group of mqm, is created automatically during the installation of WebSphere MQ, unless you are using Caldera OpenLinux. After installation, the mqm user ID owns the directories and files that contain the resources associated with the product.

You can create the mqm user ID and mqm group yourself. You might do this, for example, if you want to configure your security environment before installing WebSphere MQ. If you are installing on Caldera OpenLinux, you *must* create the mqm user ID and mqm group yourself *before* installing WebSphere MQ.

Installation

This installation procedure uses the Red Hat Package Manager (RPM) installer, which allows you to choose the components you want to install. The components are listed in “Components for Linux” on page 51.

1. Log in as root.
2. Mount WebSphere MQ Client CD-ROM 2 on the target machine.
3. Select the set of client components that you want to use for the installation, and change into the corresponding directory on the CD-ROM.

If you select a set of client components without the SSL support, you cannot install the IBM Global Security Kit (GSKit) V6 or the WebSphere MQ support for iKeyman. Table 2 shows each set of client components and its corresponding directory on the CD-ROM.

Table 2. Sets of client components on Client CD-ROM 2

Set of client components	Directory on WebSphere MQ Client CD-ROM 2
Linux for Intel without the SSL support	/linux_intel/MQClient
Linux for Intel with the SSL support	/linux_intel/MQClientwithSSL
Linux for zSeries without the SSL support	/linux_zseries/MQClient
Linux for zSeries with the SSL support	/linux_zseries/MQClientwithSSL

Notes:

- a. If you do not have a locally attached CD-ROM drive, you can copy the contents of the directory you require from a machine that does have a CD-ROM drive to the target machine using, for example, the ftp utility. If you do this, ensure that you copy the entire directory structure for the set of client components you have selected, and that you maintain the same directory structure on the target machine.
You can now install the client components from the local copy of the directory.
 - b. If the machine hosting the CD-ROM is an NFS server, you can mount the contents of the CD-ROM on the target machine using NFS.
4. Run the `mqlicense.sh` script.
The license is displayed. If you accept the license, the installation continues. If you decline, the message:
Product cannot be installed until the license agreement has been accepted...
is displayed, and the installation fails.

Installing on Linux

If you are performing a silent or remote install, you can run the `mqlicense.sh` script with the `-accept` option, so that the license is accepted without being displayed.

5. Use the **rpm -i** command to install a minimum client configuration. For example:

- If you are installing on an Intel machine, enter the following commands, in the order shown, to install the Runtime and Client components:

```
rpm -i MQSeriesRuntime-5.3.0-1.i386.rpm
rpm -i MQSeriesClient-5.3.0-1.i386.rpm
```
- If you are installing on a zSeries machine, enter the following commands, in the order shown, to install the Runtime and Client components:

```
rpm -i MQSeriesRuntime-5.3.0-1.s390.rpm
rpm -i MQSeriesClient-5.3.0-1.s390.rpm
```
- To install only the Java messaging component on an Intel machine, enter the following command:

```
rpm -i MQSeriesJava-5.3.0-1.i386.rpm
```

The WebSphere MQ license notice is shown only when the first component is installed. It is not shown again for subsequent components.

6. Use the **rpm -i** command to install any other components that you require. For example, to install the IBM Global Security Kit (GSKit) V6 and the WebSphere MQ support for iKeyman on a zSeries machine, enter the following commands in the order shown:

```
rpm -i gsk6bas-6.0-n.nn.s390.rpm
rpm -i MQSeriesKeyman-5.3.0-1.s390.rpm
```

Look for the `gsk6bas` package on the CD-ROM to find the value of `n.nn`.

Note: Distributions that do not use the Red Hat Package Manager (RPM) installer by default might generate an error when you install the WebSphere MQ client.

Translated messages

Messages in U.S. English are always available. If you require messages in a different language, ensure that:

1. You install the appropriate message catalog. See “Components for Linux” on page 51.
2. Your `NLSPATH` environment variable includes the appropriate directory. For example, to select messages in German use the following:

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

To find out which language is currently installed, use the **locale** command.

Removing the WebSphere MQ client from Linux

Before you attempt to remove the WebSphere MQ client, check that no WebSphere MQ client application is running on your system.

To remove the WebSphere MQ client, you must first find out the package names of the components currently installed on your system. To list the package names with their version information, enter the following commands:

```
rpm -q -a | grep MQ
rpm -q -a | grep gsk
```

Alternatively, to list the package names without their version information, enter the following commands:

```
rpm -q -a --queryformat "%{NAME}\n" | grep MQ
rpm -q -a --queryformat "%{NAME}\n" | grep gsk
```

To remove a component, with package name MQSeriesSamples for example, enter the following command:

```
rpm -e MQSeriesSamples
```

Some of the components are dependent on others. The **rpm** command does not remove a component if others are dependent on it. For this reason, you must remove the components in an order such that each component you remove has no other component dependent on it. To list all the components on which a specific component depends, MQSeriesClient for example, enter the following command:

```
rpm -q --requires MQSeriesClient
```

Alternatively, remove the components in the order shown in Table 3. Remove only those components that you have installed on your system.

Table 3. Order for removing components

Component	Package name
Message catalogs	MQSeriesMsg_xx ¹
Sample programs	MQSeriesSamples
Client	MQSeriesClient
SDK	MQSeriesSDK
Runtime	MQSeriesRuntime
WebSphere MQ support for iKeyman	MQSeriesKeyMan
IBM Global Security Kit (GSKit) V6	gsk6bas ²
Java messaging	MQSeriesJava
Notes: <ol style="list-style-type: none"> 1. xx identifies the national language. 2. Other IBM products might use the IBM Global Security Kit. 	

After removing the WebSphere MQ client, delete the /var/mqm directory, unless you are migrating to a later release of the WebSphere MQ client.

Migrating to and from the WebSphere MQ SSL support

To upgrade a WebSphere MQ client without the SSL support to one with the SSL support, install the two additional components, IBM Global Security Kit (GSKit) V6 and WebSphere MQ support for iKeyman. To downgrade a WebSphere MQ client with the SSL support to one without the SSL support, simply remove these two components. You do not have to remove any other components.

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 some MQSeries products, or from the IBM SupportPac web site.

Note: If you plan to install an MQSeries client and server on the same machine, see “Installing WebSphere MQ clients on the same machine as the server” on page 37. If you currently have a manually installed MQSeries client on your OS/2 Warp system from a previous release of MQSeries, you must delete it manually before attempting to install the Version 5.1 client. Before you delete the previous version, save your MSQ.INI file because the deletion process will delete this file.

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.

Components for OS/2 Warp

The components you can install on OS/2 Warp systems are:

MQSeries Client

The MQSeries client code for your platform.

MQSeries Development Toolkit

This includes:

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

MQSeries Client for Java

This allows Java applets running on your client machine to communicate with WebSphere MQ. 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 Java 1.1.1 (or later compatible version) runtime code on your machine. The MQSeries client for Java must be installed on an HPFS formatted drive.

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

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 Warp 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.

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

8. If there is not enough space on your hard disk to install all the components, select an option that uses less disk space. If there is too little space on your hard disk for any of the MQSeries for OS/2 Warp 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 might be created when you use the MQSeries client (Working directory).

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

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

Installing on OS/2 Warp

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 Warp 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 Warp desktop, containing objects as follows:
 - READ.ME
 - MQSeries Installation and Maintenance
 - MQSeries Information

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

13. Remove the installation CD-ROM from the drive.
14. If your CONFIG.SYS file has been updated, shut down the system and restart. If the CONFIG.SYS file was not updated, rename the CONFIG.ADD file to CONFIG.SYS before shutting down the system (CONFIG.ADD will be in the same directory as CONFIG.SYS).
15. To obtain MQSeries messages in the correct language two environment variables must be set correctly. The installation process sets the environment variable NLSPATH including <installation directory>\LOCALE\%L\%N, where <installation directory> is the directory in which you choose to install the product. You must set the environment variable LANG=<lang>, where <lang> is the subdirectory in which the message catalog for your language is installed, selecting a language from the list below:

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

Now go to Chapter 6, “Verifying the installation” on page 113.

Unattended installation on OS/2 Warp

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

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

Attention

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

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

Starting unattended installation

The following steps describe how to perform an unattended installation:

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

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

3. Change to the subdirectory samples.
4. Edit the supplied response file, AMQISM2.RSP for client installation, or create a new response file. The supplied response file can be used for install and delete actions. For more information on response files, see "Installation response files" on page 61
5. Type install together with the required parameters (described in "Installation and maintenance parameters" on page 60). For example:

```
INSTALL /A:I /R:J:\INSTMQS\AMQISM2.RSP
        /S:J:\INSTMQS\en_us
        /L1:J:\INSTMQS\MQM.OUT
        /L2:J:\INSTMQS\MQM.HIS /X
```

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

In this example, the MQSeries client is installed from the redirected drive J: on the server, according to the options supplied in the response file AMQISM2.RSP (See "Installation response files" on page 61 for more details). Note that drive J: must be a writeable drive that you must create, and the .RSP file must be put on that drive. The installation log files will also be created on the J: drive. The response file specifies the drive and directory in which to install the client. Errors are logged in MQM.OUT and the history log is contained in MQM.HIS. The /X parameter specifies that the installation is non-interactive.

Installing on OS/2 Warp

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

Installation and maintenance parameters

The installation and maintenance parameters are as follows:

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

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

/A:action

Specifies the action to be performed by the installation program. If you specify this parameter, the main window of the installation program is not displayed.

Valid values for *action* are:

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

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

/G:include path

Specifies the drive and path of a general response file to be included by the specific response file. For more information about response files, see "Installation response files" on page 61.

/L1:error log

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

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

Example:

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

/L2:history log

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

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

Example:

/L2:D:\LOG\INSMQM.HIS

/R:response file

Specifies the drive, path, and file name of a response file; see “Installation response files”.

Example:

/R:L:\MQMINS\AMQISAM2.RSP

/S:source location

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

/T:install target directory

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

/TU:update CONFIG.SYS directory

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

/X Specifies that the installation is fully automatic.

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

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

Installation response files

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

In an installation response file you can specify:

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

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

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

Installing on OS/2 Warp

a specific response file that contains options specific to a particular workstation, and the other a general installation response file that contains options common to all workstations.

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

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

Response file structure

There are two kinds of line in a response file:

- **Comment lines**

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

- **Response lines**

Response lines are used to determine the options and configurations to install on the target system. Response lines have the following syntax:

keyword = value

Keyword-value pairs can be in any order. However, there can be only one pair per line. You can enter keywords in upper or lower case letters. You cannot include spaces within keywords.

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

Keywords for response files

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

CFGUPDATE

Specifies whether CONFIG.SYS is updated automatically. Valid values for this keyword are:

AUTOAutomatically updates CONFIG.SYS

MANUALDoes not update CONFIG.SYS

Actions: INSTALL, UPDATE, DELETE.

COMP

Specifies the name of a component on which to perform an action. See the server or client component lists for valid values.

You do not need to enclose the product names within quotes.

Actions: ALL actions.

DELETEBACKUP

Specifies whether to delete only the backup version of MQSeries for OS/2 Warp or the entire product. Valid values are YES and NO.

When you remove MQSeries for OS/2 Warp, none of your user-supplied information, for example queue manager data, is deleted. This is particularly important if you intend to delete and reinstall the product, because your previous queue manager data will remain.

This can lead to an unexpected directory and file structure, if you attempt to recreate queue managers with identical names to those used in the previous installation, because such data might already be present.

Actions: DELETE.

FILE Specifies the drive and directory for MQSeries for OS/2 Warp code. For example the C:\MQM directory.

Actions: INSTALL.

WORK

Specifies the drive and directory for MQSeries data files. For example the C:\MQM directory.

Actions: INSTALL.

INCLUDE

Specifies which general response files to include with a specific response file. The format of this keyword is:

INCLUDE = *filespec*

Where *filespec* specifies the general response file to be included. If the file specification contains any global characters (* or ?), the first file found that matches the specification is included. If the specification is not valid, no general response file is included.

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

Actions: All actions.

OVERWRITE

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

Actions: INSTALL, UPDATE.

SAVEBACKUP

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

Actions: UPDATE.

Removing an MQSeries client from OS/2 Warp

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

Installing on Solaris

To install the WebSphere MQ client on a Solaris system, use WebSphere MQ Client CD-ROM 1, which is supplied with WebSphere MQ Version 5.3.

Note: If you plan to install a WebSphere MQ client and server on the same machine, see “Installing WebSphere MQ clients on the same machine as the server” on page 37.

Components for Solaris

The components you can install on Solaris systems are:

WebSphere MQ Client

The WebSphere MQ client code for your UNIX platform.

Samples

Sample application programs.

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.

WebSphere MQ Client for Java

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

Solaris

Java version 1.1.1

For information about Java runtime see the *WebSphere MQ Using Java* book.

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

Before installation

Before you can install a WebSphere MQ client on your Solaris system you:

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

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

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

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

After installation, this user ID (mqm) owns the directories and files that contain the resources associated with the product. This group and user must be defined for any machine on which the WebSphere MQ 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

Note: The WebSphere MQ product is installed into the `/opt/mqm` directory. This *cannot* be changed.

Carry out the following procedure:

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

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

If the Volume Manager is running, the CD is mounted on `/cdrom/MQ53Client1` automatically. If it is not running, mount the CD by typing the following commands:

```
mkdir -p /cdrom/MQ53Client1
mount -F hsfs -r /dev/dsk/<cntndnsn> /cdrom/MQ53Client1
```

substituting `<cntndnsn>` with the name of your CD-ROM drive.

2. Run the `mqlicense.sh` script to accept the license by typing one of the following commands:

- For the WebSphere MQ client without the WebSphere MQ SSL support:
`/cdrom/MQ53Client1/solaris/MQClient/mqlicense.sh`
- For the WebSphere MQ client with the WebSphere MQ SSL support:
`/cdrom/MQ53Client1/solaris/MQClientwithSSL/mqlicense.sh`

The license is displayed. If you accept the license, the installation continues. If you do not accept the license, you cannot continue the installation process.

3. Use the Solaris **pkgadd** program to install the WebSphere MQ client software by carrying out the following procedure:

- a. Type one of the following commands:

- For the WebSphere MQ client without the WebSphere MQ SSL support:
`pkgadd -d /cdrom/MQ53Client1/solaris/MQClient/mqs530.img`
- For the WebSphere MQ client with the WebSphere MQ SSL support:
`pkgadd -d /cdrom/MQ53Client1/solaris/MQClientwithSSL/mqs530.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**.

The component **WebSphere MQ Client** for Java should be installed only if you have Java 1.1.1 (or later compatible) runtime code on your machine. You also require Version 2.6 or later of the Solaris operating system.

- c. Press the Enter key.

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

Installing on Solaris

Translated messages

Messages in U.S. English are always available. If you require one of the other languages supported by WebSphere MQ for Solaris, you *must* ensure that your NLSPATH environment variable includes the appropriate directory.

For example:

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

Removing a WebSphere MQ client from Solaris

If you have previously installed WebSphere MQ on your system, you must remove the product using the **pkgrm** program.

If the product is present, but not installed correctly, you might need manually to delete the files and directories contained in:

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

Migrating to and from the WebSphere MQ SSL support

To upgrade a WebSphere MQ client without the SSL support to one with the SSL support, install the image from the directory on WebSphere MQ Client CD-ROM 1 that contains the set of client components with the WebSphere MQ SSL support. When you are asked whether you really want to install the image, answer “yes”.

To downgrade a WebSphere MQ client with the SSL support to one without the SSL support, remove all the components of the client and install the client again, this time using the set of client components without the WebSphere MQ SSL support.

Installing on Sun Solaris, Intel Platform Edition

Install the MQSeries for Sun Solaris, Intel Platform Edition client from the server CD-ROM.

Before Installation

Before you can install a client on your Sun Solaris, Intel Platform Edition system, you:

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

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

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

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

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

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

Installation

Carry out the following procedure:

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

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

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

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

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

2. Use the Sun Solaris, Intel Platform Edition **pkgadd** program, to install the MQSeries client software by carrying out the following procedure:
 - a. Type `pkgadd -d /cdrom/mqclient/solaris/mqs510.img`.
 - b. You are prompted for a list of components to be installed. Select the ones you require - if you want to install all the components, select **all**.
The component **MQSeries Client** for Java should be installed only if you have Java 1.1.8_09a (Year 2000 compatible) runtime code on your machine.
 - c. Press the Enter key.

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

Installing on Sun Solaris, Intel Platform Edition

Uninstalling the Sun Solaris, Intel Platform Edition client

Use the Sun Solaris, Intel Platform Edition **pkgrm** command to uninstall the product. For further information on using **pkgrm** to uninstall software packages, see the Sun Solaris documentation.

Kernel Configuration

See the WebSphere MQ product family web site for a SupportPac that gives additional performance information.

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, Intel Platform Edition, you *must* ensure that your NLSPATH environment variable includes the appropriate directory.

For example:

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

Installing on Windows

See the following sections for information on how to install the WebSphere MQ client on Windows:

- “Preparing to install the WebSphere MQ client”
- “Installing the WebSphere MQ client”
- “Installing from a LAN” on page 72
- “Unattended (silent) installation” on page 73
- “Using Microsoft System Management Server” on page 75
- “Uninstalling a WebSphere MQ client” on page 76

To install the WebSphere MQ client on Windows, use WebSphere MQ Client CD-ROM 1, which is supplied with WebSphere MQ Version 5.3.

Preparing to install the WebSphere MQ client

Before you install, you can decide what type of installation you require. Table 4 shows the installation types available, and the features that are installed with each option. For the prerequisites required for each feature, see “Windows client: hardware and software required” on page 31.

Table 4. Features installed with each type of installation

Installation type	Features installed	Comments
Typical	<ul style="list-style-type: none"> • Windows Client • Development Toolkit 	The default option. Features are installed to default locations.
Compact	<ul style="list-style-type: none"> • Windows Client only 	The feature is installed to the default location.
Custom	By default, the following features are preselected: <ul style="list-style-type: none"> • Windows Client • Development Toolkit 	All the available features are listed and you can select which ones to install, and where to install them.

Installing the WebSphere MQ client

To install a WebSphere MQ client, you must be logged on to Windows as an administrator.

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

Typical client installation

The following instructions assume that you are installing the WebSphere MQ client using WebSphere MQ Client CD-ROM 1, which is supplied with WebSphere MQ Version 5.3. If you plan to install a WebSphere MQ client and server on the same machine, see the notes in the previous section of this chapter.

1. Insert WebSphere MQ Client CD-ROM 1 into the CD-ROM drive.

If autorun is enabled, the installation process starts. If it is not, double-click the **Setup** icon in the root folder on the CD-ROM to start the process.

The Select Setup Language window is displayed.

Installing on Windows

2. On the Select Setup Language window, select the national language of your choice from the list, then click **OK**.

The WebSphere MQ Client Setup window is displayed.

3. Click **Next** to continue.

If the current version of WebSphere MQ client is already installed, the Program Maintenance panel is displayed with two options: Modify or Remove.

- a. If you select Modify, see “Modifying the client installation” on page 71.
- b. If you select Remove, see “Uninstalling WebSphere MQ client using the installation process” on page 76.

If the current version of WebSphere MQ client is not installed, the License Agreement panel is displayed.

4. Read the information and license terms on the panel.

To change the language that the license agreement is displayed in, click **Change Language** then select the language you require from the list provided. Select the option to accept the license terms, then click **Next**.

5. If there was no previous version of this product installed on the machine, the Setup Type panel is displayed.

Select the type of installation you want, then click **Next**. Table 4 on page 69 shows the installation types and the features that are installed with each option.

- a. If you select Custom, go to the procedure “Custom client installation” on page 71.
- b. If you select Typical or Compact, go to step 7.

6. If there was a previous version of MQSeries installed on the machine, the Type of Installation Process panel is displayed. Select one of the following options, then click **Next**:

- Update. Installs the same features as the previous version. Go to the next step.
- Custom. You can select which features to install.
If you select this option, a Destination Folders panel for data files is displayed, then the Features panel is displayed. Follow the procedure “Custom client installation” on page 71 from step 3 or 4 as appropriate.

7. The WebSphere MQ Client Setup window displays a summary of the installation you selected.

To continue, click **Install**.

8. Wait until the progress bar is complete.

When the WebSphere MQ client is successfully installed, the WebSphere MQ Client Setup window displays the following message:

Installation Wizard Completed Successfully

Click **Finish** to close the window.

9. The installation of the WebSphere MQ client is now complete. Note that WebSphere MQ clients are sets of services and do not have to be explicitly run.

10. You now need to verify that the client was installed successfully (see Chapter 6, “Verifying the installation” on page 113).

Compact client installation

Follow the steps for a typical client installation, as described in “Typical client installation” on page 69. The only difference is that, at step 5 on page 70, you select **Compact** on the **Setup Type** window. This installs only the Client feature of WebSphere MQ for Windows.

Custom client installation

During custom installation, you can choose the destination folders for program files and data files. However, after installation, you cannot change these (except by removing the product, then reinstalling). Therefore, plan and select your destination folders carefully.

1. Follow steps 1 to 5 of the “Typical client installation” on page 69.
2. At step 5, select **Custom** on the **Setup Type** window.
3. The Destination Folder panel is displayed.

To accept the default folder for the program files, select **Next**.

To change the folder for the program files, select **Change**, select the required folder in the resulting dialog box, select **OK**, then select **Next**.
4. The Destination Folders panel is displayed.

To accept the default folder for the data files, select **Next**.

To change the folder for the data files, select **Change**, select the required folder in the resulting dialog box, select **OK**, then select **Next**.

If you want to install either the Client or the Java Messaging feature, you require a data files folder. Otherwise, you can ignore this panel (that is, accept the default).
5. The Features panel is displayed.
6. To change the installation of a feature:
 - a. Click the symbol to the left of the feature name to display a drop-down menu.
 - b. Select the required option from:
 - Install this feature
 - Install this feature and all its subfeatures (if any)
 - Do not install this feature (remove if already installed)

The symbol to the left of the feature name changes to show the current installation option. For more information, click **Help** to display the Custom Setup Tips page, which explains the icons used in the feature list.
7. Optionally, to check that there is enough disk space, press the **Space bar**.

The Disk Space Requirements panel is displayed. This shows the disk space available and the amount of disk space that your current selections will take. It highlights any volumes that do not have enough disk space.

To close the panel and return to the Features panel, click **OK**.
8. When your selections are complete, click **Next**.
9. Follow from step 7 on page 70 to the final step of the procedure.

Modifying the client installation

You modify the installation when WebSphere MQ for Windows, V5.3 client is installed and you want to remove or install some WebSphere MQ client features.

1. Insert WebSphere MQ Client CD-ROM 1 into the CD-ROM drive.
2. If autorun is installed, the installation process starts.

Otherwise, double-click on the Setup icon in the root folder of the CD-ROM to start the installation process.

Installing on Windows

The Program Maintenance panel is displayed.

3. Select **Modify**, then click **Next**.

The Features panel is displayed.

4. To change the installation of a feature:

- a. Click on the symbol to the left of the feature name to display a drop-down menu.
- b. Select the required option from:
 - Install this feature
 - Install this feature and all its subfeatures (if any)
 - Do not install this feature (remove if already installed).

The symbol to the left of the feature name changes to show the current installation option.

5. When your selections are complete, click **Next**.

6. The WebSphere MQ Client Setup window displays a summary of the installation you selected.

To continue, click **Modify**.

7. Wait until the progress bar is complete.

When the WebSphere MQ client is successfully installed, the WebSphere MQ Client Setup window displays the following message:

Installation Wizard Completed Successfully

Click **Finish** to close the window.

Modifying the client installation using Add/Remove Programs

1. From the Windows task bar, select **Start**→ **Settings**→ **Control Panel**.

2. Select **Add/Remove Programs**.

3. Select **IBM WebSphere MQ**.

4. For Windows 2000 or Windows XP, select **Change**.

The WebSphere MQ Setup window with the Program Maintenance panel is displayed. Follow the procedure for modifying the installation using the process from step 3 to the end.

5. For Windows 98 or Windows NT, select **Add/Remove**.

The WebSphere MQ Setup window with the Program Maintenance panel is displayed. Follow the procedure for modifying WebSphere MQ using the process from step 3 to the end.

Other methods of installing the WebSphere MQ client

This section contains instructions on how to install the WebSphere MQ client from a LAN, and how to install the WebSphere MQ client using System Management Server (SMS).

Installing from a LAN

There are two ways to put WebSphere MQ installation files on a LAN server for easier access:

- You can make the drive, into which WebSphere MQ Client CD-ROM 1 is inserted, shareable.
- You can copy the installation files from the CD-ROM to a server. To do this, use the following steps:

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

```
md m:\instmqc
```

Installing client from a LAN

2. Load WebSphere MQ Client CD-ROM 1. If autorun is enabled, the WebSphere MQ language_selection window is displayed. Select **Cancel** to close this window.
3. Copy the entire CD-ROM to the installation folder. For example:

```
xcopy e:\*. * m:\instmqc /e
```
4. Give all licensed users access to the folder that now contains the CD-ROM image (in this example, the m: drive).
5. From a command prompt on the target machine, type the following:

```
\\servername\installation_folder\setup.exe
```

where *servername* is the name of the server and *installation_folder* is the full path of the installation folder.

Alternatively:

- a. Map the shared resource to a drive letter. You can use the net use command, or the Windows Explorer.
 - b. Change to the installation folder.
 - c. Type setup, then press Enter.
6. Follow the prompts.

Unattended (silent) installation

WebSphere MQ for Windows client is installed using the Microsoft Installer (MSI). You can invoke MSI directly, without using setup.exe.

This means that you can install WebSphere MQ on a machine without interaction. This process is called unattended (or silent) installation, and is particularly useful for installing WebSphere MQ over a network on a remote machine, because you can install from a shared drive on a LAN server.

To do this, you can invoke MSI with a parameter that calls a response file. A response file is an ASCII text file that contains the parameter values you want to set for the installation.

The machine on which you want to install must be able to share the WebSphere MQ Server CD-ROM, or a copy of the files on it, and you must be able to execute a command on that machine.

Notes:

1. The response file you use to install WebSphere MQ for Windows using WebSphere MQ Client CD-ROM 1 is *not* the same as the one used with earlier non-MSI versions of MQSeries. For details about the response file you use with WebSphere MQ Client CD-ROM 1, see “Unattended (silent) installation”.
2. There are several other methods to invoke MSI without setup.exe. For details, see “Advanced installation methods” on page 75.

To invoke a silent installation using a response file, you use the Msiexec command.

The response file is an ASCII text file, with a format similar to a Windows .ini file, that contains the stanza **[Response]**. This stanza contains parameters that the Msiexec command can use, in the form of PROPERTY=value pairs. The Msiexec command ignores any other stanzas in the file. An example response file, Response.ini, is supplied with WebSphere MQ. This file contains default installation parameters.

Unattended installation

There are three ways to create a response file for installation:

- Copy and edit the file Response.ini that is supplied on WebSphere MQ Client CD-ROM 1, using an ASCII file editor.
- Create your own response file using an ASCII file editor.
- Use an advanced method to invoke an installation and specify the SAVEINI property (and optionally, the ONLYINI property) to generate a response file that contains the same installation options. For more information, see *WebSphere MQ for Windows, V5.3 Quick Beginnings*.

In the response file, all text is in English, and comments begin with a ; character.

Invoking a silent installation: To invoke a typical silent installation, enter the following command at a command line:

```
Msiexec /i "path\MSI\IBM WebSphere MQ.msi" /q  
TRANSFORMS=:1033
```

where:

/q Specifies a silent installation.

TRANSFORMS=:1033 specifies that the installation is in US English. For further information about installing in different national languages, see *WebSphere MQ LotusScript Extension*.

You can also specify PROPERTY=value pairs on the command line (the property must be in upper case), for example:

```
Msiexec /i "path\MSI\IBM WebSphere MQ.msi" /q ADDLOCAL="JavaMsg"  
TRANSFORMS=:1033 AGREETOLICENSE="yes"
```

- PROPERTY strings must be in upper case.
- Value strings are not case sensitive, except for feature names. They can be enclosed in double quotation marks. If a value string includes a blank, it must be enclosed in double quotation marks.
- For a property that can take more than one value, use the format:
ADDLOCAL="Client,JavaMsg"

See *WebSphere MQ for Windows, V5.3 Quick Beginnings* for details of the features that can be values for the ADDLOCAL and REMOVE properties.

Table 5. Valid feature names

Feature Name	Description
Client	The WebSphere MQ for Windows client.
JavaMsg	The files needed for messaging using Java (includes Java Message Service support).
Toolkit	Sample WebSphere MQ program source and sample executable code.

An example of a typical response file is:

```
[Response]  
PGMFOLDER="c:\mqm"  
DATFOLDER="c:\mqm\data"  
AGREETOLICENSE="yes"  
ADDLOCAL="Client"  
REMOVE="JavaMsg,Toolkit"
```


Advanced installation methods

WebSphere MQ for Windows is installed using the Microsoft Installer (MSI). It is possible to install WebSphere MQ by invoking MSI directly, without using setup.exe. You can use this process for more complex unattended (or silent) installation, or for interactive installation, from a command line. For more information, see *WebSphere MQ for Windows, V5.3 Quick Beginnings*.

Using Microsoft System Management Server

There are two major steps to install WebSphere MQ using the Microsoft System Management Server (SMS):

1. Create an SMS software package (see “Creating the WebSphere MQ SMS software package”).
2. Create an SMS job to distribute and install the package (see “Creating the WebSphere MQ SMS job” on page 76).

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

Creating the WebSphere MQ SMS software package: To create the SMS software installation package:

1. From the Microsoft SMS Administrator application, open the **Packages** folder and create a new package.
2. In the SMS **Package Properties** dialog, click on the **Import** button to create the software package by importing a Package Definition File (PDF).
3. In the **File Browser** dialog, select the drive where WebSphere MQ Client CD-ROM 1 is located.
4. Select the current root folder, which contains the package definition file WebSphere MQ.pdf.
You can also find the WebSphere MQ.pdf file in the local drive, or shared network drive to where you copied the WebSphere MQ Installation software.
5. Select the **WebSphere MQ.pdf** file and click on the **OK** button.
6. Click on the **Workstation** button. In the **Source Directory** entry field, specify the fully qualified path name to the WebSphere MQ root folder that contains the WebSphere MQ installation software.
7. Select the appropriate Workstation Command Line:
 - **Automated Uninstallation of IBM WebSphere MQ client**
 - **Automated Installation of IBM WebSphere MQ client (US English)**
8. Click on the **Properties** button for each process and review the **Command Line** entry field to ensure that the parameters are what you require.
9. Click on the **Close** button to close the **Workstation Properties** dialog.

Note: If you specified a local path in the **Source Directory** entry field, you will get a pop-up dialog warning you that the local path you specified might not be accessible to SMS components running on another machine. Click on the **OK** button to continue.

10. Click on the **OK** button to close the **Package Properties** window.
A pop-up dialog appears indicating that SMS will update the software package at all sites. Click the **OK** button to continue.

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

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

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

Notes:

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

Uninstalling a WebSphere MQ client

This section describes how to uninstall (remove) a WebSphere MQ client if you installed it using WebSphere MQ Client CD-ROM 1.

You can uninstall (remove) WebSphere MQ client in attended mode or unattended (silent) mode.

Before you uninstall WebSphere MQ client, ensure that there are no WebSphere MQ client programs running.

Uninstalling WebSphere MQ client from Windows

There are three ways to uninstall WebSphere MQ from your machine:

- Start the installation process, then select the appropriate option.
- Use the Add/Remove Programs facility in the Windows Control Panel.
- Perform a removal from the command line.

You can use these methods to uninstall the WebSphere MQ client, as long as the original installation used the WebSphere MQ Client CD.

You can also uninstall WebSphere MQ client by using the appropriate parameters with advanced installation methods, or by using Microsoft System Management Server (SMS). See “Advanced installation methods” on page 75.

Uninstalling WebSphere MQ client using the installation process: This procedure uninstalls WebSphere MQ from your machine in attended mode. It removes all the currently installed features.

1. Insert WebSphere MQ Client CD-ROM 1 into the CD-ROM drive.
2. If autorun is installed, the installation process starts.
Otherwise, double-click the **Setup** icon in the root folder of the CD-ROM to start the installation process.
The Program Maintenance panel is displayed.
If this panel is not displayed, WebSphere MQ for Windows, V5.3 is not installed on this machine.
3. Select **Remove**, then click **Next**.
4. The Remove WebSphere MQ panel is displayed, with a summary of the installation to be removed.
Click **Remove** to continue.
5. The Removing WebSphere MQ panel is displayed.

Wait for the progress bar to complete.

If there are any messages that state that locked files are found, ensure that there are no WebSphere MQ client programs running.

Uninstallation should then continue.

6. The WebSphere MQ Setup window displays the following message:

Uninstallation Completed Successfully

Click **Finish**.

Uninstalling WebSphere MQ client using Add/Remove Programs:

1. From the Windows task bar, select **Start**→ **Settings**→ **Control Panel**.
2. Select **Add/Remove Programs**.
3. Select **IBM WebSphere MQ**.
4. For Windows 2000 or Windows XP, do one of the following:
 - Select **Remove**. When a confirmation prompt is displayed, select **Yes**.
The uninstall program begins. All the WebSphere MQ files are removed.
 - Select **Change**. The WebSphere MQ Setup window with the Program Maintenance panel is displayed. Follow the procedure for uninstalling WebSphere MQ using the process from step 3 on page 76 to the end.
5. For Windows 98 or Windows NT, select **Add/Remove**.
The WebSphere MQ Setup window with the Program Maintenance panel is displayed. Follow the procedure for uninstalling WebSphere MQ using the process from step 3 on page 76 to the end.

Uninstalling WebSphere MQ using the command line: This procedure can be used for removing the WebSphere MQ files in unattended (silent) mode.

To invoke an uninstallation, you use the Msiexec command.

To uninstall all WebSphere MQ client features, enter one of the following commands:

- Msiexec /i "*path*\MSI\IBM WebSphere MQ.msi" REMOVE="All"
This command gives you an interactive uninstallation of all features.
- Msiexec /i "*path*\MSI\IBM WebSphere MQ.msi" /q REMOVE="All"
This command invokes a silent uninstall of all features.
- Msiexec /x "*path*\MSI\IBM WebSphere MQ.msi"
This command displays only a progress dialog whilst uninstalling all features.
- Msiexec /x "*path*\MSI\IBM WebSphere MQ.msi" /q
This command invokes a silent uninstall of all features.

Alternatively, you can use the Msiexec command with a parameter that calls a response file. A response file is an ASCII text file that contains the parameter values you want to set for the uninstallation. The response file has a format similar to a Windows .ini file, and contains the stanza **[Response]**. This stanza contains parameters that the Msiexec command can use, in the form of PROPERTY=value pairs. The Msiexec command ignores any other stanzas in the file.

You can set which features to uninstall.

Note: The response file you use to uninstall WebSphere MQ for Windows when it was installed using WebSphere MQ Client CD-ROM 1 is *not* the same as the one used with earlier non-MSI versions of WebSphere MQ. For details about

Uninstalling WebSphere MQ

the response file you use with WebSphere MQ Client CD-ROM 1, see
“Unattended (silent) installation” on page 73.

To uninstall WebSphere MQ using a response file, enter the following command:

```
Msixexec /i "path\MSI\IBM WebSphere MQ.msi" /q USEINI="response_file"
```

where *response_file* is the file that contains the [Response] stanza and the required
PROPERTY=value pairs.

An example of a typical uninstallation response file is:

```
[Response]  
REMOVE="Client,JavaMsg"
```

Chapter 4. Installing MQSeries clients with other MQSeries products

Note: In this chapter only:

- The term *UNIX* denotes any of the following UNIX operating systems:
 - AT&T GIS UNIX (NCR UNIX)
 - DC/OSx
 - SINIX
- The term *MQSeries on UNIX systems* means:
 - MQSeries for AT&T GIS UNIX
 - MQSeries for SINIX and DC/OSx

This chapter describes how to install the client components for the following MQSeries products:

- MQSeries for AT&T GIS UNIX V2.2.1
- MQSeries for Compaq OpenVMS Alpha V5.1
- MQSeries for SINIX and DC/OSx V2.2.1

If you are using a WebSphere MQ or MQSeries product not listed in this chapter, see Chapter 3, “Installing client components from WebSphere MQ products and Version 5 MQSeries products (not z/OS)” on page 35.

You install the MQSeries client and server system in two parts: one for the MQSeries server on your server machine and one for the MQSeries client on your client machine. You can install the MQSeries client files from your MQSeries product, as explained here, or from an IBM Transaction Processing SupportPac[™]. See “Obtaining WebSphere MQ clients from IBM Transaction Processing SupportPacs” on page 81 for more information about this.

This chapter also describes how to connect WebSphere MQ clients on other platforms to the following products:

- WebSphere MQ for z/OS
- MQSeries for Compaq NonStop Kernel V5.1
- MQSeries for VSE/ESA Version 2 Release 1

WebSphere MQ for z/OS, MQSeries for Compaq NonStop Kernel, and MQSeries for VSE/ESA can accept connections from WebSphere MQ clients on other platforms. A WebSphere MQ client cannot run on z/OS, Compaq NonStop Kernel, or VSE/ESA, and the files for WebSphere MQ clients are not supplied with these products. If you want to connect WebSphere MQ clients with these platforms, install the WebSphere MQ clients from another WebSphere MQ product, or from an IBM Transaction Processing SupportPac (see “Obtaining WebSphere MQ clients from IBM Transaction Processing SupportPacs” on page 81).

If you are installing the MQSeries client for VM/ESA, you do not need the installation procedure detailed for the other platforms. See “MQSeries client for VM/ESA” on page 88.

This chapter discusses the following topics:

Installing clients (other MQSeries products)

- “Obtaining MQSeries clients from the MQSeries products”
- “Obtaining WebSphere MQ clients from IBM Transaction Processing SupportPacs” on page 81
- “Installing the MQSeries server” on page 81
- “Installing MQSeries clients from MQSeries for Compaq OpenVMS Alpha” on page 82
- “Installing MQSeries clients from MQSeries on UNIX systems” on page 85
- “MQSeries client for VM/ESA” on page 88
- “Changing the config.sys and autoexec.bat files” on page 88

Obtaining MQSeries clients from the MQSeries products

The way you install the MQSeries client files depends on whether your client platform is the same as, or different from, the server platform.

MQSeries client and server on the same platform

If your MQSeries client platform is the same as your server platform, you can install both of them in the normal way, directly from the media (diskette, CD-ROM, or tape, according to your platform). For information, see the *System Management Guide* for your platform.

Alternatively, you can install the MQSeries client from a LAN server on which you have already installed the base product and server (this is not recommended for a WebSphere MQ client on Windows NT). Use FTP or a similar method to do this. For NetView[®] instructions, see the *System Management Guide* for your platform.

Information about how to copy the required MQSeries client files to the client machine, including the file names and directories, are given in this chapter.

MQSeries client and server on different platforms

Depending on which server platform you are using, a set of clients might be supplied with the base product and server. Clients for DOS, OS/2 Warp, and Windows 3.1 are supplied with MQSeries for Compaq OpenVMS Alpha, MQSeries for SINIX and DC/OSx, and MQSeries for AT&T GIS UNIX. These clients are supplied in addition to the files for the MQSeries client on the same platform as your server.

Install MQSeries on the server machine in the normal way, directly from the media (diskette, CD-ROM, or tape, according to your platform). If your server platform is Compaq OpenVMS Alpha or a UNIX system, you can also install the MQSeries client files that you need for your other platform or platforms at the same time. Then you copy the required MQSeries client files to the build and run environment on the client machine. Information about how to do this, including the file names and directories, are given in this chapter.

WebSphere MQ clients cannot run on z/OS, Compaq NonStop Kernel, or VSE/ESA. No WebSphere MQ client files are provided on these platforms (see “z/OS, Compaq NonStop Kernel, and VSE/ESA” on page 81).

Obtaining WebSphere MQ clients from IBM Transaction Processing SupportPacs

The WebSphere MQ client files can be copied from the IBM Transaction Processing SupportPacs for use as needed.

The IBM Transaction Processing SupportPacs library consists of material that complements the family of CICS® and MQSeries products marketed by IBM.

WebSphere MQ client software is available at no charge and is subject to the IPLA and License Information terms defined when requesting the WebSphere MQ clients on the Internet. You have the right to make as many copies of the WebSphere MQ client as necessary.

Installing the MQSeries server

Install the MQSeries base product and server on the machine you want to use as your MQSeries server. Full instructions are given in the *System Management Guide* for your platform.

For WebSphere MQ for z/OS, install the WebSphere MQ base product and client attachment feature on the machine you want to use as your WebSphere MQ server. Full instructions are given in the *WebSphere MQ for z/OS Program Directory*.

MQSeries client files on the server

During the installation of the server, you might be able to include MQSeries client files for other platforms. This depends on your server platform, as explained below:

Compaq OpenVMS Alpha and UNIX systems

When you install MQSeries on Compaq OpenVMS Alpha or a UNIX system, the files for DOS, OS/2 Warp, and Windows 3.1 clients are supplied. Files for MQSeries clients on the same platform as the server are also supplied.

On the server, select the MQSeries client or clients that you require on the appropriate menu during the installation. Then continue with the installation of the server. The MQSeries client files are then copied onto the server, ready for you to copy onto your client machine, as described in this chapter.

z/OS, Compaq NonStop Kernel, and VSE/ESA

You can connect WebSphere MQ clients on other platforms to WebSphere MQ for z/OS, and to MQSeries for Compaq NonStop Kernel and VSE/ESA. WebSphere MQ clients cannot be run on these systems, so the files for WebSphere MQ clients are not supplied with them.

If you have only these WebSphere MQ products, and you want to install WebSphere MQ clients on other platforms, see “Obtaining WebSphere MQ clients from IBM Transaction Processing SupportPacs”.

If you have another MQSeries product available, you might be able to install the MQSeries client files you require from that product, as described in the following sections of this chapter.

Installing MQSeries clients from MQSeries for Compaq OpenVMS Alpha

When you have included the MQSeries client files in the installation of your MQSeries for Compaq OpenVMS Alpha server machine, the files are located in the following directories:

Compaq OpenVMS Alpha files

```
SYS$LIBRARY
SYS$SYSTEM
SYS$COMMON:[SYSHLP.EXAMPLES.MQSERIES]
```

OS/2 Warp files

```
SYS$COMMON:[SYSHLP.EXAMPLES.MQSERIES.OS2_CLIENT.BIN]
SYS$COMMON:[SYSHLP.EXAMPLES.MQSERIES.OS2_CLIENT.DLL]
SYS$COMMON:[SYSHLP.EXAMPLES.MQSERIES.OS2_CLIENT.INC]
SYS$COMMON:[SYSHLP.EXAMPLES.MQSERIES.OS2_CLIENT.LIB]
SYS$COMMON:[SYSHLP.EXAMPLES.MQSERIES.OS2_CLIENT.MSG]
SYS$COMMON:[SYSHLP.EXAMPLES.MQSERIES.OS2_CLIENT.SAMP.BIN]
```

DOS files

```
SYS$COMMON:[SYSHLP.EXAMPLES.MQSERIES.DOS_CLIENT.BIN]
SYS$COMMON:[SYSHLP.EXAMPLES.MQSERIES.DOS_CLIENT.INC]
SYS$COMMON:[SYSHLP.EXAMPLES.MQSERIES.DOS_CLIENT.LIB]
SYS$COMMON:[SYSHLP.EXAMPLES.MQSERIES.DOS_CLIENT.MSG]
SYS$COMMON:[SYSHLP.EXAMPLES.MQSERIES.DOS_CLIENT.SAMP]
```

Windows 3.1 files

```
SYS$COMMON:[SYSHLP.EXAMPLES.MQSERIES.WIN_CLIENT.BIN]
SYS$COMMON:[SYSHLP.EXAMPLES.MQSERIES.WIN_CLIENT.DLL]
SYS$COMMON:[SYSHLP.EXAMPLES.MQSERIES.WIN_CLIENT.INC]
SYS$COMMON:[SYSHLP.EXAMPLES.MQSERIES.WIN_CLIENT.LIB]
SYS$COMMON:[SYSHLP.EXAMPLES.MQSERIES.WIN_CLIENT.MSG]
SYS$COMMON:[SYSHLP.EXAMPLES.MQSERIES.WIN_CLIENT.SAMP.BIN]
```

Note: This assumes that the logical name MQS_EXAMPLES is assigned to SYS\$COMMON:[SYSHLP.EXAMPLES.MQSERIES].

Installing an MQSeries client on Compaq OpenVMS Alpha from MQSeries for Compaq OpenVMS Alpha

If possible, do this by installing the MQSeries client directly from the media supplied with your MQSeries product. For instructions, see the *MQSeries for Compaq (DIGITAL) OpenVMS System Management Guide*.

Alternatively, you can use the following method:

1. Copy the following files from the Compaq OpenVMS Alpha server to the Compaq OpenVMS Alpha client system, into the same directories as on the server system:

```
SYS$COMMON:[SYSEXEC]DSPMQTRC.EXE
SYS$COMMON:[SYSEXEC]ENDMQTRC.EXE
SYS$COMMON:[SYSEXEC]RUNMQTRC.EXE
SYS$COMMON:[SYSEXEC]STRMQTRC.EXE
SYS$COMMON:[SYSLIB]AMQCC62A.EXE
SYS$COMMON:[SYSLIB]AMQCCDCA.EXE
SYS$COMMON:[SYSLIB]AMQCCTCA.EXE
SYS$COMMON:[SYSLIB]AMQTRC.FMT
SYS$COMMON:[SYSLIB]MQIC.EXE
SYS$COMMON:[SYSLIB]MQICB.EXE
SYS$COMMON:[SYSLIB]MQMCS.EXE
```

2. Create the following directory on the Compaq OpenVMS Alpha client system:

Installing from Compaq OpenVMS Alpha

```
SYS$COMMON:[SYSHLP.EXAMPLES.MQSERIES]
```

Copy all the files from the same directory on the Compaq OpenVMS Alpha server into this directory

Installing an MQSeries client on OS/2 Warp from MQSeries for Compaq OpenVMS Alpha

1. Create the following directory structure on the OS/2 Warp system:

```
<drive>:\mqm\bin  
<drive>:\mqm\dll  
<drive>:\mqm\inc  
<drive>:\mqm\lib  
<drive>:\mqm\msg
```

2. Copy the following files into the above directories on the OS/2 Warp system from Compaq OpenVMS Alpha:

```
<drive>:\mqm\bin
```

```
SYS$COMMON:[SYSHLP.EXAMPLES.MQSERIES.OS2_CLIENT.BIN]  
SYS$COMMON:[SYSHLP.EXAMPLES.MQSERIES.OS2_CLIENT.SAMP.BIN]
```

```
<drive>:\mqm\dll
```

```
SYS$COMMON:[SYSHLP.EXAMPLES.MQSERIES.OS2_CLIENT.DLL]
```

```
<drive>:\mqm\inc
```

```
SYS$COMMON:[SYSHLP.EXAMPLES.MQSERIES.OS2_CLIENT.INC]
```

```
<drive>:\mqm\lib
```

```
SYS$COMMON:[SYSHLP.EXAMPLES.MQSERIES.OS2_CLIENT.LIB]
```

```
<drive>:\mqm\msg
```

```
SYS$COMMON:[SYSHLP.EXAMPLES.MQSERIES.OS2_CLIENT.MSG]
```

See Chapter 6, “Verifying the installation” on page 113 for how to use the sample executables `amqsputc.exe` and `amqsgetc.exe`. Alternatively, you can build these yourself from the corresponding source files `amqsput0.c` and `amqsget0.c`.

3. Create an MQSeries configuration file (`mqs.ini`) in the `<drive>:\mqm` directory and add the following stanza:

```
AllQueueManagers:  
  DefaultPrefix=<drive>:\mqm
```

4. Make the following changes to your `config.sys` file on the OS/2 Warp system:

```
PATH:      include <drive>:\mqm\bin  
DPATH:     include <drive>:\mqm\bin;c:\mqm\msg  
LIBPATH:   include <drive>:\mqm\dll  
HELP:      include <drive>:\mqm\bin  
LIB:       include <drive>:\mqm\lib
```

For information about how to make the changes, see “Changing the OS/2 Warp `config.sys` file” on page 88.

Reboot your system for these changes to take effect.

5. Now go to Chapter 6, “Verifying the installation” on page 113.

Installing from Compaq OpenVMS Alpha

Installing an MQSeries client on DOS from MQSeries for Compaq OpenVMS Alpha

1. Create a suitable directory on the DOS system.
2. Copy the following files into the directory you have created on the DOS system from MQSeries for Compaq OpenVMS Alpha
SYS\$COMMON:[SYSHLP.EXAMPLES.MQSERIES.DOS_CLIENT]:

*.exe
*.msg
*.lib
*.h

See Chapter 6, "Verifying the installation" on page 113 for information about how to use the sample executables amqsputc.exe and amqsgetc.exe (included when you copy *.exe). Alternatively, you can build these yourself from the corresponding source files (amqsput0.c and amqsget0.c) in <drive>:\mqm\tools\c\samples.
3. Make changes to the PATH statement and the DOS APPEND statement in your autoexec.bat file to include the directory you have used (for information about how to make the changes, see "Changing the autoexec.bat file for DOS and Windows 3.1" on page 89).
4. Now go to Chapter 6, "Verifying the installation" on page 113.

Installing an MQSeries client on Windows 3.1 from MQSeries for Compaq OpenVMS Alpha

1. Create a suitable directory on the Windows 3.1 system.
2. Copy the following files, into the directory you have created on the Windows 3.1 system from MQSeries for Compaq OpenVMS Alpha
SYS\$COMMON:[SYSHLP.EXAMPLES.MQSERIES.WIN_CLIENT.BIN]:

*.exe
*.dll
*.msg
*.lib
*.h

See Chapter 6, "Verifying the installation" on page 113 for information about how to use the sample executables amqsputw.exe and amqsgetw.exe (included when you copy *.exe). Alternatively, you can build these yourself from the corresponding source files (amqsputw.c and amqsgetw.c) in <drive>:\mqm\win.
3. Make changes to the PATH statement and the DOS APPEND statement in your autoexec.bat file to include the directory you have used (for details of how to make the changes, see "Changing the autoexec.bat file for DOS and Windows 3.1" on page 89).
4. Now go to Chapter 6, "Verifying the installation" on page 113.

Note: You can also install the Windows 3.1 client code on WIN-OS/2 under OS/2 Warp, Windows NT, Windows 95, and Windows 98.

Installing MQSeries clients from MQSeries on UNIX systems

When you have included the MQSeries client files in the installation of MQSeries on your UNIX server machine, the files are located in these directories:

UNIX system files

```
/opt/mqm/xxx_client/bin
/opt/mqm/xxx_client/inc
/opt/mqm/xxx_client/lib
/opt/mqm/xxx_client/msg
/opt/mqm/xxx_client/samp/bin
```

Where xxx identifies the UNIX platform of your server machine.

OS/2 Warp files

```
/opt/mqm/os2_client/bin
/opt/mqm/os2_client/dll
/opt/mqm/os2_client/inc
/opt/mqm/os2_client/lib
/opt/mqm/os2_client/msg
/opt/mqm/os2_client/samp/bin
```

DOS files

```
/opt/mqm/dos_client/bin
/opt/mqm/dos_client/lib
/opt/mqm/dos_client/inc
/opt/mqm/dos_client/msg
/opt/mqm/dos_client/samp/bin
```

Windows 3.1 files

```
/opt/mqm/win_client/bin
/opt/mqm/win_client/dll
/opt/mqm/win_client/inc
/opt/mqm/win_client/lib
/opt/mqm/win_client/msg
/opt/mqm/win_client/samp/bin
```

Installing an MQSeries client on a UNIX system from a UNIX system

If possible, do this by installing the MQSeries client directly from the media supplied with your MQSeries product. For instructions, see the *System Management Guide* for your platform.

Alternatively, you can use the following method:

1. Create the following directory structure on the UNIX system:

```
/opt/mqm/bin
/opt/mqm/inc
/opt/mqm/lib
/opt/mqm/msg
```

2. Copy the following files into the above directories on the client UNIX system from the server UNIX system:

```
/opt/mqm/bin
/opt/mqm/xxx_client/bin
/opt/mqm/xxx_client/samp/bin/amqsputc.exe
/opt/mqm/xxx_client/samp/bin/amqsgetc.exe
```

```
/opt/mqm/inc
/opt/mqm/xxx_client/inc
```

```
/opt/mqm/lib
```

Installing from UNIX systems

```
/opt/mqm/xxx_client/lib  
  
/opt/mqm/msg  
/opt/mqm/xxx_client/msg
```

See Chapter 6, “Verifying the installation” on page 113 for how to use the sample executables `amqsputc.exe` and `amqsgetc.exe`. Alternatively, you can build these yourself from the corresponding source files `amqsput0.c` and `amqsget0.c`.

For national-language specific directories, see the README file supplied with MQSeries on the server platform.

3. Copy the file `amq.cat` from `/opt/mqm/msg` into the default message catalog directory for your system, or a directory referenced in your `NLSPATH` environment variable.

For example, on AT&T GIS UNIX (NCR UNIX):

```
/usr/lib/locale/C/LC_MESSAGES
```

4. Make a directory `/var/mqm`. Create an MQSeries configuration file (`mqs.ini`) in the `/var/mqm` directory and add the following stanza:

```
AllQueueManagers:  
DefaultPrefix=/var/mqm
```

5. This step is optional:
 - a. Symbolically link the C header files from `opt/mqm/inc` into `/usr/include`
 - b. Link the libraries from `opt/mqm/lib` into `/usr/lib`
 - c. Link the programs from `opt/mqm/bin` into `/usr/bin`
6. Now go to Chapter 6, “Verifying the installation” on page 113.

Installing an MQSeries client on OS/2 Warp from a UNIX system

1. Create the following directory structure on the OS/2 Warp system:

```
<drive>:\mqm\bin  
<drive>:\mqm\dll  
<drive>:\mqm\inc  
<drive>:\mqm\lib  
<drive>:\mqm\msg  
<drive>:\mqm\samp\bin
```
2. Copy the following files into the above directories on the OS/2 Warp system from MQSeries on the UNIX system:

```
<drive>:\mqm\bin  
/opt/mqm/os2_client/bin  
  
<drive>:\mqm\dll  
/opt/mqm/os2_client/dll  
  
<drive>:\mqm\inc  
/opt/mqm/os2_client/inc  
  
<drive>:\mqm\lib  
/opt/mqm/os2_client/lib  
  
<drive>:\mqm\msg  
/opt/mqm/os2_client/msg  
  
<drive>:\mqm\samp\bin  
/opt/mqm/os2_client/samp/bin/amqsputc.exe  
/opt/mqm/os2_client/samp/bin/amqsgetc.exe
```

Installing from UNIX systems

See Chapter 6, “Verifying the installation” on page 113 for how to use the sample executables `amqsputc.exe` and `amqsgetc.exe`. Alternatively, you can build these yourself from the corresponding source files `amqsput0.c` and `amqsget0.c`.

Note: For national language-specific directories, see the README file supplied with MQSeries on the server platform.

3. Create an MQSeries configuration file (`mqs.ini`) in the `<drive>:\mqm` directory and add the following stanza:

```
AllQueueManagers:
  DefaultPrefix=<drive>:\mqm
```

4. Make the following changes to your `config.sys` file on the OS/2 Warp system (for information about how to make the changes, see “Changing the OS/2 Warp `config.sys` file” on page 88):

```
PATH:      include <drive>:\mqm\bin
DPATH:     include <drive>:\mqm\bin;c:\mqm\msg
LIBPATH:   include <drive>:\mqm\dll
HELP:      include <drive>:\mqm\bin
LIB:       include <drive>:\mqm\lib
INCLUDE:   include <drive>:\mqm\inc
```

5. Now go to Chapter 6, “Verifying the installation” on page 113.

Installing an MQSeries client on DOS from a UNIX system

1. Copy the following files into a suitable directory on the DOS system from MQSeries on the UNIX system:

```
/opt/mqm/dos_client/bin
*.exe
```

```
/opt/mqm/dos_client/inc
*.h
```

```
/opt/mqm/dos_client/lib
*.lib
```

```
/opt/mqm/dos_client/msg
*.msg
```

```
/opt/mqm/dos_client/samp/bin/amqsputc.exe
/opt/mqm/dos_client/samp/bin/amqsgetc.exe
```

See Chapter 6, “Verifying the installation” on page 113 for information about how to use the sample executables `amqsputc.exe` and `amqsgetc.exe`. Alternatively, you can build these yourself from the corresponding source files.

For national-language specific directories, see the README file supplied with MQSeries on the server platform.

2. Make changes to the `PATH` statement and the `DOS APPEND` statement in your `autoexec.bat` file to include the directory you have used (for information about how to make the changes, see “Changing the `autoexec.bat` file for DOS and Windows 3.1” on page 89).
3. Now go to Chapter 6, “Verifying the installation” on page 113.

Installing an MQSeries client on Windows 3.1 from a UNIX system

1. Copy the following files into a suitable directory on the Windows system from MQSeries on the UNIX system (for example, `c:\mqm`):

Installing from UNIX systems

```
/opt/mqm/win_client/bin
*.exe

/opt/mqm/win_client/dll
*.dll

/opt/mqm/win_client/lib
*.lib

/opt/mqm/win_client/msg
*.msg

/opt/mqm/win_client/inc/
*.h
*.hpp (Version 5.1 products)

/opt/mqm/win_client/samp/bin/amqspwtw.exe
/opt/mqm/win_client/samp/bin/amqsgetw.exe
/opt/mqm/win_client/samp/bin/imq*c.exe (Version 5.1 products)
```

See Chapter 6, “Verifying the installation” on page 113 for how to use the sample executables amqspwtw.exe and amqsgetw.exe. Alternatively, you can build these yourself from the corresponding source files.

For national-language specific directories, see the README file supplied with MQSeries on the server platform.

2. Make changes to the PATH statement and the DOS APPEND statement in your autoexec.bat file to include the directory you have used (for information about how to make the changes, see “Changing the autoexec.bat file for DOS and Windows 3.1” on page 89).
3. Now go to Chapter 6, “Verifying the installation” on page 113.

Note: You can also install the Windows 3.1 client code on WIN-OS/2 under OS/2 Warp, Windows 95, and Windows.

MQSeries client for VM/ESA

The MQSeries client for VM/ESA is supplied as part of the VM/ESA product. The client is installed as a component of CMS during the installation of CMS.

The client code resides on MAINT 193 minidisk and can be accessed by linking to MAINT 193.

To set up your client and server installation and to check that the communication link is working, see Chapter 6, “Verifying the installation” on page 113.

Changing the config.sys and autoexec.bat files

On OS/2 Warp, Windows 3.1, and DOS systems, you need to add some statements to the files that the operating system uses when it starts up. The files are config.sys and autoexec.bat. These statements define to the operating system the path or paths to the MQSeries client directories.

The following sections explain how to do this.

Changing the OS/2 Warp config.sys file

Edit the config.sys file as follows:

1. Find the line that starts SET PATH=

2. Add to the end of the line, after a semicolon (;) <drive>:\mqm\bin
3. Find the line that starts SET DPATH=
4. Add to the end of the line, after a semicolon (;)
<drive>:\mqm\bin; <drive>:\mqm\msg
5. Find the line that starts SET HELP=
6. Add to the end of the line, after a semicolon (;) <drive>:\mqm\bin
7. Find the line that starts SET LIB=
8. Add to the end of the line, after a semicolon (;) <drive>:\mqm\lib
9. Find the line that starts LIBPATH= if there is one,
10. Add to the end of the line, after a semicolon (;) <drive>:\mqm\dll
11. If there is no LIBPATH= line, create a new line: LIBPATH=<drive>:\mqm\dll

You must restart your system before these changes take effect.

Changing the autoexec.bat file for DOS and Windows 3.1

Edit the autoexec.bat file as follows:

1. Find the line that starts PATH
2. Add to the end of the line, after a semicolon (;) <drive>:\<dir> using the drive and directory that you created on your system for the MQSeries client files.
3. If there is no PATH line, create a new line: PATH <drive>:\<dir>
4. Find the line that starts APPEND
5. Add to the end of the line, after a semicolon (;) <drive>:\<dir> using the drive and directory that you created on your system for the MQSeries client files.
6. If there is no APPEND line, create a new line: APPEND <drive>:\<dir>

You must restart your system before these changes take effect.

Installing WebSphere MQ clients

Chapter 5. Configuring communication links

This chapter tells you how to configure the WebSphere MQ client and server communication links, and how to enable the server to listen for communications from the WebSphere MQ client.

In WebSphere MQ, the logical communication links between objects are called *channels*. The channels used to connect WebSphere MQ clients to servers are called MQI channels. You set up channel definitions at each end of your link so that your WebSphere MQ application on the WebSphere MQ client can communicate with the queue manager on the server. There is a detailed description of how to do this in Chapter 8, “Using channels” on page 127.

Before you define your MQI channels, you need to:

1. Decide on the form of communication you are going to use. See “Deciding which communication type to use”
2. Define the connection at each end of the channel:

To define the connection, you need to:

- Configure the connection.
- Record the values of the parameters that you need for the channel definitions.
- Enable the server to detect incoming network requests from your WebSphere MQ client. This involves starting a *listener*.

This chapter explains how to perform these tasks.

Deciding which communication type to use

There are five types of communication for MQI channels on different platforms:

- DECnet
- LU 6.2
- NetBIOS
- SPX
- TCP/IP

When you define your MQI channels, each channel definition must specify a transmission protocol (transport type) attribute. A server is not restricted to one protocol, so different channel definitions can specify different protocols. For WebSphere MQ clients, it might be useful to have alternate MQI channels using different transmission protocols.

Your choice of transmission protocol also depends on your particular combination of WebSphere MQ client and server platforms. The possible combinations are shown in the following table.

Communication links

Table 6. Transmission protocols - combination of WebSphere MQ client and server platforms

Transmission protocol	WebSphere MQ client	WebSphere MQ server
TCP/IP	Compaq OpenVMS Alpha DOS OS/2 Warp OS/400 ¹ UNIX systems VM/ESA Windows Windows 3.1 Windows 95	Compaq NonStop Kernel Compaq OpenVMS Alpha OS/2 Warp OS/400 UNIX systems VSE/ESA Windows z/OS
LU 6.2	Compaq OpenVMS Alpha OS/2 Warp UNIX systems ² VM/ESA Windows ³	Compaq NonStop Kernel OS/2 Warp OS/400 UNIX systems ² Windows z/OS
NetBIOS	DOS OS/2 Warp Windows Windows 3.1 Windows 95	OS/2 Warp Windows
SPX	DOS OS/2 Warp Windows Windows 3.1 Windows 95	OS/2 Warp Windows
DECnet	Compaq OpenVMS Alpha	Compaq OpenVMS Alpha
Notes: 1. For Java applications only 2. Except Linux for zSeries and Sun Solaris, Intel Platform Edition 3. Except Windows 98		

Defining a connection is described in the following sections:

- “Defining a TCP/IP connection” on page 93
- “Defining an LU 6.2 connection” on page 99
- “Defining a NetBIOS connection” on page 106
- “Defining an SPX connection” on page 108
- “Defining a DECnet connection” on page 111

Defining a TCP/IP connection

The steps to take are detailed in the sections that follow:

On the WebSphere MQ client

Initialize TCP/IP.

On the server

There are three things to do:

1. Decide on a port number.
The port to connect to defaults to 1414. Port number 1414 is assigned by the Internet Assigned Numbers Authority to WebSphere MQ.
2. Initialize TCP/IP, and record the network address, in the form 1.11.1.11, of the server machine.
3. Configure files, or run a command, to specify the port number and to run a listener program (not z/OS). On z/OS, start a channel initiator and a listener to listen on that port.

For more detailed step-by-step examples, see the *WebSphere MQ Intercommunication* manual.

TCP/IP connection limits

On any platform, there is a limit to the number of outstanding connection requests that can be queued at a single TCP/IP port. This is not the same as the maximum number of clients you can attach to a WebSphere MQ server. You can connect more clients to a server, up to the level determined by the server system resources. The backlog values for connection requests are shown in the following table:

Table 7. Maximum outstanding connection requests queued at a TCP/IP port

Server platform	Maximum connection requests
AIX	100
AT&T GIS UNIX	5
Compaq NonStop Kernel	5
Compaq OpenVMS Alpha	5
Compaq Tru64 UNIX	10
HP-UX	20
Linux	100
OS/2 Warp	10
OS/400	255
Solaris	100
SINIX and DC/OSx	5
VSE/ESA	5
Windows Server	100
Windows Workstation	100
z/OS	255

If the connection limit is reached, the client receives a return code of MQRC_Q_MGR_NOT_AVAILABLE from the **MQCONN** call, and an AMQ9202 error in the client error log (mqm\errors\amqerr0n.log). See the *WebSphere MQ*

TCP/IP connection

Messages manual for an explanation of the error message. If the client retries the MQCONN request, it might be successful.

To increase the number of connection requests you can make, and avoid error messages being generated by this limitation, you can have a listener listening on more than one port, or have more than one queue manager.

Defining a TCP/IP connection on a WebSphere MQ client (any platform)

Initialize TCP/IP.

The channel definitions that you create later will include the network address and port number of the server to which the WebSphere MQ client is sending.

You can use the KeepAlive option to make TCP/IP periodically check that the other end of the connection is still open. If it is not, the channel is closed. To use this option, create a queue manager configuration file (QM.INI) and add the following entry to it:

```
TCP:
  KeepAlive=yes
```

Store the QM.INI file in the appropriate directory for your platform:

- On OS/2 Warp, Windows 95:
 <drive>:\<dir>\mqm
- On DOS, Windows 3.1, and WIN-OS/2:
 The root directory of the drive where the client is installed
- On UNIX systems:
 /var/mqm

Note: On Windows this information is in the Registry. For more information, access the WebSphere MQ Information Center.

Defining a TCP/IP connection on an OS/2 Warp server

To define the connection, you must first Initialize TCP/IP, and record the network address of the server machine. The address is displayed in a box as TCP/IP initializes.

You can use inetd or the Run Listener (RUNMQLSR) command to define the connection.

Using inetd

First, type SET ETC to return the path to the ETC subdirectory.

To use inetd to start MQI channels, configure files as follows. In the example given, the path is taken to be TCPIP. The path is case-sensitive and the entries in the two files must match.

- To TCPIP\ETC\SERVICES (or MPTN\ETC\SERVICES) add the line:
MQSeries 1414/tcp

where 1414 is the default value of the port number required. If you want to use another port number, replace 1414 the port number you have chosen.

- To TCPIP\ETC\INETD.LST, add the line:
MQSeries tcp C:\MQM\BIN\AMQCRSTA [-m QMName]

The square brackets indicate that QMName is an optional parameter; this is not required if you are using the default queue manager. If MQSeries for OS/2 Warp is not installed on the C drive, replace the C: above with the correct drive letter for your OS/2 Warp installation.

It is possible to have more than one queue manager on the server. Add a line to each of the two files for each queue manager. For example:

```
MQSeries1    1414/tcp
MQSeries2    1415/tcp

MQSeries1    tcp C:\MQM\BIN\AMQCRSTA -m QM1
MQSeries2    tcp C:\MQM\BIN\AMQCRSTA -m QM2
```

You must stop and then restart the inetd program before continuing.

Using the Run Listener (RUNMQLSR) command

You can use the Run Listener command supplied with MQSeries for OS/2 Warp to start MQI channels. Using this method starts new MQI channels as threads. For example:

```
RUNMQLSR -t tcp [-m QMNAME] [-p 1822]
```

The square brackets indicate optional parameters:

- m QMNAME is not required if you are using the default queue manager.
- p specifies the port number. This parameter is not required if you are using port number 1414, the default value.

It is possible to have more than one queue manager running on the server. Start a listener program for each one, using a different port number for each queue manager. For example:

```
RUNMQLSR -t tcp
RUNMQLSR -t tcp -m QM2 -p 1415
```

Defining a TCP/IP connection on a Windows server

TCP/IP is initialized automatically as a service when Windows starts. Before starting a listener, you must define the port number as follows.

To the file c:\winnt\system32\drivers\etc\services, add the line:

```
MQSeries    1414/tcp
```

where 1414, the default, is the port number required. (On some versions the path is c:\winnt35\system32\drivers\etc\services)

To use another port replace the value 1414 with the port number you want to use.

It is possible to have more than one queue manager on the server. Add a line, as above, for each queue manager. For example:

```
MQSeries1    1414/tcp
MQSeries2    1415/tcp
```

Using the Run Listener (RUNMQLSR) command

You can use the Run Listener command to define a connection. The Run Listener command starts new MQI channels as threads. To run the listener, you use the RUNMQLSR command. For example:

TCP/IP connection

```
RUNMQLSR -t tcp [-m QMNAME] [-p 1822]
```

The square brackets indicate optional parameters:

-m QMNAME is not required for the default queue manager.

-p 1822 is not required if the default port number 1414 is used.

For

It is possible to have more than one queue manager running on the server machine. Start a listener program for each one, on different ports. For example:

```
RUNMQLSR -t tcp  
RUNMQLSR -t tcp -m QM2 -p 1415
```

Defining a TCP/IP connection on a UNIX system server

You can use inetd or the Run Listener (RUNMQLSR) command to define a TCP/IP connection on a UNIX system server. If you use inetd, a process is started for each connection you define. If you use the RUNMQLSR command, a thread is started for each connection. This method can therefore be more efficient.

Note: The name of the installation directory on your UNIX system is represented here by **mqmtop**.

Configure the inetd daemon on the server, so that inetd will start the MQI channels. Log on as root and configure the following files:

1. In the /etc/services file, add the line:

```
MQSeries      1414/tcp
```

where 1414, the default, is the port number required.

To use another port replace the value 1414 with the port number you want to use.

2. Add a line in the inetd.conf file to call the program amqcrsta:

Note: This file is case sensitive

```
MQSeries stream tcp nowait mqm /mqmtop/bin/amqcrsta amqcrsta [-m QM1]
```

where QM1 is the queue manager name. This parameter is not required if you are using the default queue manager.

3. The updates are active after you issue the following commands from the root user ID:

On AIX:

```
refresh -s inetd
```

On other UNIX systems:

```
kill -1 <process number of inetd daemon>
```

It is possible to have more than one queue manager on the server machine. Add a line to each of the two files, as above, for each queue manager. For example, in /etc/services, add the following lines:

```
MQSeries1     1414/tcp  
MQSeries2     1415/tcp
```

and in the inetd.conf file, add the following lines:

```
MQSeries1 stream tcp nowait mqm /mqmtop/bin/amqcrsta amqcrsta -m QM1
MQSeries2 stream tcp nowait mqm /mqmtop/bin/amqcrsta amqcrsta -m QM2
```

Using the Run Listener (RUNMQLSR) command

To run the listener supplied with WebSphere MQ for AIX, HP-UX, Linux, or Solaris, use the RUNMQLSR command. For example:

```
RUNMQLSR -t tcp [-m QMNAME] [-p 1822]
```

The square brackets indicate optional parameters:

- m QMNAME is not required if you are using the default queue manager.
- p specifies the port number. This parameter is not required if you are using port number 1414, the default value.

It is possible to have more than one queue manager running on the server machine. Start a listener program for each one, on different ports. For example:

```
RUNMQLSR -t tcp
RUNMQLSR -t tcp -m QM2 -p 1415
```

Defining a TCP/IP connection on an OS/400 server

TCP/IP is normally initialized automatically as a service during OS/400 startup.

Use the Start Listener (STRMQMLSR) command to enable the WebSphere MQ server to receive incoming client connections.

The default port number for the WebSphere MQ for iSeries TCP/IP listener program is 1414.

Defining a TCP/IP connection on a z/OS server

Set up communications for WebSphere MQ for z/OS to use TCP/IP channels, as described in the *WebSphere MQ Intercommunication* manual. Then:

1. Start the channel initiator by typing the following command:
START CHINIT
2. To enable the WebSphere MQ server to receive incoming client connections, start the listener by typing the following command:
START LSTR TRPTYPE(TCP) PORT(port-number)

Defining a TCP/IP connection on a Compaq OpenVMS Alpha server

Set up communications for MQSeries for Compaq OpenVMS Alpha to use TCP/IP channels, as described in the *MQSeries for Compaq (DIGITAL) OpenVMS System Management Guide*.

See the *WebSphere MQ Intercommunication* manual for information about how to configure TCP/IP services on Compaq OpenVMS Alpha.

Note: There is a limit to the number of outstanding connection requests that can be queued at a single TCP/IP port. This is given in "TCP/IP connection limits" on page 93.

TCP/IP connection

Defining a TCP/IP connection on a Compaq NonStop Kernel server

Set up communications for MQSeries for Compaq NonStop Kernel to use TCP/IP channels, as described in the *WebSphere MQ Intercommunication* manual. Then start the listener to enable the WebSphere MQ server to receive incoming client connections. There are a number of ways of doing this. For example:

- `runmqlsr -t tcp`
- Start server `mqs-tcplis00` from pathway

By default, the MQSeries for Compaq NonStop Kernel listener program uses port 1414 and TCP/IP process `$ztc0`.

See the *MQSeries for Compaq NonStop Kernel System Administration* for information about how to configure TCP/IP services on Compaq NonStop Kernel.

Defining a TCP/IP connection on a VSE/ESA server

Define a TCP/IP connection as follows:

1. Define the listener port number and number of clients to be supported to the VSE/ESA queue manager. You can do this through the MQSeries for VSE/ESA Global System Definition.
2. Ensure that the code page you use is one that is supported for clients. See the *WebSphere MQ Intercommunication* manual for more information.
3. Restart the MQSeries for VSE/ESA queue manager with the MQSE and MQIT transactions.

When the MQSeries for VSE/ESA queue manager is started, the TCP/IP listener starts automatically.

Defining an LU 6.2 connection

The steps to take are detailed in the sections that follow:

On the WebSphere MQ client

1. Configure SNA.
2. Set TpName (see “Setting the Transaction Program name”) and TpPath.
3. Establish a valid SNA session between the WebSphere MQ client and server machines.

On the server

- If you are using z/OS, start a channel initiator and a listener
- On any other platform, start a listener, or create a listening attachment

Defining an LU 6.2 Connection on an OS/2 Warp client

The following steps will enable you to configure an LU 6.2 connection on an MQSeries for OS/2 Warp client.

Configuring SNA

First configure SNA so that an LU 6.2 conversation can be established between the WebSphere MQ client machine and the server machine. See the *Multiplatform APPC Configuration Guide* for information. (This is supplied online with some WebSphere MQ products as the *APPC Configuration Guide (Red Book)*.)

Setting the Transaction Program name

Set the Transaction Program name (TpName or TPNAME) as shown in the following table. This table applies to all platforms.:

Table 8. Settings on the WebSphere MQ client system for a server platform

Server platform	TPNAME
OS/2 Warp	As specified in the OS/2 Warp Run Listener command on the server, or defaulted from the OS/2 Warp queue manager configuration file on the server.
Windows	As specified in the Windows Run Listener command, or the invokable Transaction Program that was defined using TpSetup on Windows.
UNIX systems	The same as the corresponding TpName in the side information on the remote queue manager on the server. Note: UNIX is case-sensitive
OS/400	The same as the compare value in the routing entry on the OS/400 system.
Compaq NonStop Kernel	As specified in the channel definition on the server.
z/OS	The same as the corresponding TpName in the side information on the remote queue manager on the server.

For more detailed step-by-step examples, see the *WebSphere MQ Intercommunication* manual.

Establishing a session

Establish a valid session between the two machines. You can specify the local LU that MQSeries for OS/2 Warp will use, either by creating the WebSphere MQ client configuration file (QM.INI), or as an environment variable. An entry in the configuration file takes precedence over the environment variable. For a client on MQSeries for OS/2 Warp the QM.INI file is located in directory C:\MQM.

Defining an LU 6.2 connection

In the QM.INI file, under the LU 6.2 section add the line:

```
LocalLU = Your_LU_Name
```

or specify the environment variable:

```
SET APPNLLU=Your_LU_Name
```

If nothing has been specified, your default LU will be used.

Find out the name of the partner LU alias, as defined in the WebSphere MQ client machine's Communications Manager/2 profile. You will need this later, when you define the MQI channels - it is the Connection name (CONNNAME).

SECURITY PROGRAM is always used when MQSeries for OS/2 Warp attempts to establish an SNA session if a password and user ID are specified. Otherwise SECURITY NONE is used.

Defining an LU 6.2 Connection on an OS/2 Warp server

Start the listener program with the RUNMQLSR command, giving the TpName to listen on, or use Attach Manager in Communications Manager/2.

Using the RUNMQLSR command

Here is an example of a command to start the listener:

```
RUNMQLSR -t LU62 -n RECV [-m QMNAME]
```

where RECV is the TpName that is specified in the client channel definition at the WebSphere MQ client end (as the "TpName to start on the remote, or server, side"). The last part in square brackets is optional and is not required for the default queue manager.

It is possible to have more than one queue manager running on the server machine. Assign a different TpName to each queue manager, and then start a listener program for each one. For example:

```
RUNMQLSR -t LU62 -m QM1 -n TpName1  
RUNMQLSR -t LU62 -m QM2 -n TpName2
```

Using Communications Manager/2

You can use Attach Manager in Communications Manager/2 to start the listener program. You must supply a Transaction program (TP) definition specifying:

- TP name
- Program path and file name
- Program parameter string

You can do this using the panel configuration in Communications Manager/2, or you can edit your NDF file directly (see the information about defining transaction programs in the online *APPC Configuration Guide (Red Book)*).

Panel configuration: These are the entries required on the TP definition panel:

```
Transaction Program (TP) name      : AMQCRS6A  
OS/2 Warp program path and file name : c:/mqm/bin/amqcrs6a.exe  
Program parameter string           : -n AMQCRS6A
```

This example is for the default queue manager, for other queue managers include -m QMNAME in the Program parameter string.

Defining an LU 6.2 connection

NDF file configuration: Your node definitions file (.ndf) must contain a `define_tp` command. The following example shows what to include:

```
define_tp
  tp_name(AMQCRS6A)
  filespec(c:/mqm/bin/amqcrs6a.exe)
  parm_string(-n AMQCRS6A)
```

This example is for the default queue manager, for other queue managers include `-m QMNAME` in the `parm_string`.

Defining an LU 6.2 Connection on a Windows WebSphere MQ client

First configure SNA to allow an LU 6.2 conversation to be established between the WebSphere MQ client machine and the server machine.

Set the Transaction Program name (TpName or TPNAME) as shown in Table 8 on page 99:

Create a CPI-C Side Object (symbolic destination) and record this name to use later in your channel definitions as the Connection name (CONNAME).

In the CPI-C Side Object enter the Partner LU Name at the receiving machine, the TP Name and the Mode name. For example:

Partner LU Name	OS2R0G2
Partner TP Name	recv
Mode Name	#INTER

SECURITY PROGRAM is used, where supported by CPI-C, when WebSphere MQ attempts to establish an SNA session.

Defining an LU 6.2 Connection on a Windows server

You can use either of these methods:

- Start the listener program with the `RUNMQLSR` command, giving the TpName to listen on. This starts a thread to process each inbound client connection.
- Use one of the SNA servers (either 2.11 or 3) listed to set up an invokable transaction program (TP). Then invoke `amqcrs6a` as a separate process for each client connection. The TpName should match that specified in the CPI-C side object information referenced by CONNAME in the client-connection channel definition (see Chapter 8, “Using channels” on page 127).

Using the RUNMQLSR command

Example of the command to start the listener:

```
RUNMQLSR -t LU62 -n RECV [-m QMNAME]
```

where RECV is the TpName that is specified at the WebSphere MQ client end as the “TpName to start on the remote side (server)”. The last part in square brackets is optional and is not required for the default queue manager.

It is possible to have more than one queue manager running on the server machine. Assign a different TpName to each queue manager, and then start a listener program for each one. For example:

Defining an LU 6.2 connection

```
RUNMQLSR -t LU62 -m QM1 -n TpName1
RUNMQLSR -t LU62 -m QM2 -n TpName2
```

Using an SNA server

You can use TpSetup (from the SNA Server SDK) to define amqcrs6a as an invokable TP, or set various Registry values manually.

Defining an LU 6.2 Connection on a UNIX WebSphere MQ client

1. Configure SNA so that an LU 6.2 conversation can be established between the WebSphere MQ client machine and the server machine. See the online book *APPC Configuration Guide (Red Book)* for information or, for Solaris, see the *SunLink P2P LU 6.2 Programmer's Guide*.
2. Set the Transaction Program name (TpName or TPNAME) as shown in the Table 8 on page 99:
3. Create a CPI-C Side Object (symbolic destination) and record this name to use later in your channel definitions as the Connection name (CONNAME).
- 4.

On DC/OSx

Create an entry in the /etc/opt/lu62/cpic_cfg file:

```
sendMP01 <local LU name> <remote LU name> <mode name> <remote TP name>
```

On SINIX

Create a XSYMDEST entry in the TRANSIT KOGS file:

```
XSYMDEST sendMP01,
           RLU      = forties,
           MODE     = MODE1,
           TP       = recvMP01,
           TP-TYP   = USER,
           SEC-TYP  = NONE
```

On Solaris

Set the environment variable APPC_LOCAL_LU to refer to the name of your Local LU.

On other UNIX systems

Enter the Partner LU Name at the receiving machine, the TpName and the Mode Name in the CPI-C Side Object. For example:

```
Partner LU Name      OS2R0G2
Remote TP Name       recv
Service Transaction Program no
Mode Name            #INTER
```

SECURITY PROGRAM is used, where supported by CPI-C, when WebSphere MQ attempts to establish an SNA session.

Defining an LU 6.2 Connection on a UNIX server

1. On the server, create a TPN profile. In the TPN profile, enter the full path to the executable and the Transaction program name. For example:

```
Full path to TPN executable  /mqmtop/bin/amqcrs6a
Transaction Program name     recv
User ID                      0
```

2. On SINIX, create an XTP entry in the TRANSIT KOGS file:

```
XTP  recvMP01,
      UID      = guenther,
      TYP      = USER,
      PATH     = /home/guenther/recvMP01.sh,
      SECURE   = NO
```

The file /home/guenther/recvMP01.sh contains:

```
#!/bin/ksh
#
# script to start the receiving side for the qmgr MP01
#
exec /opt/mqm/bin/amqcrs6a -m <queue manager>
```

You cannot use LU 6.2 for a WebSphere MQ server running on DC/OSx because the DC/OSx SNA implementation does not support the ACCEPT verb (use TCP/IP instead).

The User ID field can specify a user who is a member of the mqm group.

You might need to use a queue manager other than the default queue manager. If so, define a command file that includes:

```
amqcrs6a -m Queue_Man_Name
```

and call the command file.

Defining an LU 6.2 Connection on an OS/400 server

Use ADDRTGE command to add a routing entry to a subsystem at the initiated end to enable the initiating end to start the channel. The routing entry specifies the program that is invoked when the channel starts.

Alternatively, create and start a new subsystem by using the Work with Subsystem Descriptions (WRKSBSD) panel. The ADDRTGE panel is shown in Figure 3.

Add Routing Entry (ADDRTGE)

Type choices, press Enter.

Subsystem description	QSNADS	Name
Library	*LIBL	Name, *LIBL, *CURLIB
Routing entry sequence number . .	1	1-9999
Comparison data:		
Compare value	MQSERIES	
Starting position	37	1-80
Program to call	AMQCRC6A	Name, *RTGDTA
Library	QMOM	Name, *LIBL, *CURLIB
Class	*SBSD	Name, *SBSD
Library	*LIBL	Name, *LIBL, *CURLIB
Maximum active routing steps . .	*NOMAX	0-1000, *NOMAX
Storage pool identifier	1	1-10

Bottom

F3=Exit F4=Prompt F5=Refresh F12=Cancel F13=How to use this display
F24=More keys

Figure 3. LU 6.2 communication setup panel - initiated end

Defining an LU 6.2 connection

Subsystem description

The name of your subsystem where this definition resides. Use the OS/400 WRKSBSD command to view and update the appropriate subsystem description for the routing entry.

Routing entry sequence number

A unique number in your subsystem to identify this communication definition. It can be set to a number from 1 to 9999.

Comparison data: compare value

A text string to compare with that received when the session is started by **transaction program** parameter. The value can be any unique string. The character string is derived from the Transaction program field of the sender CSI.

Comparison data: starting position

The character position in the string where the comparison is to start.

The starting position field is the character position in the string for comparison, and this is always 37.

Program to call

The name of program that runs the inbound message program to be called to start the session.

AMQCRC6A is a program supplied with WebSphere MQ for iSeries that sets up the environment and then calls AMQCRS6A.

Class The name and library of the class used for the steps started through this routing entry. The class defines the attributes of the routing step's running environment and specifies the job priority. Specify an appropriate class entry. Use the WRKCLS command, for example, to display existing classes or to create a new class.

More information about managing work requests from remote LU 6.2 systems is available in the *AS/400® Programming: Work Management Guide*.

Defining an LU 6.2 Connection on a Compaq NonStop Kernel server

Set up communications for MQSeries for Compaq NonStop Kernel to use LU 6.2 channels, as described in the *WebSphere MQ Intercommunication* manual and in the *MQSeries for Compaq NonStop Kernel System Administration*.

Ensure that you have an AUTOSTART(ENABLED) channel with an LU 6.2 responder process running if you are using SNAX or ICE.

Defining an LU 6.2 Connection on a z/OS server

Set up communications for WebSphere MQ for z/OS to use LU 6.2 channels, as described in the *WebSphere MQ Intercommunication* manual. Then:

1. Start the channel initiator

```
START CHINIT
```

2. Start the listener

```
START LSTR TRPTYPE(LU62) LUNAME(1uname)
```

Defining an LU 6.2 Connection on a Compaq OpenVMS Alpha client

On Compaq OpenVMS Alpha, SNA LU 6.2 supports only PU 2.0. Therefore communication can only be to PU 5.0 host. If you want to communicate with a server on a platform other than z/OS or Compaq NonStop Kernel, you must use another protocol. See Table 6 on page 92 for information about the protocols you can use.

1. Configure SNA so that an LU 6.2 conversation can be established between the WebSphere MQ client machine and the server machine. See the *WebSphere MQ Intercommunication* manual for information.
2. Set the Transaction Program name (TpName or TPNAME) to the same as the corresponding TpName in the side information on the remote queue manager on the server.
3. Establish a valid session between the two machines.

Defining a NetBIOS connection

The steps to take are detailed in the sections that follow:

On the client

Define a local NetBIOS name for the client

On the server

1. Define a local NetBIOS name for the server
2. Start a listener program

NetBIOS on a WebSphere MQ client (any suitable platform)

Define a local NetBIOS name for the client.

The local NetBIOS name that the WebSphere MQ processes use can be specified in two ways on the WebSphere MQ client. In order of precedence they are:

1. The MQNAME environment variable:

```
SET MQNAME=Your_env_Name
```

2. A queue manager configuration file (QM.INI) parameter.

If you want to use this method, you need to create a QM.INI file and add the following entry to it:

```
NETBIOS:
```

```
LocalName = Your_env_Name
```

Store the QM.INI file in the appropriate directory, depending on the platform:

OS/2 Warp and Windows 95

```
<drive>:\<dir>\mqm
```

DOS, Windows 3.1, WIN-OS/2

The root directory of the drive where the WebSphere MQ client is installed, or where the MQDATA environment variable points.

Windows

This information is in the Registry. For more information, see the WebSphere MQ Information Center.

For use with Novell NetBIOS emulation, or with NetBIOS on Windows, each WebSphere MQ process should use a different local NetBIOS name.

NetBIOS on an OS/2 Warp server

Define a local NetBIOS name for the server.

The local NetBIOS name that the WebSphere MQ processes use can be specified on the server in three ways. In order of precedence they are:

1. The -l parameter on the RUNMQLSR command.

2. The MQNAME environment variable:

```
SET MQNAME=Your_env_Name
```

3. A queue manager configuration file (QM.INI) parameter. QM.INI is located in \mqm\qmgrs\QueueManagerName

The entry in QM.INI is:

```
NETBIOS:
```

```
LocalName = Your_env_Name
```


For use with Novell NetBIOS emulation each WebSphere MQ process should use a different local NetBIOS name.

Start a listener program

Start the listener program with the RUNMQLSR command, optionally giving the local NetBIOS name (LOCALNAME) to listen on. For example:

```
RUNMQLSR -t netbios [-m MQNAME] [-s Sessions]
[-e NAMES] [-o COMMANDS] [-l LOCALNAME]
```

See the *WebSphere MQ System Administration Guide* manual for information about the options on the RUNMQLSR command.

Note: Both the sending end (WebSphere MQ client) and the receiving end (server) **must** have a local NetBIOS name defined. Ensure that all NetBIOS names used are unique in the network; if they are not, unpredictable results might occur.

NetBIOS on a Windows server

Define a local NetBIOS name for the server.

The local NetBIOS name that the WebSphere MQ processes use can be specified in three ways on the server. In order of precedence they are:

1. The -l parameter on the RUNMQLSR command. This defines the NetBIOS station name to be used with that instance of RUNMQLSR.
2. The MQNAME environment variable:

```
SET MQNAME=Your_env_Name
```

3. A parameter in the Registry. The entry in the Registry is:

```
NETBIOS:
LocalName = Your_env_Name
```

For use with NetBIOS on Windows, each WebSphere MQ process should use a different local NetBIOS name; if you are running multiple WebSphere MQ applications simultaneously on the WebSphere MQ client, they must use different NetBIOS names set using the environment variable MQNAME.

Start a listener program

Start the listener program with the RUNMQLSR command, optionally giving the local NetBIOS name (LOCALNAME) to listen on. For example:

```
RUNMQLSR -t netbios [-m MQNAME] [-s Sessions]
[-e NAMES] [-o COMMANDS] [-l LOCALNAME]
```

See the *WebSphere MQ System Administration Guide* manual for information about the options on the RUNMQLSR command.

Note: Both the sending end (WebSphere MQ client) and the receiving end (server) **must** have a local NetBIOS name defined. Ensure that all NetBIOS names used are unique in the network; if they are not, unpredictable results might occur.

Defining an SPX connection

The steps to take are detailed in the platform-specific sections that follow:

Defining an SPX connection on a WebSphere MQ client (any suitable platform)

There is no action required other than to ensure that SPX is running on your client machine.

The channel definitions that you create later will include the SPX network address, node address, and socket number of the server to which the WebSphere MQ client is sending.

Defining an SPX connection on an OS/2 Warp or Windows server

Determine the IPX/SPX network address of the server. Your network administrator has this information.

Record the node (LAN adaptor address) of the server machine.

Using the Run Listener (RUNMQLSR) command

To run the listener (which starts new MQI channels as threads), use the RUNMQLSR command. For example:

```
RUNMQLSR -t spx [-m QMNAME] [-x 5E87]
```

The square brackets indicate optional parameters:

- m QMNAME is not required for the default queue manager.
- x 5E87 is not required if the default socket number 5E86 is used.

It is possible to have more than one queue manager running on the server machine. Start a listener program for each one, on different socket numbers. For example:

```
RUNMQLSR -t spx  
RUNMQLSR -t spx -m QM2 -x 5E87
```

SPX and IPX parameters

You might need to modify some of the default SPX or IPX parameters of your environment to tune its use for WebSphere MQ. In most cases the default settings are suitable. The actual parameters, and the method of changing them, vary according to your platform and the provider of the SPX communications support. The following sections describe some of these parameters, particularly where they could influence the operation of WebSphere MQ channels and client connections.

SPX parameters on an OS/2 Warp client

Refer to the Novell Client for OS/2 Warp documentation for full details of the use and setting of NET.CFG parameters.

The following IPX/SPX parameters can be added to the Novell NET.CFG file, and might affect WebSphere MQ SPX channels and client connections.

SPX

- Sessions (default 16)

This specifies the total number of simultaneous SPX connections. Each WebSphere MQ channel, or client connection uses one session. You might need to increase this value, depending on the number of WebSphere MQ channels or client connections you need to run.

- Retry count (default 12)

This controls the number of times an SPX session will resend unacknowledged packets. WebSphere MQ does not override this value.

- Verify timeout, listen timeout, and abort timeout (milliseconds)

These timeout values adjust the KeepAlive behavior. If an SPX sending end does not receive anything within the verify timeout, it sends a packet to the receiving end. It then waits for the listen timeout for a response. If one is still not received, another packet is sent and a response is expected within the abort timeout period.

IPX

- Sockets (range 9 through 128, default 64)

This specifies the total number of IPX sockets available. WebSphere MQ channels use this resource, so depending on the number of channels and the requirements of other IPX/SPX applications, you might need to increase this value.

SPX parameters on a DOS or Windows 3.1 client

Refer to the Novell Client for DOS and Windows documentation for full details of the use and setting of NET.CFG parameters.

The following IPX/SPX parameters can be added to the Novell NET.CFG file, and might affect WebSphere MQ SPX channels and client connections.

SPX

- Connections (default 15)

This specifies the total number of simultaneous SPX connections. Each WebSphere MQ channel, or client connection uses one session. You might need to increase this value, depending on the number of WebSphere MQ channels or client connections you need to run.

IPX

- Sockets (default 20)

This specifies the total number of IPX sockets available. WebSphere MQ channels use this resource, so depending on the number of channels and the requirements of other IPX/SPX applications, you might need to increase this value.

- Retry count

This controls the number of times unacknowledged packets will be resent. WebSphere MQ does not override this value.

SPX parameters on a Windows client

Refer to the Microsoft documentation for full information about the use and setting of the NWLink SPX and IPX parameters. The IPX/SPX parameters are in the following paths in the Registry:

```
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Service\NWLinkSPX\Parameters
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Service\NWLinkIPX\Parameters
```

SPX parameters on a WebSphere MQ client for Windows 95

Refer to the Microsoft documentation for full information about the use and setting of the SPX and IPX parameters. They are accessed by selecting Network option in

SPX connection

the control panel, then double clicking on "IPX/SPX Compatible Transport".

Defining a DECnet connection

The following sections explain how to define a DECnet connection on the client and server machines, according to the platforms you are using:

Defining a DECnet connection on a WebSphere MQ client

There is no action required now, other than to ensure that DECnet is running on your client machine.

The channel definitions that you create later will include the DECnet nodename and object number, or task name, of the server to which the WebSphere MQ client is sending.

Defining a DECnet connection on a Compaq OpenVMS Alpha server

Decide on an object number or task name.

Determine the nodename of the server. Your network administrator has this information.

Receiving on DECnet Phase IV

To use DECnet Phase IV to start channels, you must configure a DECnet object as follows:

1. Create a file that has the DCL command to start the DECnet receiver program, `amqcrsta.exe`. Place this file in the `SYS$MANAGER` directory as follows:

```
$ create sys$manager:mqrecvdecnet.com
$ mcr amqcrsta.exe [-m Queue_Man_Name] -t DECnet
Ctrl-Z
$
```

If you have multiple queue managers you **must** make a new file and DECnet object for each queue manager.

2. Create a DECnet object to start the receiving channel program automatically:

```
$ MCR NCP
NCP> define object MQSERIES
Object number          (0-255): 0
File name              (filename):sys$manager:mqrecvdecnet.com
Privileges (List of VMS privileges):
Outgoing connect privileges (List of VMS privileges):
User ID                (1-39 characters): mqm
Password              (1-39 characters): mqseries
                        (note: you must supply the correct password for MQM)
Account                (1-39 characters):
Proxy access (INCOMING, OUTGOING, BOTH, NONE, REQUIRED):
NCP> set known objects all
NCP> exit
```

The preceding example should be amended to use proxy user identifiers rather than actual user identifiers. This will prevent any unauthorized access to the database. Information on how to set up proxy identifiers is given in the *Digital DECnet for OpenVMS Networking* manual.

3. Ensure that all known objects are set when DECnet is started.

Receiving on DECnet OSI

Configure for WebSphere MQ channel objects:

1. Start the NCL configuration interface:

DECnet connection

```
$ MC NCL  
NCL>
```

2. Create a session control application entity:

```
NCL> create session control application MARK  
NCL> set sess con app MARK address {name=MARK{  
NCL> set sess con app MARK image name -  
_SYS$SPECIFIC:[MQS_SERVER]MARK.COM  
NCL> set sess con app MARK user name "MQM"  
NCL> set sess con app MARK node synonym true  
  
NCL> show sess con app MARK all [characteristics]
```

Note that user-defined values are in uppercase.

3. Create the com file (MARK.COM above) as for DECnet Phase IV.
4. The log file for the object is net\$server.log in the sys\$login directory for the user name specified for the application.
5. The MQSC configuration for CONNAME, as for DECnet Phase IV, is:
NODE(APP_OR_OBJ_NAME)

Chapter 6. Verifying the installation

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

This chapter tells you how to use the supplied sample PUT and GET programs to verify that a WebSphere MQ client has been installed correctly, by guiding you through the following tasks:

1. "Setting up the server" on page 114
2. "Setting up the WebSphere MQ client" on page 116
3. "Putting a message on the queue" on page 117
4. "Getting the message from the queue" on page 119
5. "Ending verification" on page 120

The installation used for the example

These instructions assume that:

- The full WebSphere MQ product has been installed on a server:
 - The Base Product and the Client Attachment feature on z/OS
 - The full MQSeries for VSE/ESA product on VSE/ESA platforms
 - The *Base Product and Server* on other platforms
- The WebSphere MQ client software has been installed on a client machine, and WebSphere MQ client files have been transferred, where necessary.

The transmission protocol used in the example is TCP/IP. It is assumed that you have TCP/IP configured on the server and the WebSphere MQ client machines, and that it has been initialized on both the machines. There is more information about this in Chapter 5, "Configuring communication links" on page 91.

Compiled samples AMQSPUTC and AMQSGETC (AMQSPUTW and AMQSGETW for Windows 3.1) are included in the WebSphere MQ client directories that you installed, either directly or by copying the files across as described in "Installing the MQSeries server" on page 81.

What the example shows

The following example shows how to create a queue manager called *queue.manager.1* (not on z/OS), a local queue called *QUEUE1*, and a server-connection channel called *CHANNEL1* on the server. It shows how to create the client-connection channel on the WebSphere MQ client workstation; and how to use the sample programs to put a message onto a queue, and then get the message from the queue.

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

Security

The example does *not* address any client security issues. See Chapter 7, "Setting up WebSphere MQ client security" on page 123 for details if you are concerned with WebSphere MQ client security issues.

Setting up the server

This section does not apply to Compaq NonStop Kernel, OS/400, VSE/ESA, or z/OS. For information about setting up the server on these platforms, see the sections that follow.

Note: If you are using WebSphere MQ for Windows, you can use the WebSphere MQ Explorer as well as the command line to carry out most WebSphere MQ tasks. Details of how to use WebSphere MQ Explorer for the verification example are given in “Setting up a Windows server using WebSphere MQ Explorer” on page 115.

Create a directory to hold working files, for example `mqverify`, and make this the current directory. Then follow the steps below to set up the server workstation.

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

```
crtmqm -q queue.manager.1
```

2. Start the queue manager by entering the following command:

```
strmqm
```

3. If you are using a Version 5.1 or later product go on to the next step.

If you are using an MQSeries non-Version 5 product, define the default system objects by entering the following command:

```
runmqsc queue.manager.1 <PATH/amqscoma.tst >defobj.out
```

where PATH depends on the platform you are using; see the *System Administration Guide* or *System Management Guide* for your platform for the value of PATH. When this command has completed examine the file `defobj.out` that is written to the current directory, to confirm that all the default objects were created successfully. The last line of this file should read:

```
All valid MQSC commands processed
```

If there are commands that cannot be processed, you need to check your server installation. See the *System Administration Guide* or *System Management Guide* for your server platform.

4. Start WebSphere MQ Script (MQSC) Commands by entering the following command:

```
runmqsc
```

MQSC does not provide a prompt, but should respond with the message:

```
Starting MQSeries Commands
```

5. Create a local queue by entering the following command:

```
DEFINE QLOCAL(QUEUE1)
```

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

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

See “Access control” on page 125 for information about MCAUSER.

7. Stop MQSC by typing `end` (on non-Version 5 MQSeries products press `Ctrl+D`) and then `Enter`.
8. Configure the system to start channels

On OS/2 Warp or Windows

Start a listener by entering the following command at the command prompt:


```
RUNMQLSR -t tcp -m queue.manager.1
```

On UNIX systems

Use one of the following methods:

- Start a listener by entering the following command:

```
RUNMQLSR -t tcp -m queue.manager.1
```

or

- Configure the **inetd** daemon to start the MQI channels. See “Defining a TCP/IP connection on a UNIX system server” on page 96 for details of how to do this.

On Compaq OpenVMS Alpha

See the *MQSeries for Compaq (DIGITAL) OpenVMS System Management Guide* for information about how to configure TCP/IP services to start channels.

Setting up a Windows server using WebSphere MQ Explorer

Create a directory to hold working files, for example `mqverify`, and make this the current directory. Then follow the steps below to set up the server connection.

- Create a default queue manager
 1. Open WebSphere MQ Explorer from the Start Menu.
 2. Right-click the folder called Queue Managers, and select New.
 3. In the first entry field, type the queue manager name, *queue.manager.1*.
 4. Make this the default queue manager.
 5. Click Next twice.
 6. Create the server connection.
 7. Click Next.
 8. Enter a port number of 1414 and click finish.
- Create a local queue
 1. Expand the queue manager you have just created and right-click queues.
 2. Select new local queue.
 3. Enter the queue name and click OK.
- Define the server-connection channel
 1. Select Advanced, and right-click Channels.
 2. Select New Server-connection channel.
 3. Enter the channel name, CHANNEL1
- Run the listener

The listener will have automatically started when the queue manager was set up. You can verify this by opening MQServices, and selecting *queue.manager.1*. The listener will be running.

Setting up the server (OS/400)

These instructions assume that no queue manager or other WebSphere MQ objects have been defined. Follow these steps:

1. Create a queue manager by entering the following command:


```
CRTMQM MQMNAME('queue.manager.1')
```
2. Start the queue manager by entering the following command:


```
STRMQM MQMNAME('queue.manager.1')
```
3. Create a local queue by entering the following command:

Verifying the installation

```
CRTMQMQ QNAME(Queue1) QTYPE(*LCL)
```

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

```
CRTMQMCHL CHLNAME(Channel1) CHLTYPE(*SVRCN) TRPTYPE(*TCP)  
MCAUSRID('QMQM')
```

Note: QMQM is the default user ID.

5. Start the listener by entering the following command:

```
STRMQMLSR MQMNAME('queue.manager.1')
```

Setting up the server (z/OS)

Customize your WebSphere MQ for z/OS installation as described in the *WebSphere MQ for z/OS System Setup Guide*. This includes defining the default system objects and enabling distributed queuing (without CICS). You do not require the Batch/TSO, CICS, or IMS™ adapters to run as servers for WebSphere MQ applications running on a client. However, depending on how you choose to issue commands, you might need the Batch/TSO adapter and the operations and control panels to perform administration for clients.

Follow the steps below. You can use any of the valid command input methods to issue the WebSphere MQ commands (MQSC) shown.

1. Start the queue manager by entering the following command:

```
START QMGR
```

2. Create a local queue by entering the following command:

```
DEFINE QLOCAL(Queue1)
```

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

```
DEFINE CHANNEL(Channel1) CHLTYPE(SVRCONN) TRPTYPE(TCP) MCAUSER(' ')
```

4. Start the channel initiator by entering the following command:

```
START CHINIT
```

5. Start the listener by entering the following command:

```
START LSTR TRPTYPE(TCP) PORT(port-number)
```

Setting up the server (VSE/ESA)

Verify your VSE/ESA server installation as described in the *MQSeries Installation Verification Test* section of the *MQSeries for VSE/ESA System Management Guide*.

Setting up the WebSphere MQ client

When an WebSphere MQ application is run on the WebSphere MQ client, it requires the name of the MQI channel, the communication type, and the address of the server to be used. You provide this by defining a client-connection channel. The name used must be same as the name used for the server-connection channel defined on the server. In this example, two methods are used to define the client-connection channel;

- “Defining a client-connection channel using MQSERVER” on page 117
- “Defining a client-connection channel using WebSphere MQ Explorer” on page 117

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

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

Defining a client-connection channel using MQSERVER

Create a client-connection channel by setting the MQSERVER environment variable. (For more information, see Chapter 10, “Using WebSphere MQ environment variables” on page 143).

- For DOS, OS/2 Warp, Windows, Windows 3.1, and Windows 95 clients, enter the following command:
`SET MQSERVER=CHANNEL1/TCP/server-address(port)`
- For UNIX clients, enter the following command:
`export MQSERVER=CHANNEL1/TCP/'server-address(port)'`
- For VM/ESA clients, enter the following command:
`GLOBALV SELECT CENV SETLP MQSERVER SYSTEM.DEF.SVRCONN/TCP/server-address(port)`

Where:

server-address is the TCP/IP hostname of the server
(port) is the TCP/IP port number the server is listening on

If you do not give a port number, WebSphere MQ uses the one specified in the QM.INI file, or the registry for Windows. If no value is specified in the QM.INI file, or the registry for Windows, WebSphere MQ uses the port number identified in the TCP/IP services file for the service name MQSeries. If this entry in the services file does not exist, a default value of 1414 is used. It is important that the port number used by the client and the port number used by the server listener program are the same.

Defining a client-connection channel using WebSphere MQ Explorer

You can use WebSphere MQ Explorer to define the client-connection if you are setting up the client and server on the same machine. The following steps will define the client-connection end of the channel.

1. Select the queue manager, *queue.manager.1*
2. Select **advanced** then **client-connection** then **new**
Select a name for the client-connection, and enter the following as the connection name:
`server-connection(port)`

Where:

server-address is the TCP/IP hostname of the server
(port) is the TCP/IP port number the server is listening on

Putting a message on the queue

On the WebSphere MQ client workstation, put a message on the queue using the AMQSPUTC sample program (amqsputw on Windows 3.1, AMQSPUT0 on VM/ESA).

Information on errors at this stage can be found in “Error messages with WebSphere MQ clients” on page 172.

Verifying the installation

On the WebSphere MQ client workstation (not VM/ESA or Windows 3.1)

1. From a command prompt window, change to the directory containing the sample program amqsputc.exe. This is in the /samp/bin directory, or the \bin directory for some Version 5.1 products. Then enter the following command:
`amqsputc QUEUE1 qmgr`

where qmgr is the name of the queue manager on the server.

2. The following message is displayed:
`Sample AMQSPUT0 start`
`target qname is QUEUE1`
3. Type some message text and then press Enter twice.
4. The following message is displayed:
`Sample AMQSPUT0 end`
5. The message is now on the queue.

On the WebSphere MQ client workstation (VM/ESA)

The sample programs provided with the WebSphere MQ client for VM/ESA must be compiled on your system. For details see the *VM/ESA CMS Application Development Guide*.

1. Enter the following command:

```
AMQSPUT0 QUEUE1 qmgr
```

where qmgr is the name of the queue manager on the server.

2. The following message is displayed:
`Sample AMQSPUT0 start`
`target qname is QUEUE1`
3. Type some message text and then press Enter twice.
4. The following message is displayed:
`Sample AMQSPUT0 end`
5. The message is now on the queue.

On the WebSphere MQ client workstation (Windows 3.1)

This program has no visible interface. All messages are put in the output file, not to stdout.

This program takes four parameters, all are required:

1. The name of the output file
2. The name of the input file
3. The name of the queue manager
4. The name of the target queue

To run AMQSPUTW from the Windows Program Manager:

1. Select the menu item File/Run...
2. On the Run dialog, enter the program name followed by the parameters. For example:

```
amqspuw outfile.out infile.in qmgr QUEUE1
```

Where:

outfile.out is used to hold the messages generated when the program runs.

infile.in contains the data to be put onto the target queue. Each line of data is put as a message. infile.in must be an ASCII file.

qmgr is the name of the queue manager on the server.

It is important to always look in the output file to see what has happened, because there is no visible indication of success or failure when you run this program.

Note: The AMQSPUTC (or AMQSPUTW) sample program starts the channel between the client and the server. When you have put the message on the queue, the sample program ends and the channel between the client and server also ends automatically.

Getting the message from the queue

On the WebSphere MQ client workstation, get a message from the queue using the amqsgetc sample program (AMQSGETW on Windows 3.1, AMQSGET0 on VM/ESA):

On the WebSphere MQ client workstation (not VM/ESA or Windows 3.1)

1. Change to the directory containing the sample programs, and then enter the following command:
`amqsgetc QUEUE1 qmgr`

Where qmgr is the name of the queue manager on the server.

2. The message on the queue is removed from the queue and displayed.

On the WebSphere MQ client workstation (VM/ESA)

The sample programs provided with the WebSphere MQ client for VM/ESA must be compiled on your system. For details see the *VM/ESA CMS Application Development Guide*.

1. Enter the following command:
`AMQSGET0 QUEUE1 qmgr`

Where qmgr is the name of the queue manager on the server.

2. The message on the queue is removed from the queue and displayed.

On the WebSphere MQ client workstation (Windows 3.1)

This program has no visible interface. All messages are put in the output file, not to stdout.

This program takes three parameters, all are required:

1. The name of the output file
2. The name of the queue manager
3. The name of the target queue

To run AMQSGETW from the Windows Program Manager:

1. Select the menu item File/Run...
2. On the Run dialog, enter the program name followed by the parameters. For example:

Verifying the installation

```
amqsgetw outfile.out qmgr QUEUE1
```

where:

outfile.out is used to hold the messages generated when the program runs.

qmgr is the name of the queue manager on the server

It is important to always look in the output file to see what has happened because there is no visible indication of success or failure when you run this program.

Ending verification

The verification process is now complete.

On z/OS, you can stop the queue manager on the server by issuing the STOP CHINIT command followed by the STOP QMGR command.

On other platforms except VSE/ESA, you can stop the queue manager on the server by typing:

```
endmqm queue.manager.1
```

If you want to delete the queue manager on the server (not z/OS or VSE/ESA) type:

```
dltmqm queue.manager.1
```

For VSE/ESA, see the information about the WebSphere MQ installation verification test in the *MQSeries for VSE/ESA System Management Guide*.

Part 3. System administration

Chapter 7. Setting up WebSphere MQ client

security	123
Authentication	123
Environment variables	124
User IDs	124
Access control	125

Chapter 8. Using channels

What is a channel?	127
Message Channels	127
MQI Channels	127
Defining MQI channels	128
Automatically defined channels	128
User defined channels	129
Creating one definition on the WebSphere MQ client and the other on the server.	129
On the server	129
Creating a queue manager and starting MQSC on the server	129
Defining the server-connection channel.	130
On the WebSphere MQ client	130
Using MQSERVER	130
Using the MQCNO structure on an MQCONN call	132
Creating both definitions on the server.	132
Defining the server-connection channel.	132
Defining the client-connection channel	133
Accessing client-connection channel definitions	133
Client channel definition table.	134
Migrating to a later release level of WebSphere MQ	135
Channel exits	136
Path to exits.	136
Connecting a client to a queue-sharing group	136
Connecting to a specific queue manager	137
Connecting to the generic interface	137
Creating channel definitions	137
Stopping channels.	137

Chapter 9. The Secure Sockets Layer (SSL) on

WebSphere MQ clients	139
Specifying that an MQI channel uses SSL	139
Specifying the location of LDAP servers that hold certificate revocation lists (CRLs)	139
When a WebSphere MQ client application issues an MQCONN call	140
Using a client channel definition table	140
Using Active Directory on Windows.	141

Chapter 10. Using WebSphere MQ environment variables

MQCCSID	144
MQCHLLIB	144
MQCHLTAB	145
MQDATA (DOS and Windows 3.1 only)	145
MQNAME	146

MQ_PASSWORD (DOS, OS/2 Warp, Windows 3.1, Windows 95, and Windows 98 only).	146
MQSERVER	146
TCP/IP default port	147
SPX default socket.	147
Examples of using MQSERVER	147
Canceling MQSERVER	148
MQTRACE (DOS, Windows 3.1, and VM/ESA only)	148
MQ_USER_ID (DOS, OS/2 Warp, Windows 3.1, Windows 95, and Windows 98 only).	148
MQSWORKPATH (OS/2 Warp only)	148

Chapter 7. Setting up WebSphere MQ client security

You must consider WebSphere MQ client security, so that the client applications do not have unrestricted access to resources on the server.

There are two aspects to security between a client application and its queue manager server: authentication and access control.

Note: WebSphere MQ V5.3 provides Secure Sockets Layer (SSL) support for WebSphere MQ clients in the following products:

- WebSphere MQ for AIX
- WebSphere MQ for HP-UX
- WebSphere MQ for Linux for Intel
- WebSphere MQ for Linux for zSeries
- WebSphere MQ for Solaris
- WebSphere MQ for Windows

For more information on SSL support for WebSphere MQ and WebSphere MQ clients, see *WebSphere MQ Security*.

Authentication

There are three levels of security to consider, as shown in the following diagram. MCA is a Message Channel Agent.

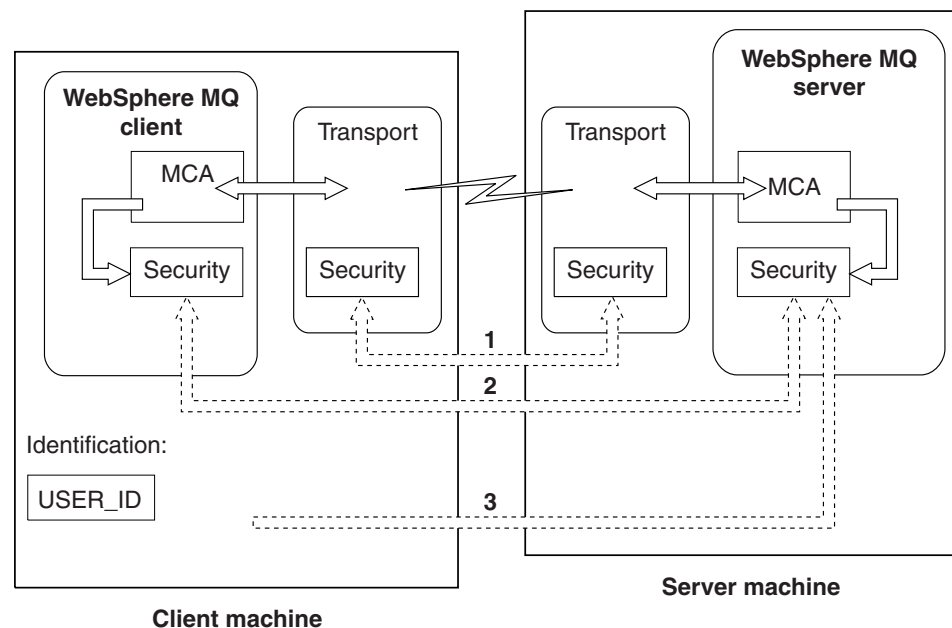


Figure 4. Security in a client-server connection

1. Transport level

This is the same as for two WebSphere MQ queue managers (server to server) and is described in the *WebSphere MQ Intercommunication* manual.

2. Channel security exits

Security

The channel security exits for client to server communication can work in the same way as for server to server communication. A protocol independent pair of exits provide mutual authentication of both the client and the server. A full description is given in the *WebSphere MQ Intercommunication* manual.

DCE security exits are supplied with WebSphere MQ products for clients on AIX, HP-UX, OS/2 Warp, Solaris, Windows, and Windows 95. These are described in the *WebSphere MQ Intercommunication* manual. If no security exits are used, access to WebSphere MQ objects is determined by the server-connection channel definition. See “Access control” on page 125 for details.

3. Identification passed to a channel security exit

In client to server communication, the channel security exits do not have to operate as a pair. The exit on the WebSphere MQ client side can be omitted. In this case the user ID is placed in the channel descriptor (MQCD) and the security exit can alter it, if required. Some clients also send additional information to assist identification.

- For clients on UNIX systems, Windows 2000, Windows NT, and Windows XP, the user ID that is passed to the server is the currently logged-on user ID on the client. In addition, a client on Windows 2000, Windows NT, or Windows XP passes the security ID of the currently logged-on user.
- For clients on DOS, OS/2 Warp, Windows 3.1, Windows 95, and Windows 98, the user ID is specified by the environment variable MQ_USER_ID. In addition, for these clients only, the environment variable MQ_PASSWORD is transmitted to the server.

The values of the user ID and, if available, the password or security ID, can be used by the server security exit to establish the identity of the WebSphere MQ client.

Environment variables

For WebSphere MQ clients on DOS, OS/2 Warp, Windows 3.1, Windows 95, and Windows 98, if a security exit is not defined, the values of two environment variables MQ_USER_ID and MQ_PASSWORD are transmitted to the server. The values are passed in the channel definition (MQCD) to the server security exit when the exit is invoked. The values can then be used by the exit to establish the identity of the WebSphere MQ client.

On these platforms, set the variables in the environment in which the WebSphere MQ client is going to run. Note that MYUSERID and MYPASSWORD must be in uppercase if the client is going to communicate with an WebSphere MQ server on OS/400. For example:

```
SET MQ_USER_ID=MYUSERID
SET MQ_PASSWORD=MYPASSWORD
```

The MQ_USER_ID and MQ_PASSWORD environment variables are not supported on UNIX systems, Windows 2000, Windows NT, and Windows XP. On these platforms, identification is established when the currently logged-on user ID of the client is passed automatically to the server.

User IDs

If the WebSphere MQ client is on Windows 2000, Windows NT, or Windows XP, and the WebSphere MQ server is also on one of these platforms and has access to the domain on which the client user ID is defined, WebSphere MQ supports user IDs of up to 20 characters.

If the WebSphere MQ server is on Windows 2000, Windows NT, or Windows XP, and the client is on a platform that uses the environment variable for specifying the user ID, the user ID can be in the format `user@domain`. The WebSphere MQ server then retrieves user account information from the specified NT domain. In this case, the maximum length for the user ID is 64 characters.

If the WebSphere MQ server is on Windows 2000, Windows NT, or Windows XP, and the client is on a platform that uses the environment variable for specifying the user ID, but no domain is specified, the WebSphere MQ server attempts to retrieve user account information from its primary domain or trusted domains. In this case, the maximum length for the user ID is 20 characters.

A WebSphere MQ for Windows server does not support the connection of a Windows 2000, Windows NT, or Windows XP client if the client is running under a user ID that contains the @ character, for example, `abc@d`. The return code to the **MQCONN** call at the client is **MQRC_NOT_AUTHORIZED**.

On all other platforms and configurations, the maximum length for user IDs is 12 characters.

Access control

Access control in WebSphere MQ is based upon the user identifier associated with the process making MQI calls. For WebSphere MQ clients, the process that issues the MQI calls is the server-connection MCA. The user identifiers used by the server-connection MCA are that contained in the **MCAUserIdentifier** and **LongMCAUserIdentifier** fields of the MQCD. The contents of these fields are determined by:

- Any values set by security exits
- The user ID (for clients on UNIX systems, Windows 2000, Windows NT, and Windows XP) or **MQ_USER_ID** environment variable (for other clients) from the client
- **MCAUSER** (in the server-connection channel definition)

Depending upon the combination of settings of the above, the user-identifier fields are set to appropriate values. If a server-connection security exit is provided, the user-identifier fields can be set by the exit. Otherwise they are determined as follows:

- If the server-connection channel **MCAUSER** attribute is nonblank, this value is used.
- If the server-connection channel **MCAUSER** attribute is blank, the user ID received from the client is used. However, for the clients that use the **MQ_USER_ID** environment variable to supply the user ID, it is possible that no environment variable is set. In this case, the user ID that started the server-connection channel is used. For Java client connections, if the client application does not provide a user ID then no client user identification is provided to the server.

For the Windows 95 and Windows 98 clients, if the **MQ_USER_ID** environment variable has not been set, the current Windows logged-on user ID provides user identification for the client. If the **MQ_USER_ID** environment variable has not been set and the current user has not logged on to Windows, no client user identification is provided to the server.

Security

When the user-identifier fields are derived from the user ID that started the server-connection channel, the following value is used:

- For z/OS, the user ID assigned to the channel initiator started task by the z/OS started procedures table. See the *WebSphere MQ for z/OS System Setup Guide* for more information.
- For TCP/IP (non-z/OS), the user ID from the `inetd.conf` entry, or the user ID that started the listener.
- For SNA (non-z/OS), the user ID from the SNA Server entry or (if there is none) the incoming attach request, or the user ID that started the listener.
- For NetBIOS or SPX, the user ID that started the listener.

If any server-connection channel definitions exist that have the MCAUSER attribute set to blank, clients can use this channel definition to connect to the queue manager with access authority determined by the user ID supplied by the client. This might be a security exposure if the system on which the queue manager is running allows unauthorized network connections. The WebSphere MQ default server-connection channel (SYSTEM.DEF.SVRCONN) has the MCAUSER attribute set to blank. **To prevent unauthorized access**, update the MCAUSER attribute of the default definition with a user ID that has no access to WebSphere MQ objects.

When you define a channel with `runmqsc`, the MCAUSER attribute is changed to uppercase unless the user ID is contained within single quotation marks.

For servers on UNIX systems and Windows, the content of the `MCAUserIdentifier` field that is received from the client is changed to lowercase.

For servers on OS/400, the content of the `LongMCAUserIdentifier` field that is received from the client is changed to uppercase.

For servers on UNIX systems, the content of the `LongMCAUserIdentifier` field that is received from the client is changed to lowercase.

Chapter 8. Using channels

This chapter discusses the following topics:

- “What is a channel?”
- “Defining MQI channels” on page 128
- “Creating a queue manager and starting MQSC on the server” on page 129
- “Creating one definition on the WebSphere MQ client and the other on the server” on page 129
- “Creating both definitions on the server” on page 132
- “Channel exits” on page 136
- “Connecting a client to a queue-sharing group” on page 136
- “Migrating to a later release level of WebSphere MQ” on page 135

What is a channel?

A channel is a logical communication link between an WebSphere MQ client and an WebSphere MQ server, or between two WebSphere MQ servers. A channel has two definitions: one at each end of the connection. The same *channel name* must be used at each end of the connection, and the *channel type* used must be compatible.

There are two categories of channel in WebSphere MQ, with different channel types within these categories:

Message Channels

A message channel is a one-way link. It connects two queue managers via *message channel agents* (MCAs). Its purpose is to transfer messages from one queue manager to another. Message channels are not required by the client server environment. More information on message channels can be found in the *WebSphere MQ Intercommunication*

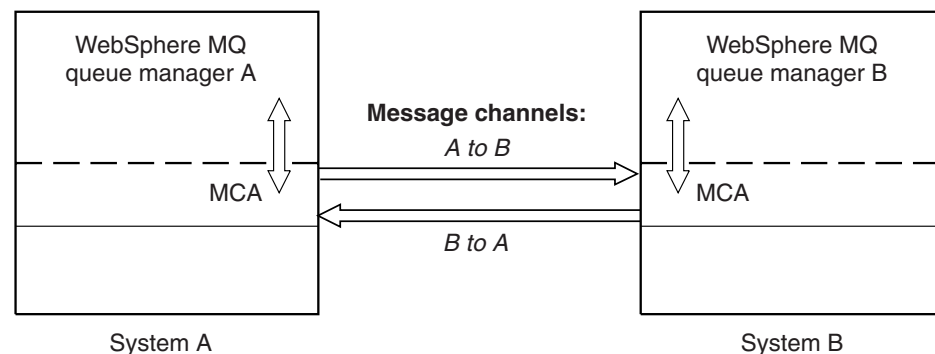


Figure 5. Message channels between two queue managers

MQI Channels

An MQI channel connects a WebSphere MQ client to a queue manager on a server machine, and is established when you issue an **MQCONN** or **MQCONNEX** call. It is a two-way link and is used for the transfer of MQI calls and responses only, including **MQPUT** calls that contain message data. There are different ways of

Using channels

creating and using the channel definitions (see “Defining MQI channels”).

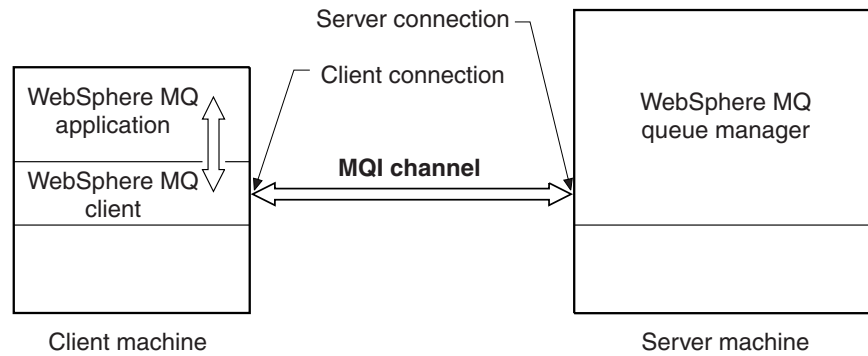


Figure 6. Client-connection and server-connection on an MQI channel

An MQI channel can be used to connect a client to a single queue manager, or to a queue manager that is part of a queue-sharing group (see “Connecting a client to a queue-sharing group” on page 136).

There are two channel types for MQI channel definitions. They define the bi-directional MQI channel.

Client-connection channel

This type is for the WebSphere MQ client.

Server-connection channel

This type is for the server running the queue manager, with which the WebSphere MQ application, running in an WebSphere MQ client environment, will communicate.

Defining MQI channels

To create a new channel, you have to create **two** channel definitions, one for each end of the connection, using the same channel name and compatible channel types. In this case, the channel types are *server-connection* and *client-connection*.

Automatically defined channels

WebSphere MQ products, and MQSeries Version 5.1, or later, products (but not z/OS) include a feature that can automatically create a channel definition on the server if one does not exist.

If an inbound attach request is received from a client and an appropriate server-connection definition cannot be found on that queue manager, WebSphere MQ creates a definition automatically and adds it to the queue manager. Automatic definitions are based on two default definitions supplied with WebSphere MQ: SYSTEM.AUTO.RECEIVER and SYSTEM.AUTO.SVRCONN. You enable automatic definition of server-connection definitions by updating the queue manager object using the ALTER QMGR command (or the PCF command Change Queue Manager).

For more information about the automatic creation of channel definitions, see the *WebSphere MQ Intercommunication* book.

User defined channels

Where the server does not automatically define channels there are two ways of creating the channel definitions and giving the WebSphere MQ application on the WebSphere MQ client machine access to the channel.

These two methods are described in detail in this chapter:

1. Create one channel definition on the WebSphere MQ client and the other on the server.

This applies to any combination of WebSphere MQ client and server platforms. Use it when you are getting started on the system, or to test your setup.

See “Creating one definition on the WebSphere MQ client and the other on the server” for details on how to use this method.

2. Create both channel definitions on the server machine.

Use this method when you are setting up multiple channels and WebSphere MQ client machines at the same time.

See “Creating both definitions on the server” on page 132 for details on how to use this method.

Whichever method you choose, first you will need to create a queue manager and start WebSphere MQ Script (MQSC) commands (see “Creating a queue manager and starting MQSC on the server”).

Creating one definition on the WebSphere MQ client and the other on the server

You can use WebSphere MQ Script (MQSC) commands to define a server-connection channel on the server machine. Because MQSC commands are not available on a machine where WebSphere MQ has been installed as a WebSphere MQ client only, you must use different ways of defining a client-connection channel on the client machine.

On the server

If your server platform is not z/OS, you first create and start a queue manager and then start MQSC commands.

Creating a queue manager and starting MQSC on the server

1. Create a queue manager, called QM1 for example:

```
crtmqm QM1
```

2. Start the queue manager:

```
strmqm QM1
```

3. Start MQSC commands.

On all platforms except VSE/ESA, start MQSC by entering the following command:

```
runmqsc QM1
```

Note: Creating MQM on Compaq NonStop Kernel requires extra mandatory parameters, See the *MQSeries for Compaq NonStop Kernel System Administration* for more details.

Using channels

If you are using one of the MQSeries Version 5.1 or later products, the default objects are defined automatically when you create the queue manager. For other MQSeries products, define the default objects as explained in “Setting up the server” on page 114.

MQSC is not supported on VSE/ESA. However, you can start MQSeries for VSE/ESA in any of the following ways:

- Use the MQSeries for VSE/ESA System Management transaction MQMT (panel 2.4).
- Use the transactions MQSE and MQIT.
- Automatically through CICS/VSE® initialization.

See the *MQSeries for VSE/ESA System Management Guide* for more information.

Note: Two WebSphere MQ administration tools are provided with MQSeries for Windows NT Version 5.1 and later. The first is a graphical administration tool called the WebSphere MQ Explorer. The WebSphere MQ Explorer runs as a snap-in within the Microsoft Management Console (MMC) under Windows, and is installed as part of the server component of WebSphere MQ for Windows. The second is a Web Administration tool which provides administrators with secure sessions through which they can administer an WebSphere MQ network. For more information about these, see the *WebSphere MQ System Administration Guide* manual.

Defining the server-connection channel

Define a channel with your chosen name and a channel type of *server-connection*. This channel definition is associated with the queue manager running on the server.

For example:

```
DEFINE CHANNEL(CHAN1) CHLTYPE(SVRCONN) TRPTYPE(TCP) +  
DESCR('Server-connection to Client_1')
```

On the WebSphere MQ client

There are two ways of defining a client-connection channel on the client machine.

Using MQSERVER

You can use the MQSERVER environment variable to specify a simple definition of a client-connection channel. It is simple in the sense that you can specify only a few attributes of the channel using this method.

- Specify a simple channel definition on DOS, OS/2 Warp, Windows, Windows 3.1, or Windows 95, as follows:

```
SET MQSERVER=ChannelName/TransportType/ConnectionName
```

Or, for OS/2 Warp using LU 6.2 only, as follows:

```
SET MQSERVER=ChannelName/LU62/ConnectionName/ModeName/TpName
```

- Specify a simple channel definition on UNIX systems as follows:

```
export MQSERVER=ChannelName/TransportType/ConnectionName
```

Where:

- ChannelName must be the same name as defined on the server. It cannot contain a forward slash.
- TransportType can be one of the following, depending on your WebSphere MQ client platform (see Table 1 on page 12):

- LU62
- TCP
- NETBIOS
- SPX
- DECNET

Note: On UNIX systems the TransportType is case sensitive and must be uppercase. An **MQCONN** or **MQCONN**X call will return 2058 if the TransportType is not recognized

- ConnectionName is the name of the server machine as defined to the communications protocol (TransportType).
- For OS/2 Warp, ModeName is the LU 6.2 mode name and TpName is the transaction program name.

Note: #INTER should be the ModeName of choice for most occasions. You can also specify Modename and TpName in your Communications Manager/2 profile. ModeName and TpName are fully described in the *WebSphere MQ Intercommunication* manual.

For example, on OS/2 Warp:

```
SET MQSERVER=CHAN1/TCP/MCID66499
```

or, on a UNIX system:

```
export MQSERVER=CHAN1/TCP/'MCID66499'
```

Note: To change the TCP/IP port number, see “MQSERVER” on page 146.

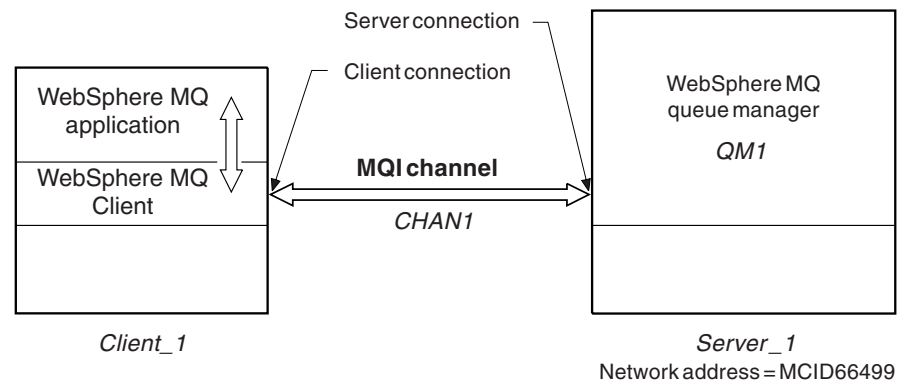


Figure 7. Simple channel definition

Some more examples of a simple channel definition on DOS, OS/2 Warp, Windows, Windows 3.1, and Windows 95 are:

```
SET MQSERVER=CHAN1/TCP/9.20.4.56
SET MQSERVER=CHAN1/NETBIOS/BOX643
```

Some examples of a simple channel definition on a UNIX system are:

```
export MQSERVER=CHAN1/TCP/'9.20.4.56'
export MQSERVER=CHAN1/LU62/BOX99
```

Where BOX99 is the LU 6.2 ConnectionName.

On the WebSphere MQ client, all **MQCONN** or **MQCONN**X requests then attempt to use the channel you have defined.

Using channels

Note: For more information on the MQSERVER environment variable see Chapter 10, “Using WebSphere MQ environment variables” on page 143.

Using the MQCNO structure on an MQCONN call

A WebSphere MQ client application can use the connect options structure, MQCNO, on an MQCONN call to reference a channel definition structure, MQCD, that contains the definition of a client-connection channel.

In this way, the client application can specify the *ChannelName*, *TransportType*, and *ConnectionName* attributes of a channel at run time, and this enables the client application to connect to multiple server queue managers simultaneously. This is not possible if you define a channel using the MQSERVER environment variable.

A client application can also specify attributes of a channel such as *MaxMsgLength* and *SecurityExit*. This allows the client application to specify values for the attributes that are not the default values, and allows channel exit programs to be called at the client end of an MQI channel.

If a channel uses the Secure Sockets Layer (SSL), a client application can also provide information relating to SSL in the MQCD structure. Additional information relating to SSL can be provided in the SSL configuration options structure, MQSCO, which is also referenced by the MQCNO structure on an MQCONN call.

For more information about the MQCNO, MQCD, and MQSCO structures, see the *WebSphere MQ Application Programming Reference*.

Note: A sample connect program called amqscnxc demonstrates the use of this function.

Creating both definitions on the server

First define a server-connection channel and then define a client-connection channel.

On the server machine, use MQSC commands to define the channels.

Note: This method cannot be used on MQSeries for VSE/ESA.

Defining the server-connection channel

1. If your server platform is not z/OS, you first create and start a queue manager and then start MQSC commands. See “Creating a queue manager and starting MQSC on the server” on page 129
2. On the server machine, define a channel with your chosen name and a channel type of *server-connection*.

For example:

```
DEFINE CHANNEL(CHAN2) CHLTYPE(SVRCONN) TRPTYPE(TCP) +  
DESCR('Server-connection to Client_2')
```

This channel definition is associated with the queue manager running on the server.

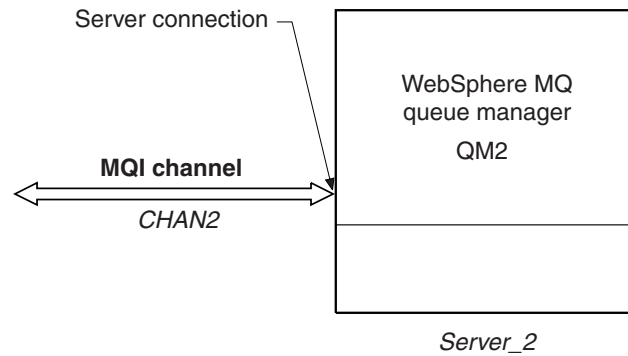


Figure 8. Defining the server-connection channel

Defining the client-connection channel

Define a channel with the *same* name and a channel type of *client-connection*.

You must state the connection name (CONNNAME). For TCP/IP this is the network address of the server machine. It is also advisable to specify the queue manager name (QMNAME) to which you want your WebSphere MQ application, running in the client environment, to connect. See Chapter 13, “Running applications on WebSphere MQ clients” on page 163.

For example:

```
DEFINE CHANNEL(CHAN2) CHLTYPE(CLNTCONN) TRPTYPE(TCP) +
CONNNAME(9.20.4.26) QMNAME(QM2) DESCR('Client-connection to Server_2')
```

On platforms other than z/OS, this channel definition is stored in a file called the client channel definition table, which is associated with the queue manager running on the server. The client channel definition table can contain more than one client-connection channel definition. For more information about the client channel definition table, and for the corresponding information about how client-connection channel definitions are stored on z/OS, see “Client channel definition table” on page 134.

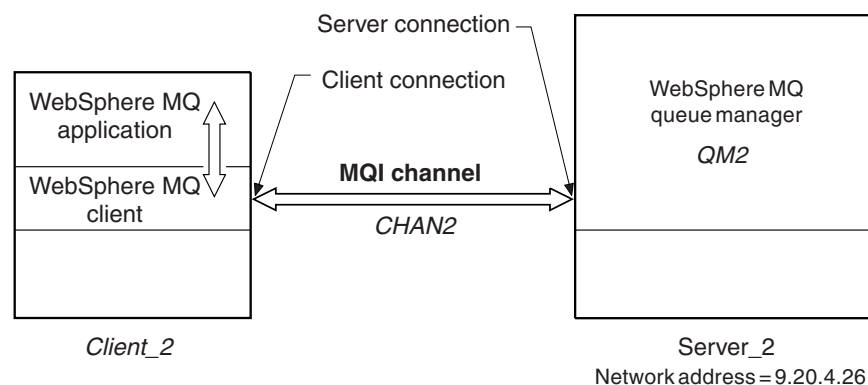


Figure 9. Defining the client-connection channel

Accessing client-connection channel definitions

If the client channel definition table on the server machine cannot be accessed from the client machine as a shared file, you must copy the client channel definition

Using channels

table to the client machine as a binary file. On the client machine, you can then use the environment variables, MQCHLLIB and MQCHLTAB, to specify the location and name of the file containing the client channel definition table. The WebSphere MQ client uses the values of these environment variables to access the client channel definition table. See “MQCHLLIB” on page 144 and “MQCHLTAB” on page 145 for more information.

For example, you can set the environment variables on a UNIX system by typing:

```
export MQCHLLIB=/mqmtop/qmgrs/QUEUEMANAGERNAME/@ipcc
export MQCHLTAB=AMQCLCHL.TAB
```

As an alternative to using environment variables MQCHLLIB and MQCHLTAB on Windows systems with Active Directory, you can use the **setmqscp** control command to publish the client-connection channel definitions in Active Directory. For information about this command and its syntax, see the *WebSphere MQ System Administration Guide*.

Note: If the MQSERVER environment variable is set, a WebSphere MQ client uses the client-connection channel definition specified by MQSERVER in preference to any definitions in the client channel definition table.

Client channel definition table

On platforms other than z/OS, the client-connection channel definition described previously is stored in the *client channel definition table* associated with the queue manager running on the server. The file containing the table is called AMQCLCHL.TAB and is a binary file that cannot be edited directly. You can use the DEFINE CHANNEL command to add an entry to the table, and the ALTER CHANNEL command to alter the attributes of a channel that already has an entry in the table.

Do not delete AMQCLCHL.TAB. It contains default channel definitions that are required when you define a channel. If you suspect that this has been deleted, for example you get error messages when you try to define a new channel, check to see that the file exists. If it has been deleted, define the default system objects again as described in “Setting up the server” on page 114.

If you install WebSphere MQ in the default location, AMQCLCHL.TAB is located in the following directory on a server machine:

- On Compaq NonStop Kernel:

\$SYSTEM.QMD

where QMD is the name of the data files subvolume of your queue manager.

This is the subvolume whose name has the suffix D. On MQSeries for Compaq NonStop Kernel, the client channel definition file is called CCHDEFS.

- On Compaq OpenVMS Alpha:

mqm_root:[mqm.qmgrs.QM.\$IPCC]

- On OS/2 Warp:

\mqm\qmgrs\QUEUEMANAGERNAME\@ipcc

- On OS/400, in the Integrated File System (IFS):

/QIBM/UserData/mqm/qmgrs/QUEUEMANAGERNAME/@ipcc

- On UNIX systems:

/mqmtop/qmgrs/QUEUEMANAGERNAME/@ipcc

Note that the name of the directory referred to by *QUEUEMANAGERNAME* is case sensitive on UNIX systems.

- On Windows:

\Program Files\IBM\WebSphere MQ\qmgrs*QUEUEMANAGERNAME*\@ipcc

On z/OS, client-connection channel definitions are stored with WebSphere MQ objects in page set zero. They are not stored in a file that can be accessed from a client machine, nor can they be copied directly to a client machine. Instead, you must use the MAKECLNT parameter of the COMMAND function of the WebSphere MQ utility program, CSQUTIL, to generate a file that contains a client channel definition table. You can then download this file to a client machine using a file transfer program. For details, see the *WebSphere MQ for z/OS System Administration Guide*.

Here is some sample JCL for making a client channel definition file:

```
//CLIENT EXEC PGM=CSQUTIL,PARM='QM2'
//STEPLIB DD DISP=SHR,DSN=thlqual.SCSQANLE
// DD DISP=SHR,DSN=thlqual.SCSQAUTH
//OUTCLNT DD DISP=OLD,DSN=MY.CLIENTS
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
COMMAND DDNAME(CMDCHL) MAKECLNT(OUTCLNT) CCSID(437)
/*
//CMDCHL DD *
DISPLAY CHANNEL(*) ALL TYPE(CLNTCONN)
DISPLAY AUTHINFO (*) ALL
/*
```

where thlqual is a high level qualifier for the WebSphere MQ library data sets. The data set for the client channel definition file, identified by the DD name OUTCLNT in the sample JCL, must have the format:

RECFM=U, LRECL=6144, BLKSIZE=2048

If you use FTP to copy the file, remember to type bin to set binary mode; do not use the default ASCII mode.

Notes:

1. On Compaq NonStop Kernel, a conversion utility (CNVCLCHL) is provided to convert the client channel definition table from a Compaq NonStop Kernel structured file to an unstructured one. See the *MQSeries for Compaq NonStop Kernel System Administration* for more details.
2. If you are connecting to an MQSeries client on a VM/ESA system, there is a limit to the number of client-connection channels you can define. The maximum number that can be held in the client channel definition table on your server for a VM/ESA client is 18.

Migrating to a later release level of WebSphere MQ

In general, the internal format of the client channel definition table might change from one release level of WebSphere MQ to the next. Because of this, a WebSphere MQ client can use a client channel definition table only when it has been prepared by a server queue manager that is at the same release level as the client, or at an earlier release level.

Using channels

For example, a Version 5.3 WebSphere MQ client can use a client channel definition table that has been prepared by a Version 2 queue manager. But a Version 2 MQSeries client cannot use a client channel definition table that has been prepared by a Version 5.3 queue manager.

Channel exits

The channel exits available to the WebSphere MQ client environment on OS/2 Warp, UNIX systems, Windows, Windows 3.1, and Windows 95 are:

- Send exit
- Receive exit
- Security exit

Channel exits are not available on DOS systems.

These exits are available at both the client and the server end of the channel. Exits are not available to your application if you are using the MQSERVER environment variable. Exits are explained in the *WebSphere MQ Intercommunication* manual.

The send and receive exit work together. There are several possible ways in which you can use them:

- Splitting and reassembling a message
- Compressing and decompressing data in a message
- Encrypting and decrypting user data
- Journaling each message sent and received

You can use the security exit to ensure that the WebSphere MQ client and server machines are correctly identified, as well as to control access to each machine.

Path to exits

This applies to WebSphere MQ clients on AIX, HP-UX, Linux, OS/2 Warp, Solaris, Windows 95, and Windows 98.

An `mqs.ini` file is added to your system during installation of the WebSphere MQ client. A default path for location of the channel exits on the client is defined in this file, using the stanza:

```
ClientExitPath:  
ExitsDefaultPath=<defaultprefix>/exits
```

Where `<defaultprefix>` is the value defined for your system in the `DefaultPrefix` stanza of the `mqs.ini` file.

On Windows 2000, Windows NT, and Windows XP, this stanza is added to the Registry.

When a channel is initialized, after an **MQCONN** or **MQCONNX** call, the `mqs.ini` file, or Registry, is searched. The `ClientExitPath` stanza is read and any channel exits that are specified in the channel definition are loaded.

Connecting a client to a queue-sharing group

You can connect a client to a queue-sharing group by creating an MQI channel between a client and a queue manager on a server machine that is a member of a queue-sharing group.

A queue-sharing group is formed by a set of queue-managers that can access the same set of shared queues. For more information on shared queues, see the *WebSphere MQ for z/OS Concepts and Planning Guide* book and the *WebSphere MQ Intercommunication* book.

A client putting to a shared queue can connect to any member of the queue-sharing group. The benefits of connecting to a queue-sharing group are possible increases in front-end and back-end availability, and increased capacity. You can connect a client to a queue-sharing group in the following ways.

Connecting to a specific queue manager

Connecting directly to a queue manager in a queue-sharing group gives the benefit that you can put messages to a shared target queue, which increases back-end availability.

Connecting to the generic interface

Connecting to the generic interface of a queue-sharing group will open a session with one of the queue managers in the group. The generic interface can be a WLM/DNS group name or a VTAM generic resource name, or another common interface to the queue-sharing group. See *WebSphere MQ Intercommunication* for more details on setting up a generic interface.

Connecting to the generic interface increases front-end availability, because the client queue manager can connect with any queue-manager in the group. You connect to the group using the generic interface when you do not want to connect to a specific queue manager within the queue-sharing group.

Creating channel definitions

To connect to the generic interface of a queue-sharing group you need to create channel definitions that can be accessed by any queue manager in the group. To do this you need to have the same definitions on each queue manager in the group.

Define the SVRCONN channel as follows:

```
DEFINE CHANNEL(CHAN1) CHLTYPE(SVRCONN) TRPTYPE(TCP) +
MCAUSER(' ') QSGDISP(GROUP)
```

Channel definitions on the server are stored in a shared DB2[®] repository. Each queue manager in the queue-sharing group makes a local copy of the definition, ensuring that you will always connect to the correct server-connection channel when you issue an MQCONN or MQCONNEX call.

Define the CLNTCONN channel as follows:

```
DEFINE CHANNEL(CHAN1) CHLTYPE(CLNTCONN) TRPTYPE(TCP) +
CONNAME(WLM/DNS groupname) QMNAME(QSG1) +
DESCR('Client-connection to Queue Sharing Group QSG1') QSGDISP(GROUP)
```

Because the generic interface of the queue-sharing group is stored in the CONNAME field in the client-connection channel, you can now connect to any queue manager in the group, and put to shared queues owned by that group.

Stopping channels

In WebSphere MQ Version 5.3, when you issue a STOP CHANNEL command against a server-connection channel, you can choose what method to use to stop the client-connection channel.

Using channels

This means that a client channel issuing an MQGET wait call can be controlled, and you can decide how and when to stop the channel.

The STOP CHANNEL command can be issued with three modes, indicating how the channel is to be stopped:

Quiesce

Stops the channel after any current message have been processed.

Force Stops the channel immediately.

Terminate

Stops the channel immediately, and can terminate the server-connection channel's process or thread.

If mode terminate is used, for example, an attempt is made to stop the server-connection channel, first with mode quiesce, then with mode force, and if necessary with mode terminate. This is a multi-stage process. The client can receive different return codes during the different stages of termination. see Table 9 for details of the return codes. If the process or thread is terminated, the client receives a communication error.

The return codes returned to the application vary according to the MQI call issued, and the STOP CHANNEL command issued. The state of the channel after the command is issued depends on the state of the fail if quiescing parameter in the API call (MQGMO_FAIL_IF_QUIESCING, MQPMO_FAIL_IF_QUIESCING, MQOO_FAIL_IF_QUIESCING, as appropriate). The return codes are listed in the table below, together with the client channel status after the command is issued:

Table 9. Stop Channel: Stop Modes and Reason Codes issued

Stop Mode	Fail if quiescing option	MQI Call CompCode	MQI call Reason Code	Client Channel Running
QUIESCE	no	Value returned from the API call		✓
QUIESCE	yes	MQCC_FAILED	MQRC_CONNECTION_QUIESCING	✓
FORCE	no	MQCC_FAILED	MQRC_CONNECTION_BROKEN	
FORCE	yes	MQCC_FAILED	MQRC_CONNECTION_QUIESCING (2)	✓
TERMINATE	no	MQCC_FAILED	MQRC_CONNECTION_BROKEN	
TERMINATE	yes	MQCC_FAILED	MQRC_CONNECTION_BROKEN	
Notes: 1. Only the state of the server channel is altered if FAIL_IF_QUIESCING is not specified. The Client Channel is not ended forcefully. 2. The channel operates for a few seconds in this state. The communications session is then ended with reason code MQRC_CONNECTION_BROKEN.				

Chapter 9. The Secure Sockets Layer (SSL) on WebSphere MQ clients

This chapter discusses the following aspects of using the Secure Sockets Layer (SSL) on WebSphere MQ client systems:

- “Specifying that an MQI channel uses SSL”
- “Specifying the location of LDAP servers that hold certificate revocation lists (CRLs)”

For more information about how to implement the SSL support within WebSphere MQ, see *WebSphere MQ Security*.

Specifying that an MQI channel uses SSL

For an MQI channel to use SSL, the value of the *SSLCipherSpec* attribute of the client-connection channel must be the name of a CipherSpec that is supported by WebSphere MQ on the client platform. You can define a client-connection channel with a value for this attribute in the following ways. They are listed in order of decreasing precedence.

1. When a WebSphere MQ client application issues an MQCONN call.
The application can specify the name of a CipherSpec in the *SSLCipherSpec* field of a channel definition structure, MQCD. This structure is referenced by the connect options structure, MQCNO, which is a parameter on the MQCONN call.
2. Using a client channel definition table.
One or more entries in a client channel definition table can specify the name of a CipherSpec. For example, if you create an entry by using the DEFINE CHANNEL MQSC command, you can use the SSLCIPH parameter on the command to specify the name of a CipherSpec.
3. Using Active Directory on Windows.
On Windows systems with Active Directory, you can use the **setmqscp** control command to publish the client-connection channel definitions in Active Directory. One or more of these definitions can specify the name of a CipherSpec.

For example, if a client application provides a client-connection channel definition in an MQCD structure on an MQCONN call, this definition is used in preference to any entries in a client channel definition table that can be accessed by the WebSphere MQ client.

Note that you cannot use the MQSERVER environment variable to provide the channel definition at the client end of an MQI channel that uses SSL.

Specifying the location of LDAP servers that hold certificate revocation lists (CRLs)

On a WebSphere MQ client system, you can specify the location of Lightweight Directory Access Protocol (LDAP) servers that hold certificate revocation lists (CRLs) in the following ways. They are listed in order of decreasing precedence.

The Secure Sockets Layer on WebSphere MQ clients

1. "When a WebSphere MQ client application issues an MQCONN call"
2. "Using a client channel definition table"
3. "Using Active Directory on Windows" on page 141

See the relevant sections for more information about each of these ways.

The intention is that each LDAP server holds the same CRLs. The reason for configuring more than one LDAP server with CRLs is to provide higher availability. If one LDAP server is not available when it is required, a WebSphere MQ client can attempt to access another.

When a WebSphere MQ client application issues an MQCONN call

On an MQCONN call, the connect options structure, MQCNO, can reference an SSL configuration options structure, MQSCO. In turn, the MQSCO structure can reference one or more authentication information record structures, MQAIR. Each MQAIR structure contains all the information a WebSphere MQ client needs to access an LDAP server that holds CRLs. For example, one of the fields in an MQAIR structure is the host address or IP address of a system on which an LDAP server runs. This address can be followed by an optional port number enclosed in parentheses. The default port number is 389.

For more information about the MQAIR structure, see the *WebSphere MQ Application Programming Reference*.

Using a client channel definition table

On a server queue manager, you can create one or more authentication information objects. The attributes of an authentication object contain all the information that is needed to access an LDAP server that holds CRLs. One of the attributes specifies the host address or IP address of a system on which an LDAP server runs. This address can be followed by an optional port number enclosed in parentheses. The default port number is 389.

To enable a WebSphere MQ client to access LDAP servers that hold CRLs, the attributes of one or more authentication information objects can be included in a client channel definition table. This is done in the following ways:

On the server platforms AIX, HP-UX, Linux, OS/400, Solaris, and Windows

You can create a namelist that contains the names of one or more authentication information objects. You can then set the queue manager attribute, *SSLCRLNameList*, to the name of this namelist. By doing this, you enable the WebSphere MQ SSL support for the queue manager to access the LDAP servers that hold CRLs.

The attributes of the authentication information objects identified by the namelist are referred to collectively here as the *CRL information*. When you set the queue manager attribute, *SSLCRLNameList*, to the name of the namelist, the CRL information is copied into the client channel definition table associated with the queue manager. If the client channel definition table can be accessed from a client system as a shared file, or if the client channel definition table is then copied to a client system, the WebSphere MQ client on that system can use the CRL information in the client channel definition table to access LDAP servers that hold CRLs.

The Secure Sockets Layer on WebSphere MQ clients

If the CRL information of the queue manager is changed subsequently, the change is reflected in the client channel definition table associated with the queue manager. If the queue manager attribute, *SSLCRLNameList*, is set to blank, all the CRL information is removed from the client channel definition table. These changes are not reflected in any copy of the table on a client system.

If you require the CRL information at the client and server ends of an MQI channel to be different, and the server queue manager is the one that is used to create the CRL information, you can do the following:

1. On the server queue manager, create the CRL information for use on the client system.
2. Copy the client channel definition table containing the CRL information to the client system.
3. On the server queue manager, change the CRL information to what is required at the server end of the MQI channel.

On the server platform z/OS

On z/OS, a client channel definition table is generated by the MAKECLNT parameter of the COMMAND function of the WebSphere MQ utility program, CSQUTIL. The DISPLAY CHANNEL commands in the input data set determine which client-connection channel definitions are included in the table. Likewise, the DISPLAY AUTHINFO commands in the input data set determine which authentication information objects are used to form the CRL information in the table.

The contents of a client channel definition table generated on z/OS do not depend on the value of any queue manager attributes, such as *SSLCRLNameList*, and cannot be updated dynamically. The only way you can change the CRL information in a client channel definition table is to generate a new table by running CSQUTIL again.

Using Active Directory on Windows

On Windows systems with Active Directory, you can use the **setmqcrl** control command to publish the current CRL information in Active Directory. For information about this command and its syntax, see the *WebSphere MQ System Administration Guide*.

Chapter 10. Using WebSphere MQ environment variables

This chapter describes the environment variables that you can use with WebSphere MQ client applications.

WebSphere MQ uses default values for those variables that you have not set. Using environment variables, you can update your system profile to make a permanent change, issue the command from the command line to make a change for this session only, or, if you want one or more variables to have a particular value dependent on the application that is running, add commands to a command script file used by the application.

The WebSphere MQ environment variables are:

- “MQCCSID” on page 144
- “MQCHLLIB” on page 144
- “MQCHLTAB” on page 145
- “MQDATA (DOS and Windows 3.1 only)” on page 145
- “MQNAME” on page 146
- “MQ_PASSWORD (DOS, OS/2 Warp, Windows 3.1, Windows 95, and Windows 98 only)” on page 146
- “MQSERVER” on page 146
- “MQTRACE (DOS, Windows 3.1, and VM/ESA only)” on page 148
- “MQ_USER_ID (DOS, OS/2 Warp, Windows 3.1, Windows 95, and Windows 98 only)” on page 148
- “MQSWORKPATH (OS/2 Warp only)” on page 148

Commands are available on all the WebSphere MQ client platforms unless otherwise stated.

Notes:

1. WebSphere MQ for z/OS does not support any WebSphere MQ environment variables. If you are using this platform as your server, see “Client channel definition table” on page 134 for information about how the client channel definition table is generated on z/OS. You can still use the WebSphere MQ environment variables on your client platform.
2. MQSeries for Compaq NonStop Kernel does not support any WebSphere MQ environment variables. MQSeries for Compaq NonStop Kernel does recognize TACL environment variables, or PARAMS. See the *MQSeries for Compaq NonStop Kernel System Administration* for details. You can still use the WebSphere MQ environment variables on your client platform.
3. MQSeries for VSE/ESA does not support any WebSphere MQ environment variables. You can still use the WebSphere MQ environment variables on your client platform.

For each environment variable, use the command relevant to your platform to display the current setting or to reset the value of a variable.

Environment variables

For example:

Command	Effect
SET MQSERVER=	Removes the variable from DOS, OS/2 Warp, Windows, Windows 3.1, and Windows 95 environments
unset MQSERVER	Removes the variable from UNIX systems environments
SET MQSERVER	Displays the current setting on DOS, OS/2 Warp, Windows, Windows 3.1, and Windows 95
echo \$MQSERVER	Displays the current setting on UNIX systems
set	Displays all environment variables for the session

MQCCSID

This specifies the coded character set number to be used and overrides the machine's configured CCSID. See "Choosing client or server coded character set identifier (CCSID)" on page 151 for more information.

The format of this command is:

- For OS/2 Warp and Windows:
SET MQCCSID=number
- For UNIX systems:
export MQCCSID=number
- For VM/ESA:
GLOBALV SELECT CENV SETLP MQCCSID number

For more information, see the *WebSphere MQ Application Programming Reference* manual.

MQCHLLIB

This specifies the directory path to the file containing the client channel definition table. The file is created on the server, but can be copied across to the WebSphere MQ client machine. If MQCHLLIB is not set, the path defaults to:

- For DOS, OS/2 Warp, Windows, Windows 3.1, and Windows 95:
Rootdrive:\mqm\
- For UNIX systems:
/var/mqm/

If you are using WebSphere MQ for z/OS as your server, the file must be kept on the WebSphere MQ client machine.

For servers on other platforms, consider keeping this file on the server to make administration easier.

The format of this command is:

- For DOS, OS/2 Warp, Windows, Windows 3.1, and Windows 95:
SET MQCHLLIB=pathname
- For UNIX systems:
export MQCHLLIB=pathname

For example:

```
SET MQCHLLIB=C:\os2
```

Notes:

1. If you change the default setting of this variable *after* you create the queue manager, you must copy your existing client channel definition table to the new location.
2. If you change the default setting of this variable *before* you create the queue manager, you do not need to copy your existing client channel definition table to the new location.

MQCHLTAB

This specifies the name of the file containing the client channel definition table. The default file name is AMQCLCHL.TAB. For information about where the client channel definition table is located on a server machine, see “Client channel definition table” on page 134.

The format of this command is:

- On DOS, OS/2 Warp, Windows, Windows 3.1, and Windows 95:
SET MQCHLTAB=filename
- On UNIX systems:
export MQCHLTAB=filename
- On VM/ESA:
GLOBALV SELECT CENV SETLP MQCHLTAB filename

For example:

```
SET MQCHLTAB=ccdf1.tab
```

MQDATA (DOS and Windows 3.1 only)

This holds the path to the directory containing the trace, error and **qm.ini** files. (The **qm.ini** file is needed for setting up NetBIOS.) The default is to root directory of the C drive.

The format of this command is:

```
SET MQDATA=pathname
```

The trace and error files are:

AMQERR01.FDC

For First Failure Data Capture messages.

AMQERR01.LOG

For error messages.

An error message is always added to the end of the log, so the files must be deleted periodically to avoid the files getting too large. If the file does not exist at the time a record needs to be added to one of these files it will be created.

These files are written in binary format. Use the **RUNMQFMT** command supplied with WebSphere MQ to reformat these files into a readable form.

MQNAME

This specifies the local NetBIOS name that the WebSphere MQ processes can use. See “Defining a NetBIOS connection” on page 106 for a full description and for the rules of precedence on the client and the server.

The format of this command is:

```
SET MQNAME=Your_env_Name
```

For example:

```
SET MQNAME=CLIENT1
```

The NetBIOS on some platforms requires a different name (set by MQNAME) for each application if you are running multiple WebSphere MQ applications simultaneously on the WebSphere MQ client.

MQ_PASSWORD (DOS, OS/2 Warp, Windows 3.1, Windows 95, and Windows 98 only)

This specifies the password of the client. It is described in Chapter 7, “Setting up WebSphere MQ client security” on page 123.

MQSERVER

This is used to define a minimal channel. It cannot be used to define an SSL channel, or a channel with channel exits. It specifies the location of the WebSphere MQ server and the communication method to be used. Note that *ConnectionName* must be a fully-qualified network name. The *ChannelName* cannot contain the forward slash (/) character because it is used to separate the channel name, transport type, and connection name. When the MQSERVER environment variable is used to define a client channel a maximum message length (MAXMSGL) of 4 MB is used, so larger messages cannot flow across this channel. For larger messages a client-connection channel must be defined using DEFINE CHANNEL, on the server, with MAXMSGL set to a larger figure.

The format of this command is:

- For DOS, OS/2 Warp, Windows, Windows 3.1, and Windows 95:

```
SET MQSERVER=ChannelName/TransportType/ConnectionName
```

- For OS/2 Warp using LU 6.2:

```
SET MQSERVER=ChannelName/LU62/ConnectionName/ModeName/TpName
```

- For UNIX systems:

```
export MQSERVER=ChannelName/TransportType/ConnectionName
```

- For Compaq OpenVMS Alpha using DECnet:

```
define mqserver "ChannelName/decnet/nodename(object number)"
```

Or a symbol:

```
mqserver := "ChannelName/decnet/nodename(object number)"
```

- For VM/ESA using TCP/IP:

```
GLOBALV SELECT CENV SETLP MQSERVER ChannelName/TCP/ConnectionName
```

- For VM/ESA using SNA LU 6.2:

```
GLOBALV SELECT CENV SETLP MQSERVER ChannelName/LU62/ModeName/TpName
```


TCP/IP default port

By default, for TCP/IP, WebSphere MQ assumes that the channel will be connected to port 1414. You can change this by:

- Adding the port number in brackets as the last part of the ConnectionName:
 - For DOS, OS/2 Warp, Windows, Windows 3.1, and Windows 95:
`SET MQSERVER=ChannelName/TransportType/ConnectionName(PortNumber)`
 - For UNIX systems:
`export MQSERVER=ChannelName/TransportType/ConnectionName(PortNumber)`
- Changing the `qm.ini` file by adding the port number to the protocol name, for example:
 TCP:
`port=2001`
- Adding WebSphere MQ to the services file as described in “Defining a TCP/IP connection” on page 93.

SPX default socket

By default, for SPX, WebSphere MQ assumes that the channel will be connected to socket 5E86. You can change this by:

- Adding the socket number in brackets as the last part of the ConnectionName:
`SET MQSERVER=ChannelName/TransportType/ConnectionName(SocketNumber)`

For SPX connections, specify the ConnectionName and socket in the form `network.node(socket)`. If the WebSphere MQ client and server are on the same network, the network need not be specified. If you are using the default socket, the socket need not be specified.

- Changing the `qm.ini` file by adding the port number to the protocol name, for example:
 SPX:
`socket=5E87`

Examples of using MQSERVER

Examples on OS/2 Warp:

```
SET MQSERVER=CHAN1/TCP/9.20.4.56(2001)
SET MQSERVER=CHAN1/NETBIOS/BOX643
SET MQSERVER=CHAN1/SPX/000001.08005A7161E5(5E88)
```

Examples on a UNIX system:

```
export MQSERVER=CHAN1/TCP/'9.20.4.56(2002) '
export MQSERVER=CHAN1/LU62/BOX99
```

Examples on Compaq OpenVMS Alpha:

```
define mqserver "chan1/DECNET/node(task)"
mqserver="chan1/TCP/9.20.4.2(2001)"
```

All **MQCONN** or **MQCONN**X requests then attempt to use the channel you have defined. However, if an **MQCD** structure has been defined in the **MQCNO** structure supplied to **MQCONN**X, the channel specified by the **MQCD** structure takes priority over any specified by the **MQSERVER** environment variable.

The **MQSERVER** environment variable takes priority over any client channel definition pointed to by **MQCHLLIB** and **MQCHLTAB**.

Environment variables

Canceling MQSERVER

To cancel MQSERVER and return to the client channel definition table pointed to by MQCHLLIB and MQCHLTAB, enter the following:

- On DOS, OS/2 Warp, Windows, Windows 3.1, and Windows 95:
SET MQSERVER=
- On UNIX systems:
unset MQSERVER

MQTRACE (DOS, Windows 3.1, and VM/ESA only)

This sets tracing on and off, as required. The default is for tracing to be turned off.

The format of this command is:

- For DOS and Windows 3.1:
SET MQTRACE=filename,options
- For VM/ESA:
GLOBALV SELECT CENV SETLP MQTRACE filename/options

For example, to direct the communication flow trace entries to *MQ.TRC* file and overwrite the previous trace file each time the program runs:

```
SET MQTRACE=MQ.TRC,cw
```

MQ_USER_ID (DOS, OS/2 Warp, Windows 3.1, Windows 95, and Windows 98 only)

This specifies the user ID of the client. It is described in Chapter 7, “Setting up WebSphere MQ client security” on page 123.

MQSWORKPATH (OS/2 Warp only)

This specifies the path to the *mqs.ini* file and is used internally by WebSphere MQ.

Part 4. Application programming

Chapter 11. Using the message queue interface

(MQI)	151
Limiting the size of a message.	151
Choosing client or server coded character set identifier (CCSID)	151
CCSID and encoding fields - multiple puts	152
Controlling application in a Windows 3.1 environment.	152
Designing applications	152
Windows 3.1 environment	152
Using MQINQ	152
Using syncpoint coordination	153
Compaq NonStop Kernel server	153
Using MQCONN	153
Shared connection handles on MQCONN	154

Chapter 12. Building applications for

WebSphere MQ clients	155
Running applications in the WebSphere MQ client environment.	156
Triggering in the client environment.	156
Process definition	157
MQSeries client for Windows 95	157
Trigger monitor	157
CICS applications (non-z/OS)	158
Linking C applications with the WebSphere MQ client code	158
Running 16-bit and 32-bit Windows clients	160
Linking C++ applications with the WebSphere MQ client code	160
Linking COBOL applications with the WebSphere MQ client code.	160
Linking PL/I applications with the WebSphere MQ client code	161
Linking Visual Basic applications with the WebSphere MQ client code.	162

Chapter 13. Running applications on

WebSphere MQ clients	163
Using environment variables	164
Using MQSERVER	164
Using MQCHLLIB and MQCHLTAB	164
Using the MQCNO structure	164
Using DEFINE CHANNEL	164
Role of the client channel definition table	165
Multiple queue managers	165
Queue-sharing groups	165
Examples of MQCONN calls	165
What the examples demonstrate	167
Example 1. Queue manager name includes an asterisk (*)	167
Example 2. Queue manager name specified	168
Example 3. Queue manager name is blank or an asterisk (*)	168

Chapter 14. Solving problems

WebSphere MQ client fails to make a connection	171
Stopping WebSphere MQ clients	172
Error messages with WebSphere MQ clients	172
Compaq OpenVMS Alpha, OS/2 Warp, UNIX systems, Windows, and Windows 95	172
DOS and Windows 3.1 clients	172
How to read the error log and FFDCs for DOS and Windows 3.1	172
Using trace on DOS and Windows 3.1	173
Example DOS trace data.	173
Using trace on OS/2 Warp, Windows, and Windows 95.	174
File names for trace files.	175
How to examine First Failure Support Technology™ (FFST) files	175
Using trace on AIX and AT&T GIS UNIX	175
Using trace on Compaq OpenVMS Alpha, HP-UX, SINIX, DC/OSx, and Solaris	176
File names for trace files.	177
How to examine FFSTs	177
Using trace on VM/ESA	177
Example VM/ESA trace data	177

Chapter 11. Using the message queue interface (MQI)

When you write your WebSphere MQ application, you need to be aware of the differences between running it in an WebSphere MQ client environment and running it in the full WebSphere MQ queue manager environment.

This chapter discusses the following topics:

- “Limiting the size of a message”
- “Choosing client or server coded character set identifier (CCSID)”
- “Controlling application in a Windows 3.1 environment” on page 152
- “Designing applications” on page 152
- “Using MQINQ” on page 152
- “Using syncpoint coordination” on page 153
- “Using MQCONN” on page 153

Limiting the size of a message

The maximum message length (MaxMsgLength) attribute of a queue manager is the maximum length of a message that can be handled by that queue manager. The default maximum message length supported depends on the platform you are using.

On WebSphere MQ products, you can increase the maximum message length attribute of a queue manager. Details are given in the *WebSphere MQ Application Programming Guide*.

You can find out the value of MaxMsgLength for a queue manager by using the MQINQ call.

If the MaxMsgLength attribute is changed, no check is made that there are not already queues, and even messages, with a length greater than the new value. After a change to this attribute, applications and channels should be restarted in order to ensure that the change has taken effect. It will then not be possible for any new messages to be generated that exceed either the queue manager's MaxMsgLength or the queue's MaxMsgLength (unless queue manager segmentation is allowed).

The maximum message length in a channel definition limits the size of a message that you can transmit along a client connection. If a WebSphere MQ application tries to use the MQPUT call or the MQGET call with a message larger than this, an error code is returned to the application.

Choosing client or server coded character set identifier (CCSID)

The data passed across the MQI from the application to the client stub should be in the local coded character set identifier (CCSID), encoded for the WebSphere MQ client. If the connected queue manager requires the data to be converted, this is done by the client support code.

The client code assumes that the character data crossing the MQI in the client is in the CCSID configured for that machine. If this CCSID is an unsupported CCSID or is not the required CCSID, it can be overridden with the MQCCSID environment variable, for example:

Using the MQI

SET MQCCSID=850

Or, on UNIX systems: export MQCCSID=850

Set this in the profile and all MQI data will be assumed to be in code page 850.

Note: This does not apply to application data in the message.

CCSID and encoding fields - multiple puts

If your application is performing multiple PUTs that include WebSphere MQ headers after the message descriptor (MQMD), be aware that the CCSID and encoding fields of the MQMD are overwritten after completion of the first PUT. After the first PUT, these fields contain the value used by the connected queue manager to convert the WebSphere MQ headers. Ensure that your application resets the values to those it requires.

Controlling application in a Windows 3.1 environment

An MQSeries for Windows 3.1 client runs within a full Windows environment, not under a DOS prompt.

Normally, when you issue a request from a non-WebSphere MQ application, control is not returned to that application until the request is fulfilled. This is because the Windows 3.1 environment is a cooperative multi-tasking system.

However, the WebSphere MQ client code overrides the locking of the machine and the application to enable you to start up more applications, or work on something else until the MQI call has been answered. Should the application attempt to issue a further MQI call before the previous one has been answered, the application receives a return code indicating that there is still a call in progress, and the second call will fail.

Designing applications

When designing an application, consider what controls you need to impose during an MQI call to ensure that the WebSphere MQ application processing is not disrupted.

Windows 3.1 environment

To cooperate fully in the Windows 3.1 multi-tasking environment, an MQI call results in the client code executing a GetMessage loop on behalf of the application. If an application has accelerator keys defined, these will not function until the MQI call returns and control is returned to the GetMessage loop of the application.

Note: The only way that an ongoing MQI call can be cancelled is by the application receiving a WM_QUIT message.

Using MQINQ

Some values queried using MQINQ are modified by the client code.

CCSID

is set to the client CCSID, not that of the queue manager.

MaxMsgLength

is reduced if it is restricted by the channel definition. This will be the lower of:

- The value defined in the queue definition, or
- The value defined in the channel definition

For more information, see the *WebSphere MQ Application Programming Guide*.

Using syncpoint coordination

Within WebSphere MQ, one of the roles of the queue manager is syncpoint control within an application. If an application runs on an WebSphere MQ client, it can issue MQCMIT and MQBACK, but the scope of the syncpoint control is limited to the MQI resources.

Applications running in the full queue manager environment on the server can coordinate multiple resources (for example databases) via a transaction monitor. On the server you can use the Transaction Monitor supplied with the Version 5.1 and above MQSeries products, or another transaction monitor such as CICS. You cannot use a transaction monitor with a client application. The WebSphere MQ verb MQBEGIN is not valid in a client environment.

Compaq NonStop Kernel server

When a WebSphere MQ client connects to a queue manager on Compaq NonStop Kernel:

- Any MQGET, MQPUT, or MQPUT1 with an MQ*_SYNCPOINT option initiates a Compaq NonStop Kernel transaction, if one has not already been associated with the connection handle.
- Any MQGET, MQPUT, or MQPUT1 with neither an MQ*_SYNCPOINT nor an MQ*_NO_SYNCPOINT option initiates a Compaq NonStop Kernel transaction, if one has not already been associated with the connection handle.
- The MQCMIT call commits a Compaq NonStop Kernel transaction, if one is associated with the connection handle. The MQBACK call cancels the Compaq NonStop Kernel transaction, if one is associated with the connection handle.

In all cases, if the Compaq NonStop Kernel BEGINTRANSACTION fails, a *CompCode* of MQCC_FAILED, and a *Reason* of MQRC_SYNCPOINT_NOT_AVAILABLE are returned to the caller.

Using MQCONN

You can use the MQCONN call to specify a channel definition (MQCD) structure in the MQCNO structure. This allows the calling client application to specify the definition of the client-connection channel at run-time. For more information, see “Using the MQCNO structure on an MQCONN call” on page 132. When you use MQCONN, the call issued at the server depends on the server level and listener configuration.

When you use MQCONN from a client, the following options are ignored:

- MQCNO_STANDARD_BINDING
- MQCNO_FASTPATH_BINDING

The MQCD structure you can use depends on the MQCD version number you are using. For information on MQCD versions (MQCD_VERSION), see *WebSphere MQ*

Using the MQI

Intercommunication. You can use the MQCD structure, for instance, to pass channel-exit programs to the server. If you are using MQCD Version 3 or later, you can use the structure to pass an array of exits to the server. You can use this function to perform more than one operation on the same message, such as encryption and compression, by adding an exit for each operation, rather than modifying an existing exit. If you do not specify an array in the MQCD structure, the single exit fields will be checked. For more information on channel-exit programs, see *WebSphere MQ Intercommunication*.

Shared connection handles on MQCONN

On AIX, HP-UX, Linux, OS/400, Solaris, and Windows, you can share handles between different threads within the same process, using shared connection handles. When you specify a shared connection handle, the connection handle returned from the MQCONN call can be passed in subsequent MQI calls on any thread in the process.

Note: You can use a shared connection handle on a WebSphere MQ client to connect to a server queue manager that does not support shared connection handles.

For more information, see the *WebSphere MQ Application Programming Reference*.

Chapter 12. Building applications for WebSphere MQ clients

This chapter lists points to consider when running an application in an WebSphere MQ client environment, and describes how to link your application code with the WebSphere MQ client code.

It discusses the following topics:

- “Running applications in the WebSphere MQ client environment” on page 156
- “Triggering in the client environment” on page 156
- “Linking C applications with the WebSphere MQ client code” on page 158
- “Linking C++ applications with the WebSphere MQ client code” on page 160
- “Linking COBOL applications with the WebSphere MQ client code” on page 160
- “Linking PL/I applications with the WebSphere MQ client code” on page 161
- “Linking Visual Basic applications with the WebSphere MQ client code” on page 162

If an application is to run in a client environment, you can write it in the languages shown in the following table:

Table 10. Programming languages supported in client environments

Client platform	Assembler	C	C++ ¹	COBOL	Java ²	PL/I	REXX	RPG	Visual Basic
AIX		✓	✓	✓	✓				
AT&T GIS UNIX		✓							
Compaq Tru64 UNIX		✓	✓	✓	✓				
Compaq OpenVMS Alpha		✓		✓					
DOS		✓							
HP-UX		✓	✓	✓	✓				
Linux		✓	✓		✓				
OS/2 Warp		✓	✓	✓	✓	✓			
OS/400					✓				
SINIX and DC/OSx		✓		✓					
Solaris		✓	✓	✓	✓				
Sun Solaris, Intel Platform Edition		✓	✓	✓	✓				
VM/ESA	✓	✓		✓		✓	✓		
Windows 3.1		✓	✓	✓					✓
Windows 95		✓	✓	✓					✓
Windows 98		✓	✓	✓					✓
Windows 2000		✓	✓	✓	✓	✓			✓
Windows NT		✓	✓	✓	✓	✓			✓
Windows XP		✓	✓	✓	✓	✓			✓

Building applications

Table 10. Programming languages supported in client environments (continued)

Client platform	Assembler	C	C++ ¹	COBOL	Java ²	PL/I	REXX	RPG	Visual Basic
Note: <ol style="list-style-type: none">1. If you are using an WebSphere MQ client supplied with an WebSphere MQ Version 5 product, you can write applications to run on the client in C++. Programs that use the WebSphere MQ C++ classes can be used successfully with WebSphere MQ Version 5 and WebSphere MQ for z/OS servers only. To see how to link your C++ applications and for information about all aspects of using C++ see the <i>WebSphere MQ Using C++</i> manual.2. If you have installed the WebSphere MQ client for Java, you can write Java applets that communicate with WebSphere MQ. Remember to link your application to the relevant client library file. For information, see the <i>WebSphere MQ Using Java</i> manual.									

Running applications in the WebSphere MQ client environment

You can run a WebSphere MQ application both in a full WebSphere MQ environment and in a WebSphere MQ client environment without changing your code, provided that:

- It does not need to connect to more than one queue manager concurrently
- The queue manager name is not prefixed with an asterisk (*) on an **MQCONN** or **MQCONNX** call
- It does not need to use any of the exceptions listed in “What applications run on a WebSphere MQ client?” on page 5

Note: The libraries you use at link-edit time determine the environment in which your application must run.

When working in the WebSphere MQ client environment, remember that :

- Each application running in the WebSphere MQ client environment has its own connections to servers. It will have one connection to each server it requires, a connection being established with each **MQCONN** or **MQCONNX** call the application issues.
- An application sends and gets messages synchronously.
- All data conversion is done by the server, but see also “MQCCSID” on page 144.

Triggering in the client environment

Triggering is explained in detail in the *WebSphere MQ Application Programming Guide*.

Messages sent by WebSphere MQ applications running on WebSphere MQ clients contribute to triggering in exactly the same way as any other messages, and they can be used to trigger programs on the server. The trigger monitor and the application to be started must be on the same system.

The default characteristics of the triggered queue are the same as those in the server environment. In particular, if no MQPMO syncpoint control options are specified in a client application putting messages to a triggered queue that is local to a z/OS queue manager, the messages are put within a unit of work. If the triggering condition is then met, the trigger message is put on the initiation queue within the same unit of work and cannot be retrieved by the trigger monitor until the unit of work ends. The process that is to be triggered is not started until the unit of work ends.

Process definition

You must define the process definition on the server, because this is associated with the queue that has triggering set on.

The process object defines what is to be triggered. If the client and server are not running on the same platform, any processes started by the trigger monitor must define *ApplType*, otherwise the server takes its default definitions (that is, the type of application that is normally associated with the server machine) and causes a failure.

For example, if the trigger monitor is running on a Windows NT client and wants to send a request to an OS/2 Warp server, MQAT_WINDOWS_NT must be defined otherwise OS/2 Warp uses its default definitions (that is, MQAT_OS2) and the process fails.

For a list of application types, see the *WebSphere MQ Application Programming Reference* manual.

MQSeries client for Windows 95

The MQSeries client for Windows 95 runs in 32-bit mode. It is also possible to run the client for Windows 3.1 in 16-bit mode on a Windows 95 platform. If a trigger monitor is running on a Windows 95 client you must make sure that you define the correct *ApplType*:

MQAT_WINDOWS

Windows 3.1 client or 16-bit Windows application.

MQAT_WINDOWS_NT

Windows client or 32-bit Windows application.

Trigger monitor

The trigger monitor provided by non-z/OS WebSphere MQ products runs in the client environments for Compaq OpenVMS Alpha, OS/2 Warp, UNIX systems, Windows, Windows 3.1, and Windows 95. To run the trigger monitor, issue the command:

```
runmqtmc [-m QMgrName] [-q InitQ]
```

The default initiation queue is SYSTEM.DEFAULT.INITIATION.QUEUE on the default queue manager. This is where the trigger monitor looks for trigger messages. It then calls programs for the appropriate trigger messages. This trigger monitor supports the default application type and is the same as runmqtrm except that it links the client libraries.

The command string, built by the trigger monitor, is as follows:

1. The *ApplicId* from the relevant process definition. This is the name of the program to run, as it would be entered on the command line.
2. The MQTMC2 structure, enclosed in quotes, as got from the initiation queue. A command string is invoked that has this string, exactly as provided, in quotes in order that the system command will accept it as one parameter.
3. The *EnvrData* from the relevant process definition.

The trigger monitor does not look to see if there is another message on the initiation queue until the completion of the application it has just started. If the

Triggering

application has a lot of processing to do, this might mean that the trigger monitor cannot keep up with the number of trigger messages arriving. There are two ways to deal with this:

1. Have more trigger monitors running
If you choose to have more trigger monitors running, you can control the maximum number of applications that can run at any one time.
2. Run the started applications in the background
If you choose to run applications in the background, WebSphere MQ imposes no restriction on the number of applications that can run.

To run the started application in the background on an OS/2 Warp system, within the *ApplicId* field you must prefix the name of your application with a start command; for example, start amqsinq /B.

To run the started application in the background on a UNIX system, you must put an & (ampersand) at the end of the *EnvrData* of the process definition.

CICS applications (non-z/OS)

A non-z/OS CICS application program that issues an **MQCONN** or **MQCONNX** call must be defined to CEDA as RESIDENT. To make the resident code as small as possible, you can link to a separate program to issue the **MQCONN** or **MQCONNX** call.

If the **MQSERVER** environment variable is used to define the client connection, it must be specified in the **CICSENV.CMD** file.

WebSphere MQ applications can be run in an WebSphere MQ server environment or on an WebSphere MQ client without changing code. However, in an WebSphere MQ server environment, CICS can act as syncpoint coordinator, and you use **EXEC CICS SYNCPOINT** and **EXEC CICS SYNCPOINT ROLLBACK** rather than **MQCMIT** and **MQBACK**. If a CICS application is simply relinked as a client, syncpoint support is lost. **MQCMIT** and **MQBACK** must be used for the application running on an WebSphere MQ client.

Linking C applications with the WebSphere MQ client code

Having written your WebSphere MQ application that you want to run on the WebSphere MQ client, you must link it to a queue manager. You can do this in two ways:

1. Directly, in which case the queue manager must be on the same machine as your application
2. To a client library file, which gives you access to queue managers on the same or on a different machine

WebSphere MQ provides a client library file for each environment:

AIX libmqic.a library for non-threaded applications, or libmqic_r.a library for threaded applications.

AT&T GIS UNIX

libmqic.so and libmqmcs.so.

If you want to use the programs on a machine that has only the MQSeries client for AT&T GIS UNIX installed, you must recompile the programs to link them with the client library:

```
$ /bin/cc -o <prog> <prog>.c -lmqic -lmqmcs -lmqzmse -lnet \
-lns1 -lsocket -ldl -lc
```

Compaq OpenVMS Alpha

MQIC.EXE in SYS\$SHARE.

Compaq Tru64 UNIX

libmqic.so library for Version 4 non-threaded applications, libmqic_r.so for Version 4 threaded applications, or libmqic.so for Version 5 applications.

DOS MQIC.LIB.

Your application must also be linked with at least three of the following libraries, one for each protocol, indicating whether you do or do not require it.

MQICN

NetBIOS required

MQICDN

NetBIOS not required

MQICS

SPX required

MQICDS

SPX not required

MQICT

TCP/IP required

MQICDT

TCP/IP not required

SOCKETL

Link to this from the DOS TCP/IP product (if using TCP/IP)

When compiling programs in these environments there are many options available. For example, using Microsoft C7:

```
/A1fw /Gw /Zpl /J
```

with a stack size greater than 8 KB, preferably 16 KB.

HP-UX

libmqic.sl.

Linux libmqic.a library for non-threaded applications, or libmqic_r.a library for threaded applications.

OS/2 Warp

MQIC.LIB.

SINIX and DC/OSx

libmqic.so and libmqmcs.so.

If you want to use the programs on a machine that has only the MQSeries client for SINIX and DC/OSx installed, you must recompile the programs to link them with the client library:

```
$cc -o <prog> <prog>.c -lmqic -lmqmcs -lmqzmse -lnsl \
-lsocket -ldl -lproc -ltext
```

For DC/OSx append -liconv to the above command line.

Solaris

libmqic.so and libmqmcs.so.

If you want to use the programs on a machine that has only the WebSphere MQ client for Solaris installed, you must recompile the programs to link them with the client library:

Linking applications

```
$ /opt/SUNWspro/bin/cc -o <prog> <prog> c -mt -lmqic \
-lmqmcs -lsocket -lc -lnsl -ldl
```

The parameters must be entered in the correct order, as shown.

Sun Solaris, Intel Platform Edition

libmqic.so, libmqmcs.so, or libmqmzse.so

If you want to use the programs on a machine that has only the MQSeries client for Sun Solaris, Intel Platform Edition installed, you must recompile the programs to link them with the client library:

```
$ /opt/SUNWspro/bin/cc -o <prog> <prog>.c -mt -lmqic \
-lmqmcs -lmqmzse -lsocket -lc -lnsl -ldl
```

The parameters must be entered in the correct order, as shown.

Windows 3.1

LIBW.LIB, LLIBCEW.LIB, MQIC.LIB.

Windows 95, Windows

MQIC32.LIB.

Running 16-bit and 32-bit Windows clients

Previous versions of the MQSeries clients for Windows 95, Windows 98, and Windows NT included a version of MQIC.DLL that was 32-bit. When client code that was compiled as 16-bit is run using this DLL, it fails due to a name clash caused by the file MQIC.DLL. This has been rectified by replacing MQIC.DLL with MQIC32.DLL.

The file MQIC.DLL is no longer included in the 32-bit Windows client. If you have code linked with the MQIC.DLL you need to relink. If this is not possible, you can copy the MQIC32.DLL file to MQIC.DLL. Note that doing this will prevent you from running mixed 16-bit and 32-bit environments.

Linking C++ applications with the WebSphere MQ client code

If you are using a WebSphere MQ client supplied with an MQSeries Version 5.1 product and above, you can write your applications to run on the client in C++. Programs that use the WebSphere MQ C++ classes can be used successfully with MQSeries V5.1 or later and WebSphere MQ for z/OS servers only. For information about how to link your C++ applications and for full details of all aspects of using C++, see the *WebSphere MQ Using C++* manual.

Linking COBOL applications with the WebSphere MQ client code

AIX Link your COBOL application with the libmqicb.a library.

AT&T GIS UNIX

There is no COBOL support on AT&T GIS UNIX.

Compaq Tru64 UNIX

Link your COBOL application with the lmqmich library for non-threaded applications, or lmqmich_r library for threaded applications.

Compaq OpenVMS Alpha

Link your COBOL application with the MQICB.EXE library in SYS\$SHARE.

HP-UX

Link your COBOL application with the libmqicb.sl library.

If you are not using LU 6.2, consider linking to libsnaustubs.a (in /opt/lib for HP-UX) to fully resolve function names. The need to link to this library depends on how you are using the -B flag during the linking stage. For more information see the *WebSphere MQ Application Programming Guide*.

OS/2 Warp

If you have an OS/2 Warp COBOL application that you want to run in the client environment, link your application code with the MQICCB16 library for 16-bit COBOL, or the MQICCB library for 32-bit COBOL.

As with any WebSphere MQ application, you must compile it with the LITLINK directive. The COBLIB library must appear before the DOSCALLS library in the library list, and you need a stack size greater than 8 KB.

Additionally, your application needs a runtime stack size of at least 16 KB. More might be required depending on your application. One way to set this is to use the COBSW environment variable. For example:

```
set COBSW=/S16384
```

This stack size is sufficient to run the sample COBOL applications as clients.

SINIX and DC/OSx

If you have a COBOL application that you want to run in the client environment, you must recompile the programs to link them with the client library, libmqmcb.so:

```
cob -xU <prog>.cbl -lmqmb -lmqm -mqmcs -mqmzse -lmproc
```

For DC/OSx append liconv to the above command line.

Note: -lmqmb **must** come before -lmqm on the command line.

Solaris and Sun Solaris Intel Platform Edition

Link your COBOL application with the libmqicb.so library.

Windows and Windows 95

If you have a Windows or Windows 95 COBOL application that you want to run in the client environment, link your application code with the MQICCB library for 32-bit COBOL. The WebSphere MQ client for Windows and the MQSeries client for Windows 95 do not support 16-bit COBOL.

Linking PL/I applications with the WebSphere MQ client code

OS/2 Warp

Link your PL/I application with the MQIC.LIB library.

Windows 2000, Windows NT, and Windows XP

Link your PL/I application with the MQIC32.LIB library.

See the *WebSphere MQ Application Programming Guide* for further details.

Linking Visual Basic applications with the WebSphere MQ client code

You can link Visual Basic applications with the WebSphere MQ client code on Windows, Windows 3.1, and Windows 95.

Link your Visual Basic application with the following include files:

CMQB.bas

MQI

CMQBB.bas

MQAI

CMQCFB.bas

PCF commands

CMQXB.bas

Channels

Set mqtype=2 for the client in the Visual Basic compiler, to ensure the correct automatic selection of the client dll:

MQIC.dll

Windows 3.1

MQIC32.dll

Windows 95 and Windows 98

MQIC32.dll

Windows 2000, Windows NT, and Windows XP

Chapter 13. Running applications on WebSphere MQ clients

This chapter explains the various ways in which an application running in a WebSphere MQ client environment can connect to a queue manager. It covers the following topics:

- “Using environment variables” on page 164
- “Using the MQCNO structure” on page 164
- “Using DEFINE CHANNEL” on page 164
- “Role of the client channel definition table” on page 165
- “Examples of MQCONN calls” on page 165

When an application running in an WebSphere MQ client environment issues an MQCONN or MQCONNEX call, the client identifies how it is to make the connection. When an MQCONNEX call is issued by an application on a WebSphere MQ client, the MQI client library searches for the client channel information in the following order:

1. Using the contents of the *ClientConnOffset* or *ClientConnPtr* fields of the MQCNO structure (if supplied). These identify the channel definition structure (MQCD) to be used as the definition of the client connection channel.
2. If the MQSERVER environment variable is set, the channel it defines is used.
3. If the MQCHLLIB and MQCHLTAB environment variables are set, the client channel definition table they point to is used.
4. Finally, if the environment variables are *not* set, the client searches for a client channel definition table whose path and name are established from the DefaultPrefix in the mqs.ini file or the Registry for Windows. If this fails, the client uses the following paths:
 - Compaq OpenVMS Alpha
mqs_root:[mqm]amqc1chl.tab
 - OS/2 Warp
MQSWORHPATH\amqc1chl.tab

Or, if MQSWORHPATH is not set:

bootdrive:\mqm\amqc1chl.tab

- UNIX systems
/var/mqm/AMQCLCHL.TAB
- Windows and Windows 95
bootdrive:\mqm\amqc1chl.tab

Where bootdrive is obtained from the Software\IBM\MQSeries\CurrentVersion Registry entry under HKEY_LOCAL_MACHINE. This value is established when the WebSphere MQ client software is installed. If it is not found, a value of 'C' is used for bootdrive.

The first of the options described above (using the *ClientConnOffset* or *ClientConnPtr* fields of MQCNO) is supported only by the MQCONNEX call when running on a WebSphere MQ client, or MQSeries V5.1, or later, clients. If the application is running on an earlier MQSeries client, or if it is using MQCONN

Running applications

rather than MQCONN, the channel information is searched for in the remaining three ways in the order shown above. If the client fails to find any of these, the MQCONN or MQCONN call fails.

The channel name (for the client connection) must match the server-connection channel name defined on the server for the MQCONN or MQCONN call to succeed.

If you receive an MQRC_Q_MGR_NOT_AVAILABLE return code from your application with an error message in the error log file of AMQ9517 - File damaged, see “Migrating to a later release level of WebSphere MQ” on page 135.

Using environment variables

Client channel information can be supplied to an application running in a client environment by the MQSERVER, MQCHLLIB, and MQCHLTAB environment variables.

Using MQSERVER

If you use the MQSERVER environment variable to define the channel between your WebSphere MQ client machine and a server machine, this is the only channel available to your application, and no reference is made to the client channel definition table. In this situation, the ‘listener’ program that you have running on the server machine determines the queue manager to which your application will connect. It will be the same queue manager as the listener program is connected to.

If the MQCONN or MQCONN request specifies a queue manager other than the one the listener is connected to, or if *TransportType* is not recognized, the MQCONN or MQCONN request fails with return code MQRC_Q_MGR_NAME_ERROR.

Using MQCHLLIB and MQCHLTAB

The MQCHLLIB environment variable on the server specifies the path to the directory containing the client channel definition table. The MQCHLTAB environment variable (also on the server) specifies the name of the client channel definition table. For more information about these environment variables, see Chapter 10, “Using WebSphere MQ environment variables” on page 143.

Using the MQCNO structure

You can specify the definition of the channel in a channel definition structure (MQCD), which is supplied using the MQCNO structure of the MQCONN call. For more information see “Using the MQCNO structure on an MQCONN call” on page 132.

Using DEFINE CHANNEL

If you use the MQSC DEFINE CHANNEL command, the details you provide are placed in the client channel definition table. This file is accessed by the client, in channel name sequence, to determine the channel an application will use.

The contents of the *Name* parameter of the MQCONN or MQCONN call determines which server the client connects to.

Role of the client channel definition table

The client channel definition table is created when you define the first of the connections between a WebSphere MQ client and a server. See “Defining MQI channels” on page 128 for more information about what you have to define and how you do it.

Note: The same file can be used by more than one WebSphere MQ client. You change the name and location of this file using the MQCHLLIB and MQCHLTAB WebSphere MQ environment variables. See Chapter 10, “Using WebSphere MQ environment variables” on page 143 for information about environment variables.

Multiple queue managers

You might choose to define connections to more than one server machine because:

- You need a backup system.
- You want to be able to move your queue managers without changing any application code.
- You need to access multiple queue managers, and this requires the least resource.

Define your client-connection and server-connection channels on one queue manager only, including those channels that connect to a second or third queue manager. Do *not* define them on two queue managers and then try to merge the two client channel definition tables; this cannot be done. Only one client channel definition table can be accessed by the client.

Queue-sharing groups

You can connect your application to a queue manager that is part of a queue-sharing group. This can be done by using the queue-sharing group name instead of the queue manager name on the MQCONN or MQCONNX call.

The client channel definition should use the queue sharing group generic interface to connect to an available queue manager in the group. For more information, see “Connecting to the generic interface” on page 137. A check is made to ensure that the queue manager the listener connects to is a member of the queue sharing group.

For more information on shared queues, see the *WebSphere MQ for z/OS Concepts and Planning Guide* book, and the *WebSphere MQ Intercommunication* book.

Examples of MQCONN calls

In each of the following examples, the network is the same; there is a connection defined to two servers from the same WebSphere MQ client. (In these examples, the MQCONNX call could be used instead of the MQCONN call.)

There are two queue managers running on the server machines, one named SALE and the other named SALE_BACKUP.

Running applications

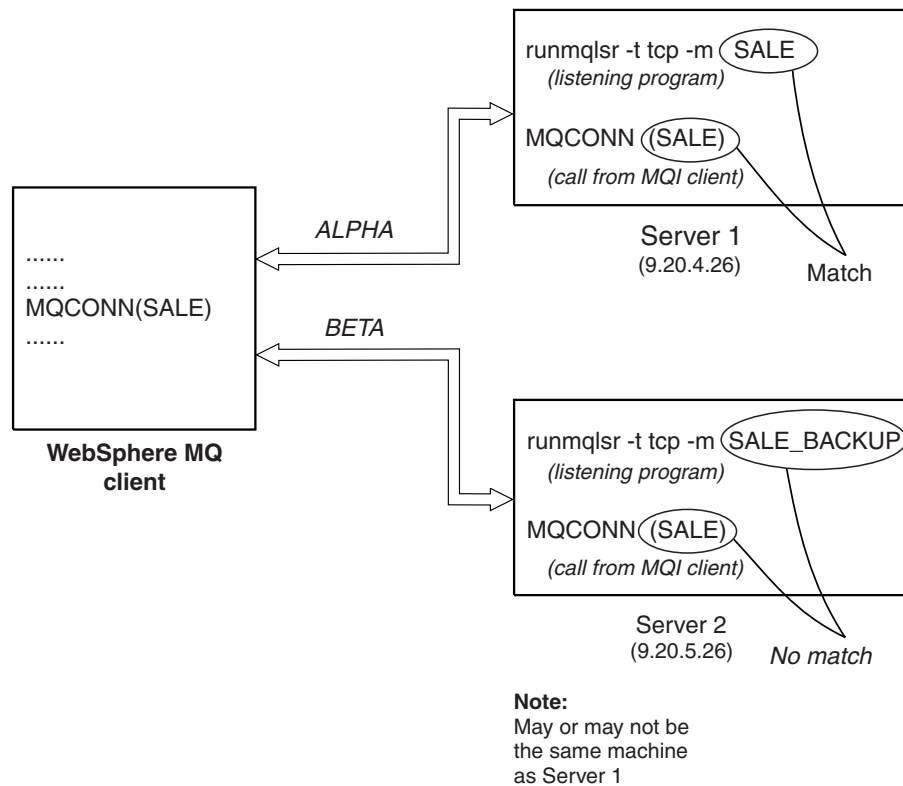


Figure 10. MQCONN example

The definitions for the channels in these examples are:

SALE definitions:

```
DEFINE CHANNEL(ALPHA) CHLTYPE(SVRCONN) TRPTYPE(TCP) +  
DESCR('Server connection to WebSphere MQ client')
```

```
DEFINE CHANNEL(APLHA) CHLTYPE(CLNTCONN) TRPTYPE(TCP) +  
CONNNAME(9.20.4.26) DESCR('WebSphere MQ client connection to server 1') +  
QMNAME(SALE)
```

```
DEFINE CHANNEL(BETA) CHLTYPE(CLNTCONN) TRPTYPE(TCP) +  
CONNNAME(9.20.5.26) DESCR('WebSphere MQ client connection to server 2') +  
QMNAME(SALE)
```

SALE_BACKUP definition:

```
DEFINE CHANNEL(BETA) CHLTYPE(SVRCONN) TRPTYPE(TCP) +  
DESCR('Server connection to WebSphere MQ client')
```

The client channel definitions can be summarized as follows:

Name	CHLTYPE	TRPTYPE	CONNNAME	QMNAME
ALPHA	CLNTCONN	TCP	9.20.4.26	SALE
BETA	CLNTCONN	TCP	9.20.5.26	SALE

What the examples demonstrate

Suppose the communication link to Server 1 is temporarily broken. The use of multiple queue managers as a backup system is demonstrated.

Each example covers a different MQCONN call and gives an explanation of what happens in the specific example presented, by applying the following rules:

1. WebSphere MQ searches the client channel definition table, in *channel name order*, looking in the queue manager name (QMNAME) field for an entry corresponding to the one given in the MQCONN call.
2. If a match is found, the transmission protocol and the associated connection name are extracted.
3. An attempt is made to start the channel to the machine identified by the connection name (CONNNAME). If this is successful, the application continues. It requires:
 - A listener to be running on the server.
 - The listener to be connected to the same queue manager as the one the client wishes to connect to (if specified).
4. If the attempt to start the channel fails and there is more than one entry in the client channel definition table (in this example there are two entries), the file is searched for a further match. If a match is found, processing continues at step 1.
5. If no match is found, or there are no more entries in the client channel definition table and the channel has failed to start, the application is unable to connect. An appropriate reason code and completion code are returned in the MQCONN call. The application can take action based on the reason and completion codes returned.

Example 1. Queue manager name includes an asterisk (*)

In this example the application is not concerned about which queue manager it connects to. The application issues:

```
MQCONN (*SALE)
```

Following the rules, this is what happens in this instance:

1. The client channel definition table is scanned (in channel name order) for the queue manager name SALE, matching with the application MQCONN call.
2. The first channel definition found to match is ALPHA.
3. An attempt to start the channel is made – this is NOT successful because the communication link is broken.
4. The client channel definition table is again scanned for the queue manager name SALE and the channel name BETA is found.
5. An attempt to start the channel is made – this is successful.
6. A check to see that a listener is running shows that there is one running. It is not connected to the SALE queue manager, but because the MQI call parameter

Running applications

| has an asterisk (*) included in it, no check is made. The application is
| connected to the SALE_BACKUP queue manager and continues processing.

Example 2. Queue manager name specified

The application requires a connection to a specific queue manager, named SALE, as seen in the MQI call:

```
MQCONN (SALE)
```

Following the rules, this is what happens in this instance:

1. The client channel definition table is scanned (in channel name order) for the queue manager name SALE, matching with the application MQCONN call.
2. The first channel definition found to match is ALPHA.
3. An attempt to start the channel is made – this is *not* successful because the communication link is broken.
4. The client channel definition table is again scanned for the queue manager name SALE and the channel name BETA is found.
5. An attempt to start the channel is made – this is successful.
6. A check to see that a listener is running shows that there is one running, but it is not connected to the SALE queue manager.
7. There are no further entries in the client channel definition table. The application cannot continue and receives return code MQRC_Q_MGR_NOT_AVAILABLE.

Example 3. Queue manager name is blank or an asterisk (*)

In this example the application is not concerned about which queue manager it connects to. This is treated in the same way as “Example 1. Queue manager name includes an asterisk (*)” on page 167.

Note: If this application were running in an environment other than a WebSphere MQ client, and the name was blank, it would be attempting to connect to the default queue manager. This is *not* the case when it is run from a client environment, as there can be more than one default queue manager.

The application issues:

```
MQCONN ("")
```

or

```
MQCONN (*)
```

Following the rules, this is what happens in this instance:

1. The client channel definition table is scanned (in channel name order) for a queue manager name that is blank, matching with the application MQCONN call.
2. The entry for the channel name ALPHA has a queue manager name in the definition of SALE. This does *not* match the MQCONN call parameter, which requires the queue manager name to be blank.
3. The next entry is for the channel name BETA.
4. The queue manager name in the definition is SALE. Once again, this does *not* match the MQCONN call parameter, which requires the queue manager name to be blank.

5. There are no further entries in the client channel definition table. The application cannot continue and receives return code MQRC_Q_MGR_NOT_AVAILABLE.

Running applications

Chapter 14. Solving problems

This chapter discusses the following topics:

- “WebSphere MQ client fails to make a connection”
- “Stopping WebSphere MQ clients” on page 172
- “Error messages with WebSphere MQ clients” on page 172
- “How to read the error log and FFDCs for DOS and Windows 3.1” on page 172
- “Using trace on DOS and Windows 3.1” on page 173
- “Using trace on OS/2 Warp, Windows, and Windows 95” on page 174
- “Using trace on AIX and AT&T GIS UNIX” on page 175
- “Using trace on Compaq OpenVMS Alpha, HP-UX, SINIX, DC/OSx, and Solaris” on page 176
- “Using trace on VM/ESA” on page 177

An application running in the WebSphere MQ client environment receives MQRC_* reason codes in the same way as WebSphere MQ server applications. However, there are additional reason codes for error conditions associated with WebSphere MQ clients. For example:

- Remote machine not responding
- Communications line error
- Invalid machine address

The most common time for errors to occur is when an application issues an MQCONN or MQCONNX and receives the response MQRC_Q_MQR_NOT_AVAILABLE. Look in the client error log for a message explaining the failure. There might also be errors logged at the server, depending on the nature of the failure. Also, check that the application on the WebSphere MQ client is linked with the correct library file.

WebSphere MQ client fails to make a connection

When the WebSphere MQ client issues an MQCONN or MQCONNX call to a server, socket and port information is exchanged between the WebSphere MQ client and the server. For any exchange of information to take place, there must be a program on the server machine whose role is to ‘listen’ on the communications line for any activity. If there is no program doing this, or there is one but it is not functioning correctly, the MQCONN or MQCONNX call fails, and the relevant reason code is returned to the WebSphere MQ application.

If the connection is successful, WebSphere MQ protocol messages are exchanged and further checking takes place. During the WebSphere MQ protocol checking phase, some aspects are negotiated while others cause the connection to fail. It is not until all these checks are successful that the MQCONN or MQCONNX call succeeds.

For information about the MQRC_* reason codes, see the *WebSphere MQ Application Programming Reference* manual.

Stopping WebSphere MQ clients

Even though an WebSphere MQ client has stopped, it is still possible for the process at the server to be holding its queues open. The queues will be closed when the communications layer detects that the partner has gone.

Error messages with WebSphere MQ clients

When an error occurs with a WebSphere MQ client system, error messages are put into the error files associated with the server, if possible. If the error cannot be placed there, the WebSphere MQ client attempts to place the error message in an error log in the root directory of the WebSphere MQ client machine.

Compaq OpenVMS Alpha, OS/2 Warp, UNIX systems, Windows, and Windows 95

Error messages for WebSphere MQ clients on Compaq OpenVMS Alpha, OS/2 Warp, UNIX systems, Windows, and Windows 95 are placed in the error logs in the same way they are for the respective WebSphere MQ server systems. Typically these files appear in:

- MQS_ROOT:[MQM.ERRORS] on Compaq OpenVMS Alpha
- /var/mqm/errors on UNIX systems
- /mqm/errors on OS/2 Warp, Windows, and Windows 95

DOS and Windows 3.1 clients

The log file **AMQERR01.LOG** is held on C:\ unless the MQDATA environment variable is used to override the default. See Chapter 10, "Using WebSphere MQ environment variables" on page 143 for details on how to use this and all other WebSphere MQ environment variables.

How to read the error log and FFDCs for DOS and Windows 3.1

RUNMQFMT reformats the trace, error, and FFDC files. Before running RUNMQFMT you must have access to the error message file, amq9.msg. You can give RUNMQFMT access either by putting the file in the local directory or by adding its location to the DOS APPEND statement.

RUNMQFMT has one optional parameter, the name of the file to be processed. If you do not specify a file name and tracing is on, FORMAT TRACE/ERROR FILE attempts to format the trace file; if tracing is not on, it attempts to format the error file. The output is written to *stdout* to enable you to browse it; alternatively you can redirect the output to a printer. The oldest message is listed first.

To print the output file: RUNMQFMT filename > printername

Note: The normal default 'printer name' is LPT1, which is the port assigned to a printer. Alternatively, you can direct the output to a file, replacing the printer name with the file name when you issue the command.

Your application program should handle any MQI reason codes to allow your program to end in a controlled manner because there is no MQI error handling within the product.

There are three ways of using RUNMQFMT:

1. Specify the full path and name of the error log.
2. Specify the name of the error log, in which case the default path is used.
3. Enter only the command name, RUNMQFMT. The command assumes that the error log is in the default location, unless this has been changed by the MQDATA environment variable. If tracing is on, the trace is formatted; if tracing is off, the error log is formatted.

Using trace on DOS and Windows 3.1

Use the MQTRACE environment variable to set tracing on (see “MQTRACE (DOS, Windows 3.1, and VM/ESA only)” on page 148). Specify the name of the file to which you want all the trace entries to be put. You can further define the use of this file by specifying flags:

- c** Trace the communications flow.
- m** Do not query the configuration of the machine your application is running on. Use this option if exceptions occur when normal tracing is switched on.
- w** Write a new instance of the trace file for each program. If this is not set, the trace entries continue to be added to a single trace file.

Note: If you are using this trace option on two or more applications at the same time, you must specify a different trace file name for each of the applications to guarantee that no trace entries are lost.

Figure 11 on page 174 shows an example.

Example DOS trace data

The following example shows an extract from a trace for a DOS MQSeries client:

Trace

```
Trace started on Mon Oct 19 10:48:42 1998
PC 750-P133 - DOS V20.10 (Rev.2) RAM [BIOS Rev.5]
10:48:42 MQCONN
      rrxOpenChannelDef
      rrxOpenChannelDef RC=0 OK
      rrxGetFirstChannelDef
      rrxGetFirstChannelDef RC=0 OK
      rriInitSess
      rriAddStatusEntry
      rriAddStatusEntry RC=0 OK
      rriInitExits
      rriInitExits RC=0 OK
      ccxNetWorkInit
...
...
      ccxQueryProcAddr RC=0 OK
      cciLoadLibrary RC=0 OK
      ccxNetWorkInit RC=0 OK
      ccxAllocConv
      cciNetbAllocConv
      cciNetbAllocConv RC=0 OK
      ccxAllocConv RC=0 OK
      ccxAllocMem
      ccxAllocMem RC=0 OK
      ccxSend
      cciNetbSend
-----
I 10:48:53 Outbound 72 bytes.
I 54534820 00000048 02010100 00000000 TSH....H.....
I 00000000 22020000 52030000 49442020 .....R...ID..
I 02250000 00000000 FE0F0000 00004000 .....
I 00000000 4F533250 4743312E 53525620 ....OS2PGC1.SRV.
I 20202020 20202020 .....
-----
      cciNetbSend RC=0 OK
      ccxSend RC=0 OK
...
```

Figure 11. Extract from a DOS client trace

The entries in the box represent data sent or received over communications links.

Using trace on OS/2 Warp, Windows, and Windows 95

MQSeries for OS/2 Warp and WebSphere MQ for Windows use the following commands for the client trace facility:

strmqtrc

to start early tracing

endmqtrc

to end tracing

A client on Windows or Windows 95 uses the following commands for the client trace facility:

strmqtrc -t(TraceType)

to start tracing

endmqtrc

to end tracing

File names for trace files

Trace file names are constructed in the following way:

AMQppppp.TRC

where ppppp is the process ID (PID) of the process producing the trace.

Notes:

1. The value of the process ID can contain fewer or more digits than shown in the example.
2. There is one trace file for each process running as part of the entity being traced.

How to examine First Failure Support Technology™ (FFST) files

The files are produced already formatted and are in the \mqm\errors directory.

Note: For V5 clients on OS/2 Warp, Windows, and Windows 95, the errors directory is not created automatically during installation. The errors directory should be created before starting the trace.

These are normally severe, unrecoverable errors and indicate either a configuration problem with the system or a WebSphere MQ internal error.

The files are named AMQnnnnn.mm.FDC, where:

- nnnnn is the process id reporting the error
- mm is a sequence number, normally 0

When a process creates an FFST™ it also sends a record to syslog. The record contains the name of the FFST file to assist in automatic problem tracking.

The syslog entry is made at the “user.error” level.

The WebSphere MQ trace utility is explained in detail in the *WebSphere MQ System Administration Guide* manual.

Using trace on AIX and AT&T GIS UNIX

WebSphere MQ for AIX and MQSeries for AT&T GIS UNIX use the standard UNIX system trace. Tracing is a two step process:

1. Gather the data
2. Format the results

WebSphere MQ uses two trace hook identifiers:

X'30D' This event is recorded by WebSphere MQ on entry to or exit from a subroutine.

X'30E' This event is recorded by WebSphere MQ to trace data such as that being sent or received across a communications network.

Trace provides detailed execution tracing to help you to analyze problems. IBM service support personnel might ask for a problem to be recreated with trace enabled. The files produced by trace can be very large, so it is important to qualify a trace, where possible. For example, you can optionally qualify a trace by time and by component.

Trace

The best way to trace a single WebSphere MQ application is to run trace synchronously, and run the application from within the trace program. For example:

```
trace -j30D,30E -o trace.trc
```

At the prompt, enter the command that you want traced, prefixed by an exclamation mark (!). For example, type:

```
!amqspuic TESTQ
```

When the command is complete, type quit to stop trace and exit the trace program.

To view the trace, you must format the file using the trcrpt command. You can then view the trace. The following example allows you to view the trace in the file trace.fmt:

```
trcrpt -t /usr/mqm/lib/amqtrc.fmt trace.trc > trace.fmt
```

If you want to trace more than one application at the same time, or you want to start trace while an application is already running, start trace asynchronously using the -a flag. For example:

```
trace -a -j30D,j30E -o trace.trc
```

Trace continues to run until you stop it using the trstop command. The trace file can then be formatted as before using the trcrpt command.

The WebSphere MQ trace utility is explained in detail in the *WebSphere MQ System Administration Guide* manual and the *MQSeries for AT&T GIS UNIX System Management Guide*.

Using trace on Compaq OpenVMS Alpha, HP-UX, SINIX, DC/OSx, and Solaris

Compaq OpenVMS Alpha, HP-UX, SINIX, Solaris, and DC/OSx use the following commands for the WebSphere MQ client trace facility:

strmqtrc -e

to start early tracing

endmqtrc -e

to end early tracing

dspmqtrc <filename>

to display a formatted trace file

For more information about the trace commands, see the *WebSphere MQ System Administration Guide* manual for Version 5.3 products, or the *System Management Guide* for your platform for products earlier than Version 5.3.

The trace facility uses a number of files, which are:

- One file for each entity being traced, in which trace information is recorded
- One additional file on each machine, to provide a reference for the shared memory used to start and end tracing
- One file to identify the semaphore used when updating the shared memory

Files associated with trace are created in a fixed location in the file tree, which is /var/mqm/trace.

On Compaq OpenVMS Alpha systems, the files are in the MQS_ROOT:[MQM.TRACE] directory.

All queue manager tracing, all early tracing, and all @SYSTEM tracing takes place to files in this directory.

You can handle large trace files by mounting a temporary file system over this directory.

File names for trace files

Trace file names are constructed in the following way:

AMQppppp.TRC

where ppppp is the process ID (PID) of the process producing the trace.

Notes:

1. The value of the process ID can contain fewer or more digits than shown in the example.
2. There is one trace file for each process running as part of the entity being traced.

How to examine FFSTs

FFST logs are written when a severe WebSphere MQ error occurs. They are written to the directory /var/mqm/errors, except on OS/2. On OS/2 they are written to the FFST logging facility.

These are normally severe, unrecoverable errors and indicate either a configuration problem with the system or a WebSphere MQ internal error.

The files are named AMQnnnnn.mm.FDC, where:

nnnnn is the process id reporting the error
mm is a sequence number, normally 0

When a process creates an FFST it also sends a record to syslog. The record contains the name of the FFST file to assist in automatic problem tracking.

The syslog entry is made at the "user.error" level.

The WebSphere MQ trace utility is explained in detail in the *WebSphere MQ System Administration Guide* manual for WebSphere MQ Version 5.3 products, and the relevant *System Management Guide* for other platforms.

Using trace on VM/ESA

Use the MQTRACE environment variable to set the tracing on (see "MQSERVER" on page 146). Specify the name of the file to which you want all the trace entries to be put. Unlike MQTRACE for DOS and Windows 3.1, there are no optional flags that can be set. Setting the MQTRACE variable initiates a trace for all the functions of the MQSeries client for VM/ESA.

Example VM/ESA trace data

The following example shows an extract of a trace for a VM/ESA MQSeries client:

Application programming

```
MQSeries Trace started at 10/09/95 16:07:14
< xcsInitialize (rc = OK)
-> MQCONN
--> rrxOpenChannelDef
----> xcsGetMem
<-- xcsGetMem (rc = OK)
<- rrxOpenChannelDef (rc = OK)
--> rrxGetFirstChannelDef
<- rrxGetFirstChannelDef (rc = OK)
--> rriInitSess
----> xcsGetMem
<-- xcsGetMem (rc = OK)
...

...
----> rriTermExits
<-- rriTermExits (rc = OK)
----> rriDeleteStatusEntry
----> xcsFreeMem
<-- xcsFreeMem (rc = OK)
<- rriDeleteStatusEntry (rc = OK)
----> xcsFreeMem
<-- xcsFreeMem (rc = OK)
<- rriFreeSess (rc = OK)
< MQDISC
```

Figure 12. Extract from a VM/ESA client trace

Appendix. Notices

This information was developed for products and services offered in the United States. IBM may not offer the products, services, or features discussed in this information in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

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

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
U.S.A.

For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

IBM World Trade Asia Corporation
Licensing
2-31 Roppongi 3-chome, Minato-ku
Tokyo 106, Japan

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law:

INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY, OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the information. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this information at any time without notice.

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

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

Notices

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

IBM United Kingdom Laboratories,
Mail Point 151,
Hursley Park,
Winchester,
Hampshire,
England
SO21 2JN.

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

The licensed program described in this information and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Programming License Agreement, or any equivalent agreement between us.

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

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs. You may copy, modify, and distribute these sample programs in any form without payment to IBM for the purposes of developing, using, marketing, or distributing application programs conforming to IBM's application programming interfaces.

Trademarks

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

AIX	AS/400	CICS
CICS/VSE	DB2	ESCO
First Failure Support Technology	FFST	IBM
IBMLink	IMS	iSeries
MQSeries	OS/2	OS/390
OS/400	POWERparallel	PowerPC
POWERserver	RS/6000	SP
SupportPac	Visual Age	VM/ESA
VSE/ESA	VTAM	WebSphere
WIN-OS/2	z/OS	zSeries

NetView is a trademark of Tivoli Systems Inc. in the United States, or other countries, or both.

LotusScript is a trademark of Lotus Development Corporation in the United States, or other countries, or both.

Intel and Pentium are registered trademarks of Intel Corporation in the United States and/or other countries.

Microsoft, Visual Basic, Windows, and Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States and/or other countries.

Java and all Java-based trademarks and logos are trademarks of Sun Microsystems, Inc. in the United States and/or other countries.

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

Other company, product, or service names may be trademarks or service marks of others.

Index

Numerics

16-bit mode 157
32-bit mode 157

A

access control 125
Active Directory
 accessing client-connection channel definitions 134
 locating LDAP servers that hold CRLs 141
 specifying that an MQI channel uses SSL 139
add routing entry 103
addrtge 103
ADDRTGE 103
advanced installation
 client 75
advantages of using WebSphere MQ clients 5
AIX client
 components for 38
 custom installation 41
 error messages 172
 hardware and software 15
 installing 40
 migrating to and from SSL support 44
 removing 44
 trace 175
AMQCLCHL.TAB 134
AMQERR01.FDC 145
AMQERR01.LOG 145
AMQSGETO sample program 119
AMQSGETC sample program 113, 119
AMQSGETW sample program 113
AMQSPUT0 sample program 118
AMQSPUTC sample program 113, 118
AMQSPUTW sample program 113, 118
applications
 building 155
 connected to multiple queue managers 163
 connected to multiple servers 163
 connection to server 163
 designing 152
 in different environments 156
 non-version 5 12
 version 5.1 12
applications on WebSphere MQ clients 5
AppIType 157
AT&T GIS UNIX client
 hardware and software 17
 trace 175
authentication 123
authentication information object 140
automatic definition of channels 128
AUTOSTART 104

B

base product 81
benefits of using WebSphere MQ clients 5
building applications 155
building applications for WebSphere MQ clients
 MQI 136

C

C applications
 linking 158
C++ applications
 linking 155, 160
 using 155
CCSID (coded character set identifier) 151
CEDA 158
certificate revocation list (CRL) 139
channel
 client connection 164
 client-connection 128
 definition
 automatic 128
 by user 129
 connecting to a queue-sharing group 137
 on the client and server 129
 using MQSC commands 129
 message 127
 MQI 127
 overview 127
 server-connection 128, 132
 starting 114
 user-defined 129
channel definition
 maximum message length 151
 on the client 130
 on the server 129, 132
 overview 127
channel exits
 ClientExitPath 136
 ExitsDefaultPath 136
 overview 136
 path to exits 136
 receive 136
 security 136
 send 136
channel initiator, starting
 z/OS example 116
channels
 stopping 137
CICS applications (non-z/OS)
 CEDA 158
 CICSENV.COMD file 158
 environment specifics 158
 MQSERVER 158
class of routing entry 104

client
 installation
 advanced methods 75
 LAN 72
 unattended 73
 unattended uninstall 77
 uninstalling 76
Client CD-ROM 35
client channel definition table
 directory path 144
 how it is used 165
 locating LDAP servers that hold CRLs 140
 migrating to a later release level of WebSphere MQ 135
 name of 145
 specifying that an MQI channel uses SSL 139
 where to find it 134, 163
client files on the server 81
client library file 158
client setup
 example 116
client to server connection 4, 128
client-connection channel
 defining 128
 example 117
client-connection, defining 133
ClientExitPath 136
clients (Version 5)
 removing 37
clients available from Internet 12
clients overview 3
CLNTCONN 133
COBOL
 link libraries 160
 linking applications 160
coded character set identifier (CCSID) 151
commands
 MQSC 129
communication protocol
 DECnet 91
 LU 6.2 91
 NetBIOS 91
 overview 12
 SPX 91
 TCP/IP 91
communication type 91
communications
 configuring 91
 overview 12
Communications Manager/2 100
compact installation
 client 71
Compaq NonStop Kernel server 81
Compaq OpenVMS Alpha
 hardware and software 19
 installing clients 82
Compaq OpenVMS Alpha client
 error messages 172

- Compaq OpenVMS Alpha client
 - (*continued*)
 - trace 176
- Compaq Tru64 UNIX client
 - hardware and software 18
 - installing 45
 - NLSPATH environment variable 46
 - translated messages 46
 - uninstalling 46
- components
 - Linux client 51
 - OS/2 Warp client 56
 - Solaris client 64
- configuring communications 91
- connection
 - client to server 4, 128
 - DECnet 111
 - LU 6.2 99
 - NetBIOS 106
 - overview 12
 - queue managers 163
 - server to client 128
 - SPX 108
 - TCP/IP 93
 - to queue-sharing group 136
- creating
 - groups
 - on AIX client 38
 - on Compaq Tru64 UNIX client 45
 - on Linux client 53
 - Sun Solaris, Intel Platform Edition client 67
 - user ID
 - on AIX client 38
 - on Compaq Tru64 UNIX client 45
 - on Linux client 53
 - Sun Solaris, Intel Platform Edition client 67
- creating the SMS software package 75
- CRL
 - See* certificate revocation list (CRL)
- CRL information 140
- CSQUTIL
 - See* WebSphere MQ utility program (CSQUTIL)
- custom installation
 - AIX client 41
 - client 71
- customization
 - z/OS example 116

D

- data compression 136
- data conversion 156
- data encryption 136
- DC/OSx
 - hardware and software 26
- DC/OSx client
 - trace 176
- DCE security exits 124
- DECnet 91
- DECnet connections 111
- default object definition
 - example 114

- defining channels
 - connecting to a queue-sharing group 137
 - on the client 130
 - on the client and server 129
 - on the server 129, 132
 - overview 127
 - using MQSC 164
 - using MQSC commands 129
- delete queue manager 120
- desktop client 80
- dltmqm 120
- DOS client
 - error messages 172
 - hardware and software 20
 - trace 148
 - example 173
- DQM, starting
 - z/OS example 116
- dspmqrtr trace command 176

E

- ending verification 120
- endmqm 120
- endmqtrtr trace command 174, 176
- environment variables 143
 - change setting 144
- COBSW 161
- connecting to a client 164
- creating channel definitions 128
- display current setting 144
- MQ_PASSWORD 124, 146
- MQ_USER_ID 124, 125, 148
- MQCCSID 144
- MQCHLLIB 144, 163
- MQCHLTAB 145, 163
- MQDATA 145, 172
- MQNAME 146
- MQSERVER 146, 163
- MQSWORHPATH 148
- MQTRACE 148

- error files, formatting 172
- error log
 - AMQERR01.LOG 145
 - for automatic installation 60
 - overview 171
 - reading 172

- error messages 171
- example
 - customize z/OS 116
 - default object definition 114
 - inetd setup 114
 - installation verification 113
 - local queue, creating 114
 - MQSC, starting 114
 - MQSC, stopping 114
 - queue manager
 - creating 114
 - starting 114
 - server-connection channel,
 - creating 114
 - setting up the server 114
 - start the channel initiator (OS/390) 104
 - start the channel initiator (z/OS) 97
 - start the listener (OS/390) 97, 104

- example (*continued*)
 - trace data (DOS) 173
 - trace data (VM/ESA) 177
- exits
 - overview 136
 - receive 136
 - security 124, 136
 - send 136
- ExitsDefaultPath 136
- Explorer, WebSphere MQ 130

F

- features
 - for a client installation 69
 - unattended installation 74
- FFDC (First Failure Data Capture) 172
- FFST, examining 175, 177
- First Failure Data Capture (FFDC) 172
- First Failure Support Technology 175
- format trace and error files
 - MQDATA 145
- formatting error files 172
- formatting trace files 172

G

- getting a message from a queue
 - example (not VM/ESA or Windows 3.1) 119
 - example (VM/ESA) 119
 - example (Windows 3.1) 119
- groups, creating
 - on AIX client 38
 - on Compaq Tru64 UNIX 45
 - on HP-UX client 48
 - on Linux client 53
 - on Solaris client 64
 - on Sun Solaris, Intel Platform Edition 67

H

- hardware requirements
 - AIX client 15
 - AT&T GIS UNIX client 17
 - Compaq OpenVMS Alpha client 19
 - Compaq Tru64 UNIX client 18
 - DOS client 20
 - HP-UX client 21
 - Linux client 22
 - NCR UNIX client 17
 - OS/2 Warp client 24
 - SINIX and DC/OSx client 26
 - Solaris client 28
 - Sun Solaris, Intel Platform Edition client 29
 - VM/ESA client 30
 - Windows 3.1 client 33
 - Windows 95 client 34
 - Windows client 31
- history log 60
- HP-UX client
 - groups, creating 48
 - hardware and software 21
 - installing 49

HP-UX client (*continued*)

- migrating to and from SSL support 50
- national language support 49
- trace 176
- uninstalling 50
- user ID, creating 48

I

inetd setup 114

- z/OS example 116

installation

- client
 - advanced methods 75
 - LAN 72
 - unattended 73
- client types 69
- compact
 - client 71
- custom
 - client 71
- directory (Compaq Tru64 UNIX) 45
- modifying
 - using Add/Remove Programs 72
 - using the Client CD-ROM 71
- preparing for 11
- response file
 - client 74
- typical
 - client 69
- uninstalling the Compaq Tru64 UNIX client 46
- verification 113

installation (non-Version 5)

- Base product 79
- clients 79
- preparing 79
- server 79

installation (Version 5)

- clients 35
- server 35

installation parameters

- OS/2 Warp client 60

installation response files

- OS/2 Warp client 61

installing

- AIX client 40
- AIX client (custom install) 41
- client on the server 37
- clients from Compaq OpenVMS Alpha 82
- clients from MQSeries products 80
- clients from SupportPacs 81
- Compaq OpenVMS Alpha client from MQSeries for Compaq OpenVMS Alpha 82
- Compaq Tru64 UNIX client 45
- DOS client from MQSeries for Compaq OpenVMS Alpha 84
- DOS client from UNIX systems 87
- error log parameters 60
- history log parameter 60
- HP-UX client 49
- Linux client 51
- MQSeries server 81
- OS/2 Warp client 56

installing (*continued*)

- OS/2 Warp client from MQSeries for Compaq OpenVMS Alpha 83
- OS/2 Warp client from UNIX systems 86
- Solaris client 65
- Sun Solaris, Intel Platform Edition client 67
- UNIX client from UNIX system 85
- VM/ESA client 88
- Windows 3.1 client from MQSeries for Compaq OpenVMS Alpha 84
- Windows 3.1 client from UNIX systems 87
- Windows client 69

installing (non-Version 5) 79

- clients from MQSeries on UNIX systems 85

installing (Version 5) 35

- Internet 81
- Internet - clients available 12
- IPX parameters 108

K

kernel configuration

- Sun Solaris, Intel Platform Edition client 68

keywords for response files 62

L

LAN installation

- client 72

language

- installation from client CD-ROM 70

languages supported 155

LDAP server

- See* Lightweight Directory Access Protocol (LDAP) server

library file, client 158

Lightweight Directory Access Protocol (LDAP) server 139

linking applications for WebSphere MQ clients

- MQI 136

linking with WebSphere MQ client code

- C applications 158
- C++ applications 160
- COBOL applications 160
- PL/I applications 161
- Visual Basic applications 162

Linux client

- components 51
- hardware and software 22
- installing 51
- migrating to and from SSL support 55
- national language support 54
- uninstalling 54

listener 91

listener, starting

- z/OS example 116

listening on a Windows server 101

listening on OS/2 Warp 100

listening on TCP/IP

- Compaq NonStop Kernel 98
- MQSeries for Compaq OpenVMS Alpha 97
- OS/2 Warp 94
- OS/390 97
- OS/400 97
- UNIX 96
- VSE/ESA 98
- Windows server 95

local name definition 106, 107

local queue

- example 114

local queue, creating

- example (z/OS) 116

log, error 172

LU 6.2 91

LU 6.2 connections 99

LU 6.2 on a client

- Compaq OpenVMS Alpha 105
- OS/2 Warp 99
- UNIX systems 102
- Windows 101

LU 6.2 on a server

- Compaq NonStop Kernel 104
- OS/2 Warp 100
- OS/390 104
- OS/400 103
- UNIX systems 102
- Windows 101

M

maintenance parameters

- OS/2 Warp client 60

maximum channels on VM/ESA 135

MCAUSER 125

MCAUserIdentifier 125

message

- errors on AIX 172
- errors on Compaq OpenVMS Alpha 172
- errors on DOS and Windows 3.1 172
- errors on OS/2 Warp 172
- errors on UNIX systems 172
- errors on Windows 172
- errors on Windows 95 172
- maximum length 151

message channel 127

Message Queue Interface (MQI) 151

messages

- translated 44

Microsoft Management Console 130

Microsoft System Management Server (SMS) 75

migrating to and from SSL support

- AIX client 44
- HP-UX client 50
- Linux client 55
- Solaris client 66

MMC 130

modifying the installation

- using Add/Remove Programs client 72
- using the Client CD-ROM 71

MQ_PASSWORD 124, 146

MQ_USER_ID 124, 125, 148

- MQAIR structure 140
- MQBACK 153
- MQCCSID
 - what it does 144
- MQCD structure
 - introduction 132
 - specifying that an MQI channel uses SSL 139
- MQCHLLIB
 - how it is used 163
 - introduction 134
 - what it does 144
- MQCHLTAB
 - how it is used 163
 - introduction 134
 - what it does 145
- MQCMIT 153
- MQCNO structure
 - introduction 132
 - locating LDAP servers that hold CRLs 140
 - specifying that an MQI channel uses SSL 139
- MQCONN 163
- MQCONN or MQCONNx failure 171
- MQCONNx 163
- MQCONNx call
 - defining a client-connection channel 132
 - locating LDAP servers that hold CRLs 140
 - specifying that an MQI channel uses SSL 139
 - Using 153
- MQDATA 145
- MQI
 - application in client environment 156
 - building applications for WebSphere MQ clients 136
 - linking applications for WebSphere MQ clients 136
- MQI (Message Queue Interface) 151
- MQI channels 127
- MQINQ 152
- MQNAME 146
- mqs.ini file
 - path to exits 136
 - role in connecting a client 163
- MQSC
 - commands 129
 - starting 129
 - example 114
 - stopping 114
- MQSCO structure
 - introduction 132
 - locating LDAP servers that hold CRLs 140
- MQSeries Explorer 130
- MQSeries for Compaq OpenVMS Alpha server 81
- MQSeries server
 - installing 81
 - name of 146
- MQSERVER
 - cancelling 148
 - defining a client-connection channel 117

- MQSERVER (*continued*)
 - how it is used 163
 - relationship with client channel definition table 165
 - specifying that an MQI channel uses SSL 139
 - using 130
 - what it does 146
- MQSWORCPATH 148
- MQTRACE 173
 - trace data (DOS) 173
 - trace data (VM/ESA)
 - example 177
 - what it does 148
- multiple queue managers 100, 101, 165

N

- namelist 140
- national language
 - installation from client CD-ROM 70
- national language support 44
 - HP-UX client 49
 - Linux client 54
- NCR UNIX client
 - hardware and software 17
- NDF file configuration 101
- NetBIOS 91
 - connections 106
 - name definition 106, 107
 - on a client 106
 - on a server
 - OS/2 Warp 106
 - Windows 107
- NLSPATH environment variable 44
 - Compaq Tru64 UNIX client 46
- non-version 5 clients 12

O

- OS/2 Warp client
 - components 56
 - error messages 172
 - hardware and software 24
 - installation parameters 60
 - installation response files 61
 - installing 56
 - maintenance parameters 60
 - trace 174
 - unattended installation 58
 - uninstalling 63
- OS/400 client 4

P

- Package Definition File (PDF) 75
 - parameters for unattended installation 60
- password 124
- path to exits
 - mqs.ini file 136
- PDF (Package Definition File) 75
- PL/I
 - link libraries 161
 - linking applications 161
- platform support 12

- platforms
 - for WebSphere MQ clients 11
 - for WebSphere MQ servers 11
- platforms for WebSphere MQ clients
 - more available 12
- preparing for installation 11
- problem determination 171
- process definition 157
- putting a message on the queue
 - example (not VM/ESA or Windows 3.1) 118
 - example (VM/ESA) 118
 - example (Windows 3.1) 118

Q

- queue manager
 - definition 114
 - deleting 120
 - maximum message length 151
 - starting 114
 - starting (z/OS example) 116
 - stopping 120
- queue managers 163
- queue-sharing groups 136, 165

R

- receive exit 136
- removal response file format
 - client 78
- removing
 - client 76
 - using Add/Remove Programs
 - Windows client 77
- removing an AIX client 44
- response file
 - client 74
 - removal 78
- response files
 - installation 61
 - keywords 62
 - structure 62
- return codes 171
- routing entry class 104
- RUNMQFMT 145, 172
- runmqmtmc 157

S

- sample programs
 - AMQSGETO 119
 - AMQSGETC 113, 119
 - AMQSGETW 113
 - AMQSPUTO 118
 - AMQSPUTC 113, 118
 - AMQSPUTW 113, 118
 - GET 113
 - PUT 113
- Secure Sockets Layer (SSL) 139
- security
 - access control 125
 - authentication 123
 - environment variables 124
 - exit 124
 - on a WebSphere MQ client 123

- security (*continued*)
 - password 124
 - user ID 124
 - security exit 136
 - send exit 136
 - server
 - Compaq NonStop Kernel 81
 - connecting to a client 163
 - connection, defining 132
 - installing 81
 - MQSeries for Compaq OpenVMS Alpha 81
 - platform support 11
 - VSE/ESA 81
 - z/OS 81
 - Server CD-ROM 35
 - server setup
 - VSE/ESA example 116
 - server to client connection 128
 - server-connection channel
 - defining 128
 - example (not z/OS or VSE/ESA) 114
 - example (z/OS) 116
 - iSeries example 116
 - setmqcrl command 141
 - setmqscp command
 - introduction 134
 - specifying that an MQI channel uses SSL 139
 - setting up the server
 - AS 400 example 115
 - example 114
 - example (Windows) 115
 - using Windows explorer 115
 - z/OS example 116
 - setting up WebSphere MQ clients 6
 - shared queues 136, 165
 - silent installation
 - client 73
 - silent uninstall
 - client 77
 - simple client-connection channel
 - definition 130
 - SINIX
 - hardware and software 26
 - SINIX client
 - trace 176
 - SMIT
 - installing AIX client 41
 - using to create IDs 39
 - SMS
 - creating the software package 75
 - creating the WebSphere MQ job 76
 - using with WebSphere MQ 75
 - software requirements
 - AIX client 15
 - AT&T GIS UNIX client 17
 - Compaq OpenVMS Alpha client 19
 - Compaq Tru64 UNIX client 18
 - DOS client 20
 - HP-UX client 21
 - Linux client 22
 - NCR UNIX client 17
 - OS/2 Warp client 24
 - SINIX and DC/OSx client 26
 - Solaris client 28
 - software requirements (*continued*)
 - Sun Solaris, Intel Platform Edition
 - client 29
 - VM/ESA client 30
 - Windows 3.1 client 33
 - Windows 95 client 34
 - Windows client 31
 - Solaris client
 - components 64
 - hardware and software 28
 - installing 65
 - migrating to and from SSL support 66
 - trace 176
 - translated messages 66
 - uninstalling 66
 - solving problems 171
 - SPX 91
 - connections 108
 - default socket 147
 - on a client
 - any platform 108
 - on a server
 - OS/2 Warp, Windows 108
 - parameters 108
 - on DOS 109
 - on OS/2 Warp 108
 - on Windows 109
 - on Windows 3.1 109
 - SSL
 - See* Secure Sockets Layer (SSL)
 - SSLCIPH parameter 139
 - SSLCipherSpec field 139
 - SSLCRLNameList attribute 140
 - start the channel initiator
 - example (OS/390) 104
 - example (z/OS) 97
 - start the listener
 - example (OS/390) 97, 104
 - stop queue manager 120
 - strmqtrc trace command 174, 176
 - structure of response files 62
 - Sun Solaris, Intel Platform Edition client
 - hardware and software 29
 - installing 67
 - kernel configuration 68
 - uninstalling 68
 - SupportPacs 81
 - SVRCONN 132
 - syncpoint
 - considerations 153
 - coordination 158
 - system administration 123
 - System Management Interface Tool (SMIT)
 - installing AIX client 41
 - using to create IDs 39
 - System Management Server (SMS) 75
 - T
 - task list 6
 - TCP/IP 91
 - default port 147
 - TCP/IP connections 93
 - TCP/IP on a client 94
 - TCP/IP on a server
 - Compaq NonStop Kernel 98
 - MQSeries for Compaq OpenVMS Alpha 97
 - OS/2 Warp 94
 - OS/390 97
 - OS/400 97
 - UNIX systems 96
 - VSE/ESA 98
 - Windows 95
 - TPNAME
 - Compaq OpenVMS Alpha 105
 - OS/2 Warp systems 99
 - UNIX systems 102
 - Windows 101
 - trace
 - AIX 175
 - AT&T GIS UNIX 175
 - Compaq OpenVMS Alpha 176
 - data example (DOS) 173
 - data example (VM/ESA) 177
 - DC/OSx 176
 - DOS and Windows 173
 - HP-UX 176
 - OS/2 Warp 174
 - SINIX 176
 - Solaris 176
 - VM/ESA 177
 - Windows 174
 - Windows 95 174
 - trace command
 - dspmqrtrc 176
 - endmqtrc 174, 176
 - strmqtrc 174, 176
 - trace file
 - AMQERR01.FDC 145
 - trace files, formatting 172
 - Transaction Processing SupportPacs 81
 - translated message
 - Compaq Tru64 UNIX client 46
 - translated messages
 - client 44
 - HP-UX client 49
 - Linux client 54
 - Solaris client 66
 - transmission protocol
 - DECnet 91
 - LU 6.2 91
 - NetBIOS 91
 - SPX 91
 - TCP/IP 91
 - trigger monitor for WebSphere MQ
 - clients 157
 - types of client installation 69
 - typical installation
 - client 69
- ## U
- unattended installation
 - client 73
 - OS/2 Warp client 58
 - unattended removal
 - client 77
 - uninstallation response file format
 - client 78

- uninstalling
 - client 76
 - Compaq Tru64 UNIX client 46
 - HP-UX client 50
 - Linux client 54
 - OS/2 Warp client 63
 - Solaris client 66
 - Sun Solaris, Intel Platform Edition client 68
 - using Add/Remove Programs
 - Windows client 77
- UNIX systems (non-Version 5)
 - installing clients 85
- UNIX systems clients
 - error messages 172
- user definition of channels 129
- user ID 124
- user ID, creating
 - on AIX client 38
 - on Compaq Tru64 UNIX 45
 - on HP-UX client 48
 - on Linux client 53
 - on Solaris client 64
 - on Sun Solaris, Intel Platform Edition 67

V

- verification, ending 120
- verifying installation 113
 - example 113
- Version 5.1 clients 12
- Visual Basic
 - link libraries 162
 - linking 162
 - linking applications 162
- VM/ESA
 - client trace 148
- VM/ESA client
 - hardware and software 30
 - installing 88
 - maximum number of channels 135
 - trace 177
- VSE/ESA
 - setting up the server 116
 - TCP/IP setup 98
- VSE/ESA server 81

W

- Web - clients available 12
- Web page 81
- WebSphere MQ client 3
- WebSphere MQ Client CD-ROM 35
- WebSphere MQ environment
 - variables 143
- WebSphere MQ Script (MQSC)
 - commands 129
- WebSphere MQ server
 - platform support 11
- WebSphere MQ Server CD-ROM 35
- WebSphere MQ utility program (CSQUTIL)
 - creating a client channel definition file 135

- WebSphere MQ utility program (CSQUTIL) *(continued)*
 - creating CRL information in a client channel definition file 141
- WIN-OS/2 84, 87
- Windows 3.1 client
 - AMQSGETW sample 119
 - AMQSPUTW sample 118
 - application control 152
 - call in progress 152
 - cancelling an MQI call 152
 - design considerations 152
 - error messages 172
 - hardware and software 33
 - multi-tasking 152
- Windows 95 client
 - error messages 172
 - hardware and software 34
 - process definition 157
 - trace 174
- Windows client
 - error messages 172
 - hardware and software 31
 - installing 69
 - trace 174
 - uninstalling using Add/Remove Programs 77
- WM_QUIT message 152
- WRKCLS command 104
- WRKSBS panel 103
- WRKSBSD command 104

Z

- z/OS server 81

Sending your comments to IBM

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

Feel free to comment on what you regard as specific errors or omissions, and on the accuracy, organization, subject matter, or completeness of this book.

Please limit your comments to the information in this book and the way in which the information is presented.

To make comments about the functions of IBM products or systems, talk to your IBM representative or to your IBM authorized remarketer.

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

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

- By mail, to this address:
User Technologies Department (MP095)
IBM United Kingdom Laboratories
Hursley Park
WINCHESTER,
Hampshire
SO21 2JN
United Kingdom
- By fax:
 - From outside the U.K., after your international access code use 44-1962-816151
 - From within the U.K., use 01962-816151
- Electronically, use the appropriate network ID:
 - IBM Mail Exchange: GBIBM2Q9 at IBMMAIL
 - IBMLink[™]: HURSLEY(IDRCF)
 - Internet: idrcf@hursley.ibm.com

Whichever method you use, ensure that you include:

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



Printed in U.S.A.

GC34-6058-01

