

WebSphere MQ



Bibliography and Glossary

Note!

Before using this information and the product it supports, be sure to read the general information under “Notices”, on page 19.

Third edition (December 2002)

This is the third edition of this book that applies to WebSphere MQ. It applies to all WebSphere MQ and MQSeries products, and to all subsequent releases and modifications until otherwise indicated in new editions.

© **Copyright International Business Machines Corporation 2002. All rights reserved.**

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

Contents

About this book	v
Who this book is for.	v
WebSphere MQ and MQSeries	v
Terms used in this book	v
Changes for this edition (SC34-6113-02)	vii
Changes for the previous editions (SC34-6113-00 and -01)	ix
Bibliography.	1
WebSphere MQ cross-platform publications	1
WebSphere MQ Script (MQSC) Command Reference	1
WebSphere MQ Messages	1
WebSphere MQ Clients	2
WebSphere MQ Intercommunication	3
WebSphere MQ Programmable Command Formats and Administration Interface	4
WebSphere MQ Queue Manager Clusters.	4
WebSphere MQ Application Programming Reference	5
WebSphere MQ Application Programming Guide	6
WebSphere MQ Application Messaging Interface	6
WebSphere MQ Using Java	7
WebSphere MQ Using C++	7
WebSphere MQ System Administration Guide	8
WebSphere MQ Event Monitoring	8
WebSphere MQ Security	9
WebSphere MQ Extended Transactional Clients.	9
An Introduction to Messaging and Queuing	10
WebSphere MQ platform-specific publications.	10
WebSphere MQ for AIX, V5.3	10
WebSphere MQ for HP-UX, V5.3	10
WebSphere MQ for iSeries, V5.3	10
WebSphere MQ for Linux for Intel and Linux for zSeries, V5.3	11
WebSphere MQ for Solaris, V5.3	11
WebSphere MQ for Windows, V5.3	12
WebSphere MQ for z/OS, V5.3	12
MQSeries for AT&T GIS UNIX, V2.2	13
MQSeries for Compaq NonStop Kernel, V5.1	14
MQSeries for Compaq OpenVMS Alpha, V5.1	14
MQSeries for Compaq (DIGITAL) OpenVMS VAX.	15
MQSeries for Compaq Tru64 UNIX, V5.1	15
MQSeries for OS/2 Warp, V5.1	15
MQSeries link for R/3, Version 1.2.	15
MQSeries for SINIX and DC/OSx, V2.2	16
MQSeries for Sun Solaris, Intel Platform Edition, V5.1	16
MQSeries for VSE/ESA, V2.1.1	16
Softcopy books	17
HTML format	17
Portable Document Format (PDF)	17
Compiled HTML help format (Windows only).	17
WebSphere MQ information available on the Internet.	18
Appendix. Notices.	19
Trademarks	21
Glossary of terms and abbreviations	23
Sending your comments to IBM	45

About this book

This book provides a *one stop shop* for all your queries about WebSphere™ MQ information sources and terminology. Collecting all information previously split across the whole MQSeries® information set, this book provides a single, definitive source of:

- All information available, in any format, for any WebSphere MQ platform
- All terminology used in all members of the WebSphere MQ family

Who this book is for

This book is for anyone who wants to know what information is available for WebSphere MQ, or who wants to find the meaning of a term used in WebSphere MQ.

WebSphere MQ and MQSeries

At Version 5 Release 3, we have rebranded MQSeries to show its close relationship with the IBM® WebSphere brand of products that enable e-business. This book describes the libraries and terminology for both the Version 5.3 (WebSphere MQ) products, and products earlier than Version 5.3 (MQSeries).

In general, we use the term WebSphere MQ to refer to both levels of the product, except where we need to distinguish clearly between the two, such as in book titles.

Terms used in this book

In this book, the term **WebSphere MQ for UNIX® systems** means:

- WebSphere MQ for AIX, V5.3
- WebSphere MQ for HP-UX, V5.3
- WebSphere MQ for Linux for Intel and Linux for zSeries, V5.3
- WebSphere MQ for Solaris, V5.3

We also use the term **UNIX systems** as a general terms for the UNIX platforms.

The term **WebSphere MQ for Windows systems** means WebSphere MQ running on the Microsoft® Windows platforms:

- Windows NT®
- Windows 2000
- Windows XP

We also use the term **Windows systems** or just **Windows** as general terms for these Windows platforms.

Changes for this edition (SC34-6113-02)

This edition provides additions and clarifications for users of Version 5.1 of MQSeries for Compaq NonStop Kernel, MQSeries for Compaq OpenVMS Alpha, and MQSeries for Compaq Tru64 UNIX.

Changes for the previous editions (SC34-6113-00 and -01)

The first two editions for WebSphere MQ included the following major changes:

- Changes throughout the book to reflect the rebranding of MQSeries to WebSphere MQ.
- The addition of glossary information associated with features introduced with WebSphere MQ.
- The addition of descriptions of each publication to the Bibliography
- Some minor corrections and additions to the Glossary

Bibliography

This section describes the documentation available for all current WebSphere MQ products.

WebSphere MQ cross-platform publications

For each WebSphere MQ cross-platform publication we show the title and order number, followed by a brief description of the content of the publication and the intended audience, to help you decide whether you need that publication. We also define the members of the WebSphere MQ and MQSeries family to which the publication applies.

WebSphere MQ Script (MQSC) Command Reference

WebSphere MQ Script (MQSC) Command Reference, SC34-6055, describes the MQSC commands, used by system operators and administrators to manage queue managers. It introduces the commands and tells you how to use them, before describing the commands in detail, in alphabetic order.

This book is intended for system programmers, system administrators, and system operators. To understand this book, you need to be familiar with the system facilities for the platform on which you are using WebSphere MQ product, and to understand the basic concepts of messaging and queuing.

This book applies to the following WebSphere MQ 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[™]

It also applies to the following MQSeries products:

- MQSeries for AT&T GIS (NCR) 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

WebSphere MQ Messages

WebSphere MQ Messages, GC34-6057, describes the user messages returned by WebSphere MQ, with explanations and suggested actions. It is designed for use as a quick reference.

Bibliography

This book is for system operators, system programmers, and anyone who needs to understand and take action in response to WebSphere MQ user messages.

This book applies to the following WebSphere MQ 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

It also applies to the following MQSeries products:

- MQSeries for AT&T GIS (NCR) UNIX V2.2.1
- MQSeries for Compaq NonStop Kernel V5.1
- MQSeries for Compaq OpenVMS Alpha 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

WebSphere MQ Clients

WebSphere MQ Clients, GC34-6058, describes the WebSphere MQ client/server environment. It describes how to install WebSphere MQ clients on different platforms, how to configure communications for different protocols, and how to define WebSphere MQ channels for client and server connections, in step-by-step instructions, complete with examples. It goes on to show you how to use WebSphere MQ applications in a WebSphere MQ client/server environment, including how WebSphere MQ applications connect to a queue manager.

This book is for anyone who installs and configures WebSphere MQ clients and WebSphere MQ servers, for system administrators, and for application programmers who write programs that use the Message Queue Interface (MQI). To understand this book, you should have experience in installing and configuring the system you use for the server, experience with any client platforms that you will be using, an understanding of the purpose of the Message Queue Interface (MQI), and experience of WebSphere MQ programs in general, or familiarity with the content of the other WebSphere MQ publications.

This book applies to the following WebSphere MQ 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

It also applies to the following MQSeries products:

- MQSeries for AT&T GIS (NCR) 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

WebSphere MQ Intercommunication

WebSphere MQ Intercommunication, SC34-6059, describes intercommunication between WebSphere MQ products. It introduces the concepts of intercommunication (transmission queues, message channel agent programs, and communication links) that are brought together to form message channels. It describes how geographically-separated queue managers are linked together by message channels to form a network of queue managers. It discusses the distributed-queuing management (DQM) facility of WebSphere MQ, which provides the services that enable applications to communicate using queue managers.

This book is for anyone who needs to use WebSphere MQ intercommunication facilities including:

- Network planners responsible for designing the overall queue manager network
- Local channel planners responsible for implementing the network plan on one node
- Application programmers responsible for designing applications that include processes, queues, and channels
- Systems administrators responsible for monitoring the local system, controlling exception situations, and implementing some of the planning details
- System programmers responsible for designing and programming user exits

To use and control DQM you need a good knowledge of WebSphere MQ in general. You also need to understand the WebSphere MQ products for the platforms you will be using, and the communications protocols used on those platforms.

This book applies to the following WebSphere MQ 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

It also applies to the following MQSeries products:

- MQSeries for AT&T GIS (NCR) UNIX V2.2.1
- MQSeries for Compaq Tru64 UNIX V5.1
- MQSeries for Compaq NonStop Kernel V5.1

Bibliography

- MQSeries for Compaq OpenVMS Alpha 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

WebSphere MQ Programmable Command Formats and Administration Interface

The first part of *WebSphere MQ Programmable Command Formats and Administration Interface*, SC34-6060, describes the facilities available on WebSphere MQ products for writing programs using the Programmable Command Formats (PCFs) to administer WebSphere MQ systems either locally or remotely. The second part of this book describes the administration interface for WebSphere MQ, known as the WebSphere MQ Administration Interface (MQAI). The MQAI is a programming interface that simplifies the use of PCF messages to configure WebSphere MQ.

This book is for system programmers who write programs to monitor and administer WebSphere MQ products. To understand the part about PCFs, you need experience in writing systems management applications, an understanding of the Message Queue Interface (MQI), and experience of WebSphere MQ programs in general, or familiarity with the content of the other books in the WebSphere MQ library. To understand the part about the MQAI, you need to understand the general concepts of WebSphere MQ and how to write programs in the C programming language or in Visual Basic for Windows.

This book applies to the following WebSphere MQ 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

It also applies to the following MQSeries products :

- MQSeries for OS/2 Warp V5.1
- MQSeries for Compaq NonStop Kernel V5.1
- MQSeries for Compaq OpenVMS Alpha V5.1
- MQSeries for Compaq Tru64 UNIX V5.1
- MQSeries for Sun Solaris, Intel Platform Edition V5.1

WebSphere MQ Queue Manager Clusters

WebSphere MQ Queue Manager Clusters, SC34-6061, tells you how to create and use clusters of WebSphere MQ queue managers. It explains the concepts and terminology of clustering and shows how you can benefit by taking advantage of clustering. It describes enhancements to the message queue interface (MQI) to support clusters, and shows a number of examples of tasks you can perform to set up and maintain clusters of queue managers.

This book is for anyone who wants to use WebSphere MQ clusters, including:

- Network planners responsible for designing the overall queue manager network

- Application programmers responsible for designing applications that access queues and queue managers within clusters
- Systems administrators responsible for monitoring the local system and implementing some of the planning details
- System programmers responsible for designing and programming user exits

To understand this book, you need to understand the basic concepts of message queuing, for example the purpose of queues, queue managers, and channels. To understand fully how to make the best use of clusters, you need to be familiar with the WebSphere MQ products for the platforms you will be using, and the communications protocols used on those platforms. It is also helpful to understand how distributed queue management works.

This book applies to the following WebSphere MQ 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

It also applies to the following MQSeries products:

- MQSeries for OS/2 Warp V5.1
- MQSeries for Compaq Tru64 UNIX V5.1
- MQSeries for Compaq NonStop Kernel V5.1
- MQSeries for Compaq OpenVMS Alpha V5.1
- MQSeries for Sun Solaris, Intel Platform Edition, V5.1

WebSphere MQ Application Programming Reference

WebSphere MQ Application Programming Reference, SC34-6062, introduces the concepts of messages and queues, and gives a full description of the WebSphere MQ programming interface, including data types, function calls, attributes, return codes, and constants.

This book is for the designers of applications that will use message queuing techniques, and for programmers who have to implement these designs. To write message queuing applications using WebSphere MQ, you need to know how to write programs in at least one of the programming languages that WebSphere MQ supports. However, to understand this book, you do not need to have written message queuing programs before.

This book applies to the following WebSphere MQ 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

Bibliography

- WebSphere MQ for z/OS

It also applies to the following MQSeries products:

- MQSeries for AT&T GIS (NCR) UNIX V2.2.1
- MQSeries for Compaq Tru64 UNIX V5.1
- MQSeries for Compaq NonStop Kernel V5.1
- MQSeries for Compaq OpenVMS Alpha 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

WebSphere MQ Application Programming Guide

WebSphere MQ Application Programming Guide, SC34-6064, introduces the concepts of messages and queues, and shows you how to design, write, and build applications that use the services that WebSphere MQ provides.

This book is for the designers of applications that will use message queuing techniques, and for programmers who have to implement these designs. To write message queuing applications using WebSphere MQ, you need to know how to write programs in at least one of the programming languages that WebSphere MQ supports. However, to understand this book, you do not need to have written message queuing programs before.

This book applies to the following WebSphere MQ 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

It also applies to the following MQSeries products:

- MQSeries for AT&T GIS (NCR) UNIX V2.2.1
- MQSeries for Compaq Tru64 UNIX V5.1
- MQSeries for Compaq NonStop Kernel V5.1
- MQSeries for Compaq OpenVMS Alpha 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

WebSphere MQ Application Messaging Interface

WebSphere MQ Application Messaging Interface, SC34-6065, describes how to use the WebSphere MQ Application Messaging Interface (AMI), which provides a simple interface for those writing WebSphere MQ applications. It shows you how to use the AMI to send and receive WebSphere MQ messages, including designing publish/subscribe and point-to-point applications.

This book is for anyone who wants to use the AMI. You need to know how to write C, COBOL, C++, or Java™ programs, but you do not need previous experience of WebSphere MQ. If you are a systems administrator responsible for setting up an installation of the AMI, you need to be experienced in using the WebSphere MQ Message Queue Interface.

This book applies to the following WebSphere MQ products:

- WebSphere MQ for AIX
- WebSphere MQ for HP-UX
- WebSphere MQ for iSeries
- WebSphere MQ for Solaris
- WebSphere MQ for Windows

WebSphere MQ Using Java

WebSphere MQ Using Java, SC34-6066, tells you how to install, and write programs with, the WebSphere MQ classes for Java to access WebSphere MQ systems, and the WebSphere MQ classes for Java Message Service to access both Java Message Service (JMS) and WebSphere MQ applications

This book is for programmers who are familiar with the WebSphere MQ application programming interface; it shows them how to transfer this knowledge to become productive with the WebSphere MQ Java programming interfaces. To understand this book, you need to understand the Java programming language and the purpose of the Message Queue Interface (MQI), and have some experience of WebSphere MQ programs in general, or familiarity with the content of the other WebSphere MQ publications.

This book applies to IBM WebSphere MQ classes for Java Version 5.3 and WebSphere MQ classes for Java Message Service Version 5.3.

WebSphere MQ Using C++

WebSphere MQ Using C++, SC34-6067, describes the C++ programming-language binding to the WebSphere MQ Message Queue Interface (MQI). It introduces the binding, describes considerations associated with using C++ with WebSphere MQ, and describes the WebSphere MQ C++ classes.

The book is for application programmers who write C++ programs that use the MQI. To understand this book, you need to know the C and C++ programming languages, understand the Booch methodology, understand the purpose of the Message Queue Interface (MQI), and have experience of WebSphere MQ programs in general, or familiarity with the content of other WebSphere MQ publications

This book applies to the following WebSphere MQ 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

Bibliography

It also applies to the following MQSeries products:

- MQSeries for OS/2 Warp V5.1
- MQSeries for Compaq Tru64 UNIX V5.1
- MQSeries for Compaq NonStop Kernel V5.1
- MQSeries for Compaq OpenVMS Alpha V5.1
- MQSeries for Sun Solaris, Intel Platform Edition, V5.1

WebSphere MQ System Administration Guide

WebSphere MQ System Administration Guide, SC34-6068, describes the day-to-day management of local and remote WebSphere MQ objects. It includes topics such as security, recovery and restart, problem determination, and the dead-letter queue handler. It also includes the syntax of the WebSphere MQ control commands and tells you how to use installable services and exits to tailor your WebSphere MQ system.

This book is for system administrators and system programmers who manage the configuration and administration tasks for WebSphere MQ. It is also useful to application programmers who need to understand WebSphere MQ administration tasks. To understand this book, you need a good understanding of the operating systems it describes and of the utilities associated with them. You do not need to have worked with message queuing products before, but you should understand the basic concepts of message queuing.

This book applies to the following WebSphere MQ products:

- WebSphere MQ for AIX, V5.3
- WebSphere MQ for HP-UX, V5.3
- WebSphere MQ for Linux for Intel and Linux for zSeries, V5.3
- WebSphere MQ for Solaris, V5.3
- WebSphere MQ for Windows, V5.3

WebSphere MQ Event Monitoring

WebSphere MQ Event Monitoring, SC34-6069, describes the facilities available in WebSphere MQ products to monitor instrumentation events in a network of connected systems that use WebSphere MQ products in different operating system environments.

This book is for system programmers who write programs to monitor and administer WebSphere MQ products. To understand it, you need experience in writing systems management applications, an understanding of the Message Queue Interface (MQI), and experience of WebSphere MQ programs in general, or familiarity with the content of the other books in the WebSphere MQ library.

This book applies to the following WebSphere MQ 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

It also applies to the following MQSeries products:

- MQSeries for AT&T GIS (NCR) UNIX V2.2.1
- MQSeries for Compaq Tru64 UNIX V5.1
- MQSeries for Compaq NonStop Kernel V5.1
- MQSeries for Compaq OpenVMS Alpha 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

WebSphere MQ Security

WebSphere MQ Security, SC34-6079, describes the factors you need to consider when planning to meet your security requirements in a WebSphere MQ environment. It provides the background information for you to evaluate the security provisions offered by WebSphere MQ and related products. This book also describes the Secure Sockets Layer (SSL) support in WebSphere MQ.

This book is for anyone responsible for planning or implementing security provisions to protect WebSphere MQ objects. To understand this book, you do not need to have worked with message queuing products before, but you should understand the basic concepts of message queuing.

This book applies to the following WebSphere MQ products:

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

WebSphere MQ Extended Transactional Clients

WebSphere MQ Extended Transactional Clients, SC34-6275, describes extended transactional clients, WebSphere MQ clients that allow a client application, within the same unit of work:

- To put messages to, and get messages from, queues that are owned by the queue manager to which it is connected
- To update the resources of a resource manager other than a WebSphere MQ queue manager

This book is for anyone responsible for enabling and using extended transactional clients. To understand this book, you need a good knowledge of the concepts and terminology associated with WebSphere MQ and practical experience in implementing WebSphere MQ, particularly in the following areas:

- WebSphere MQ clients
- Using an external transaction manager to coordinate updates to WebSphere MQ resources and those of other resource managers

Bibliography

You also need to be familiar with the operating systems and communications protocols you are using.

This book applies to the following WebSphere MQ products:

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

An Introduction to Messaging and Queuing

An Introduction to Messaging and Queuing, GC33-0805, defines the problems solved by MQSeries products, explains how messaging and queuing works, examines its chief benefits, provides a small amount of technical detail about messages and message queues, and examines some typical *real world* uses.

This book is for anyone new to the subject of messaging and queuing.

This book applies to all MQSeries products.

WebSphere MQ platform-specific publications

Each WebSphere MQ product is documented in at least one platform-specific publication, in addition to the WebSphere MQ family books. This section describes those publications, organized by platform. For each platform-specific publication, we give the title and order number, followed by a brief description of the content of the publication and the intended audience, to help you decide whether you need that publication.

WebSphere MQ for AIX, V5.3

The only publication specific to the AIX platform is *WebSphere MQ for AIX, V5.3 Quick Beginnings*, GC34-6076, which tells you how to plan for WebSphere MQ for AIX, and then how to install it and verify that the installation has worked.

This book is for anyone responsible for installing WebSphere MQ for AIX. To understand this book you need a general understanding of the basic concepts of WebSphere MQ.

WebSphere MQ for HP-UX, V5.3

The only publication specific to the HP-UX platform is *WebSphere MQ for HP-UX, V5.3 Quick Beginnings*, GC34-6077, which tells you how to plan for WebSphere MQ for HP-UX, and then how to install it and verify that the installation has worked.

This book is for anyone responsible for installing WebSphere MQ for HP-UX. To understand this book you need a general understanding of the basic concepts of WebSphere MQ.

WebSphere MQ for iSeries, V5.3

Publications specific to the OS/400® platform are as follows:

WebSphere MQ for iSeries V5.3 Quick Beginnings

WebSphere MQ for iSeries V5.3 Quick Beginnings, GC34-6072, tells you how to plan for WebSphere MQ for iSeries, and then how to install it and verify that the installation has worked.

This book is for anyone responsible for installing WebSphere MQ for iSeries. To understand this book you need a general understanding of the basic concepts of WebSphere MQ.

WebSphere MQ for iSeries V5.3 System Administration Guide

WebSphere MQ for iSeries V5.3 System Administration Guide, SC34-6070, describes the system administration aspects of WebSphere MQ for iSeries, and the services provided to support commercial messaging. This includes managing the queues that applications use to receive their messages, and ensuring that applications have access to the queues that they require.

This book is for system administrators and system programmers who manage the configuration and administration tasks for WebSphere MQ. It is also useful to application programmers who must have some understanding of WebSphere MQ administration tasks. To use this book, you need a good understanding of the IBM operating system for the iSeries system, and of the utilities associated with it. You do not need to have worked with message queuing products before, but you need to understand the basic concepts of message queuing.

WebSphere MQ for iSeries V5.3 Application Programming Reference (ILE RPG)

WebSphere MQ for iSeries V5.3 Application Programming Reference (ILE RPG), SC34-6071, describes in full the WebSphere MQ for iSeries programming interface in the RPG-ILE programming language. It tells you how to build an executable application, and describes the sample programs available to help you write your RPG programs.

This book is for designers of applications that use message queuing techniques, and for programmers who implement their designs. To write message queuing applications using WebSphere MQ for iSeries, you need to know how to write programs in the RPG programming language. To use this book, you do not need to have written message queuing programs before.

WebSphere MQ for Linux for Intel and Linux for zSeries, V5.3

The only publication specific to the Linux platform is *WebSphere MQ for Linux for Intel and Linux for zSeries, V5.3 Quick Beginnings*, GC34-6078, which tells you how to plan for WebSphere MQ for Linux, and then how to install it and verify that the installation has worked.

This book is for anyone responsible for installing WebSphere MQ for Linux. To understand this book you need a general understanding of the basic concepts of WebSphere MQ.

WebSphere MQ for Solaris, V5.3

The only publication specific to the Solaris platform is *WebSphere MQ for Solaris, V5.3 Quick Beginnings*, GC34-6075, which tells you how to plan for WebSphere MQ for Solaris, and then how to install it and verify that the installation has worked.

This book is for anyone responsible for installing WebSphere MQ for Solaris. To understand this book you need a general understanding of the basic concepts of WebSphere MQ.

Bibliography

WebSphere MQ for Windows, V5.3

Publications specific to the Windows platforms are described below. WebSphere MQ for Windows means WebSphere MQ running on Windows NT, Windows 2000, or Windows XP.

WebSphere MQ for Windows V5.3 Quick Beginnings

WebSphere MQ for Windows, V5.3 Quick Beginnings, GC34-6073, tells you how to plan for WebSphere MQ for Windows, and then how to install it and verify that the installation has worked. It contains information about both the server and client features of WebSphere MQ for Windows.

This book is for anyone responsible for installing WebSphere MQ for Windows. To understand this book you need a general understanding of the basic concepts of WebSphere MQ.

WebSphere MQ for Windows Using the Component Object Model Interface

WebSphere MQ for Windows, V5.3 Using the Component Object Model Interface, SC34-5387, describes the WebSphere MQ Automation Classes for ActiveX. It tells you how to design and program your applications using the WebSphere MQ ActiveX components, and how to resolve problems using trace and reason codes. It describes in detail each of the automation classes, the ActiveX interface to the WebSphere MQ Administration interface, and the support provided by WebSphere MQ for the Microsoft Active Directory Service Interfaces (ADSI).

This book is for designers and programmers who want to use the WebSphere MQ component interfaces to develop WebSphere MQ applications that run under Windows applications, using ActiveX components. To understand this book, you need some experience of using ActiveX components and some experience or knowledge of WebSphere MQ.

WebSphere MQ for z/OS, V5.3

Publications specific to the z/OS platform are as follows:

WebSphere MQ for z/OS Concepts and Planning Guide

WebSphere MQ for z/OS Concepts and Planning Guide, GC34-6051, describes the concepts of WebSphere MQ for z/OS and tells you how to plan your WebSphere MQ for z/OS systems.

This book is for planners of z/OS systems that will use WebSphere MQ message queuing techniques and system programmers who have to install, customize, and operate WebSphere MQ for z/OS. To understand this book, you need to be familiar with the basic concepts of CICS[®], IMS[™], and WebSphere MQ. If you are going to use queue-sharing groups, you will also need to know DB2[®] and the zSeries Coupling Facility.

WebSphere MQ for z/OS System Setup Guide

WebSphere MQ for z/OS System Setup Guide, SC34-6052, tells you how to customize WebSphere MQ for z/OS after you have installed it. It also tells you how to monitor system use and performance, and how to set up security.

This book is for system programmers, system administrators, and security administrators. To understand this book, you need to be familiar with the basic concepts of CICS, IMS, the z/OS job control language (JCL), and the z/OS Time Sharing Option (TSO). If you intend to use queue-sharing groups, you also need to know DB2 and the z/OS Coupling Facility.

WebSphere MQ for z/OS System Administration Guide

WebSphere MQ for z/OS System Administration Guide, SC34-6053, tells you how to operate WebSphere MQ for z/OS using commands, panels, and utilities, and how to write applications to administer WebSphere MQ. The latter part of the book deals with termination, recovery, and restart.

This book is for system programmers and system administrators. To understand this book, you need to be familiar with the basic concepts of CICS, IMS, the z/OS job control language (JCL), and the z/OS Time Sharing Option (TSO). If you intend to use queue-sharing groups, you also need to know DB2 and the z/OS Coupling Facility. If you want to write programs to administer WebSphere MQ, you need to know how to write programs in one of the supported languages: COBOL, C, C++, Assembler, or PL/I. You do not need to have written message-queuing programs previously.

WebSphere MQ for z/OS Problem Determination Guide

WebSphere MQ for z/OS Problem Determination Guide, GC34-6054, helps you to determine the causes of WebSphere MQ for z/OS problems, resolve those problems, deal with the IBM support center, and handle APARs.

This book is for those responsible for solving problems with WebSphere MQ for z/OS systems and application programs. To understand this book, you need to be familiar with system programming concepts, z/OS diagnostic procedures, and the structure and function of the WebSphere MQ for z/OS subsystems at your site. You should also be familiar with the other systems used with WebSphere MQ for z/OS at your site, for example, CICS and IMS.

WebSphere MQ for z/OS Messages and Codes

WebSphere MQ for z/OS Messages and Codes, GC34-6056, lists all the user messages and abend reason codes returned by WebSphere MQ for z/OS, with explanations and suggested responses. It is designed for use as a quick reference, and is linked with the WebSphere MQ for z/OS Problem Determination Guide, which you should also consult if a message indicates that there is a WebSphere MQ problem.

This book is for system operators, system programmers, and anybody else who needs to understand and respond to WebSphere MQ user messages. To understand this book, you need to understand the types of message WebSphere MQ produces, the different places to which it sends these messages, and the different audiences they are intended to reach.

WebSphere MQ for z/OS Licensed Program Specifications

WebSphere MQ for z/OS Licensed Program Specifications, GC34-5893, provides a summary of the function available in WebSphere MQ for z/OS, together with detailed information about hardware and software requirements and the license under which the product is available.

WebSphere MQ for z/OS Program Directory

WebSphere MQ for z/OS Program Directory describes the material and procedures associated with installing WebSphere MQ for z/OS. It is for system programmers responsible for installing and maintaining WebSphere MQ for z/OS.

MQSeries for AT&T GIS UNIX, V2.2

The only publication specific to the AT&T GIS UNIX platform is *MQSeries for AT&T GIS UNIX System Management Guide*, SC33-1642, which describes the system administration aspects of MQSeries for AT&T GIS UNIX and the services it

Bibliography

provides to support commercial messaging in an AT&T environment. This includes managing the queues that applications use to receive their messages, and ensuring that applications have access to the queues that they require.

This book is for system administrators, and system programmers who manage the configuration and administration tasks for MQSeries. It is also useful to application programmers who must have some understanding of MQSeries administration tasks. To use this book, you need a good understanding of the AT&T operating system, and of the utilities associated with it. You do not need to have worked with message queuing products before, but you need to understand the basic concepts of message queuing.

MQSeries for Compaq NonStop Kernel, V5.1

Publications specific to the Compaq NonStop Kernel (NSK) platform are as follows:

MQSeries for Compaq NSK Quick Beginnings

MQSeries for Compaq NonStop Kernel Quick Beginnings, GC34-5887, tells you how to plan for MQSeries for Compaq NonStop Kernel, and then how to install it and verify that the installation has worked.

This book is for anyone responsible for installing MQSeries for Compaq NonStop Kernel. To understand this book you need a general understanding of the basic concepts of MQSeries.

MQSeries for Compaq NSK System Administration

MQSeries for Compaq NonStop Kernel System Administration, SC34-5886, describes the system administration aspects of MQSeries for Compaq NSK and the services it provides to support commercial messaging in a Compaq NSK environment. This includes managing the queues that applications use to receive their messages, and ensuring that applications have access to the queues that they require.

This book is for system administrators, and system programmers who manage the configuration and administration tasks for MQSeries. It is also useful to application programmers who must have some understanding of MQSeries administration tasks. To use this book, you need a good understanding of the Compaq NSK operating system and associated utilities. You do not need to have worked with message queuing products before, but you need to understand the basic concepts of message queuing.

MQSeries for Compaq OpenVMS Alpha, V5.1

Publications specific to the Compaq OpenVMS Alpha platform are as follows:

MQSeries for Compaq OpenVMS Alpha Quick Beginnings

MQSeries for Compaq OpenVMS Alpha, V5.1 Quick Beginnings, GC34-5885, tells you how to plan for MQSeries for Compaq OpenVMS Alpha, and then how to install it and verify that the installation has worked.

This book is for anyone responsible for installing MQSeries for Compaq OpenVMS Alpha. To understand this book you need a general understanding of the basic concepts of MQSeries.

MQSeries for Compaq OpenVMS Alpha System Administration

MQSeries for Compaq OpenVMS Alpha, V5.1 System Administration, SC34-5884, describes the system administration aspects of MQSeries for Compaq OpenVMS Alpha, and the services it provides to support commercial messaging in an OpenVMS environment. This includes

managing the queues that applications use to receive their messages, and ensuring that applications have access to the queues that they require.

This book is for system administrators, and system programmers who manage the configuration and administration tasks for MQSeries. It is also useful to application programmers who need to understand MQSeries administration tasks. To use this book, you need a good understanding of the OpenVMS operating system and its associated utilities. You do not need to have worked with message queuing products before, but you need to understand the basic concepts of message queuing.

MQSeries for Compaq (DIGITAL) OpenVMS VAX

The only publication specific to the Compaq (DIGITAL) OpenVMS VAX platform is *MQSeries for Compaq (DIGITAL) OpenVMS System Management Guide*, GC33-1791, which describes the system administration aspects of MQSeries for Compaq (DIGITAL) OpenVMS and the services it provides to support commercial messaging in an OpenVMS environment. This includes managing the queues that applications use to receive their messages, and ensuring that applications have access to the queues that they require.

This book is for system administrators, and system programmers who manage the configuration and administration tasks for MQSeries. It is also useful to application programmers who need to understand MQSeries administration tasks. To use this book, you need a good understanding of the OpenVMS operating system, and of the utilities associated with it. You do not need to have worked with message queuing products before, but you need to understand the basic concepts of message queuing.

MQSeries for Compaq Tru64 UNIX, V5.1

The only publication specific to the Compaq Tru64 UNIX platform is *MQSeries for Compaq Tru64 UNIX, V5.1 Quick Beginnings*, GC34-5684, which tells you how to plan for MQSeries for Compaq Tru64 UNIX, and then how to install it and verify that the installation has worked.

This book is for anyone responsible for installing MQSeries for Compaq Tru64 UNIX. To understand this book you need a general understanding of the basic concepts of MQSeries.

MQSeries for OS/2 Warp, V5.1

The only publication specific to the OS/2 Warp platform is *MQSeries for OS/2 Warp, V5.1 Quick Beginnings*, GC33-1868, which tells you how to plan for MQSeries for OS/2 Warp, and then how to install it and verify that the installation has worked.

This book is for anyone responsible for installing MQSeries for OS/2 Warp. To understand this book you need a general understanding of the basic concepts of MQSeries.

MQSeries link for R/3, Version 1.2

The only publication specific to the MQSeries link for R/3 is *MQSeries link for R/3 Version 1.2 User's Guide*, GC33-1934, which describes the MQSeries link for R/3 product: what it is, how to install it on your system, and how to use it. MQSeries link for R/3 is an interface that enables you to integrate your R/3 applications

Bibliography

with applications running in other environments, including those on SAP R/3 (referred to in this book as the R/3 system) and R/2 systems.

This book is for anyone who runs business applications that use R/3 and who needs to implement commercial messaging using the MQSeries family of products. This book is of particular interest if you intend to design, program, implement, administer, maintain, or support systems that use the MQSeries link for R/3. To use this book, you need a general understanding of R/3 application systems, together with some experience or knowledge of messaging using the MQSeries family of products. To use MQSeries link for R/3, you need a good understanding of SAP, especially the Application Enabling Layer (ALE) and IDoc formats.

MQSeries for SINIX and DC/OSx, V2.2

The only publication specific to the SINIX and DC/OSx platform is *MQSeries for SINIX and DC/OSx System Management Guide*, GC33-1768, which describes the system administration aspects of MQSeries for SINIX and DC/OSx, and the services it provides to support commercial messaging in a SINIX or DC/OSx environment. This includes managing the queues that applications use to receive their messages, and ensuring that applications have access to the queues that they require.

This book is for system administrators, and system programmers who manage the configuration and administration tasks for MQSeries. It is also useful to application programmers who need to understand MQSeries administration tasks. To use this book, you need a good understanding of the SINIX or DC/OSx operating systems, and of the utilities associated with it. You do not need to have worked with message queuing products before, but you need to understand the basic concepts of message queuing.

MQSeries for Sun Solaris, Intel Platform Edition, V5.1

The only publication specific to the Sun Solaris, Intel Platform Edition platform is *MQSeries for Sun Solaris, Intel Platform Edition Quick Beginnings*, GC34-5851, which tells you how to plan for MQSeries for Sun Solaris, Intel Platform Edition, and then how to install it and verify that the installation has worked.

This book is for anyone responsible for installing MQSeries for Sun Solaris, Intel Platform Edition. To understand this book you need a general understanding of the basic concepts of MQSeries.

MQSeries for VSE/ESA, V2.1.1

The only publication specific to the VSE/ESA platform is *MQSeries for VSE/ESA System Management Guide*, GC34-5364, which describes the system administration aspects of MQSeries for VSE/ESA, and the services it provides to support commercial messaging in a VSE/ESA environment. This includes managing the queues that applications use to receive their messages, and ensuring that applications have access to the queues that they require.

This book is for system administrators, and system programmers who manage the configuration and administration tasks for MQSeries. It is also useful to application programmers who need to understand MQSeries administration tasks. To use this book, you need a good understanding of the VSE/ESA operating system, and of the utilities associated with it. You do not need to have worked with message queuing products before, but you need to understand the basic concepts of message queuing.

Softcopy books

The WebSphere MQ books are supplied in several softcopy formats.

HTML format

Relevant WebSphere MQ documentation is provided in HTML format with these products:

- WebSphere MQ for AIX, V5.3
- WebSphere MQ for HP-UX, V5.3
- WebSphere MQ for iSeries, V5.3
- WebSphere MQ for Linux for Intel and Linux for zSeries, V5.3
- WebSphere MQ for Solaris, V5.3
- WebSphere MQ for Windows, V5.3 (compiled HTML)
- WebSphere MQ for z/OS, V5.3
- MQSeries for Compaq Tru64 UNIX, V5.1
- MQSeries for OS/2 Warp, V5.1
- MQSeries for Sun Solaris, Intel Platform Edition, V5.1
- MQSeries link for R/3, V1.2

The books are also available in HTML format from the product family Web site at:

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

Portable Document Format (PDF)

You can view and print PDF files using the Adobe Acrobat Reader.

If you need to obtain the Adobe Acrobat Reader, or would like up-to-date information about the platforms on which the Acrobat Reader is supported, visit the Adobe Systems Inc. Web site at:

<http://www.adobe.com/>

PDF versions of relevant WebSphere MQ books are supplied with these products:

- WebSphere MQ for AIX, V5.3
- WebSphere MQ for HP-UX, V5.3
- WebSphere MQ for iSeries, V5.3
- WebSphere MQ for Linux for Intel and Linux for zSeries, V5.3
- WebSphere MQ for Solaris, V5.3
- WebSphere MQ for Windows, V5.3
- WebSphere MQ for z/OS, V5.3
- MQSeries for Compaq NonStop Kernel, V5.1
- MQSeries for Compaq OpenVMS Alpha, V5.1
- MQSeries for Compaq Tru64 UNIX, V5.1
- MQSeries for OS/2 Warp, V5.1
- MQSeries for Sun Solaris, Intel Platform Edition, V5.1
- MQSeries link for R/3, V1.2

PDF versions of all current books are also available from the product family Web site at:

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

Compiled HTML help format (Windows only)

We supply the WebSphere MQ library in the form of an *Information Center* in Microsoft compiled HTML help (CHM) format. You can view this format on the following Windows systems with Internet Explorer at Version 4.01 SP1 or later:

- Windows 2000

Bibliography

- | • Windows NT 4 SP 3 or later
- | • Windows 95
- | • Windows 98
- | • Windows ME

WebSphere MQ information available on the Internet

The WebSphere MQ product family Web site is at:

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

By following links from this Web site you can:

- Obtain latest information about the product family.
- Access the books in HTML and PDF formats.
- Download a WebSphere MQ SupportPac[™].

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 International Business Machines Corporation in the United States, or other countries, or both:

AIX	AS/400	CICS
DB2	IBM	IBMLink
iSeries	Integrated Language Environment	MQSeries
OS/2	OS/400	RACF
SupportPac	SP2	VSE/ESA
WebSphere	z/OS	zSeries

UNIX is a registered trademark of The Open Group in the United States and other countries.

Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

Java and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

Other company, product, and service names may be trademarks or service marks of others.

Glossary of terms and abbreviations

This glossary defines WebSphere MQ terms and abbreviations used throughout the WebSphere MQ information set. If you do not find the term you are looking for here, try the *IBM Dictionary of Computing*, New York: McGraw-Hill, 1994.

| If you believe that we have omitted a WebSphere
| MQ term from this glossary, please use any of the
| methods described in *Sending your comments to*
| *IBM* at the back of this book to tell us.

This glossary includes terms and definitions from the *American National Dictionary for Information Systems*, ANSI X3.172-1990, copyright 1990 by the American National Standards Institute (ANSI). Copies may be purchased from the American National Standards Institute, 11 West 42 Street, New York, New York 10036. Definitions are identified by the symbol (A) after the definition.

A

abend reason code. A 4-byte hexadecimal code that uniquely identifies a problem with WebSphere MQ for z/OS. A complete list of WebSphere MQ for z/OS abend reason codes and their explanations is contained in *WebSphere MQ for z/OS Messages and Codes*.

| **abstract class.** In *object oriented programming*, a class
| that can be instantiated only as a derivation.

| **access control.** The *security service* that protects critical
| resources in a system by limiting access only to
| authorized users and their applications.

| **access control list (ACL).** A set of data that is
| maintained and used by the access control security
| service. It specifies which permissions, or access rights,
| each user or group has for a system resource, such as a
| directory or a file.

| **accountability.** Holding people responsible for their
| actions.

| **ACL.** See *access control list (ACL)*.

active log. See *recovery log*.

adapter. An interface between WebSphere MQ for z/OS and TSO, IMS, CICS, or batch address spaces. An adapter is an attachment facility that enables applications to access WebSphere MQ services.

address space. The area of virtual storage available for a particular job.

address space identifier (ASID). A unique, system-assigned identifier for an address space.

administration bag. In the MQAI, a type of data bag that is created for administering WebSphere MQ by implying that it can change the order of data items, create lists, and check selectors within a message.

administrator commands. Commands used to manage WebSphere MQ objects, such as queues, processes, and namelists.

| **advanced program-to-program communication**
| **(APPC).** The general facility characterizing the LU 6.2
| architecture and its various implementations in
| products.

affinity. An association between objects that have some relationship or dependency upon each other.

alert. A message sent to a management services focal point in a network to identify a problem or an impending problem.

alert monitor. In WebSphere MQ for z/OS, a component of the CICS adapter that handles unscheduled events occurring as a result of connection requests to WebSphere MQ for z/OS.

alias queue object. A WebSphere MQ object, the name of which is an alias for a base queue defined to the local queue manager. When an application or a queue manager uses an alias queue, the alias name is resolved and the requested operation is performed on the associated base queue.

allied address space. See *ally*.

ally. A z/OS address space that is connected to WebSphere MQ for z/OS.

| **alternate user authority.** When an application opens a
| WebSphere MQ object, it can supply a user ID on the
| MQOPEN or MQPUT1 call and ask the queue manager
| to use this user ID for authority checks instead of the
| one associated with the application. If the user ID
| associated with the application has the authority to
| supply a different user ID for authority checks, the
| former user ID is said to have *alternate user authority*.

| **alternate user security.** On z/OS, the authority checks
| that are performed when an application requests
| *alternate user authority* when opening a WebSphere MQ
| object.

APAR. See *authorized program analysis report (APAR)*.

Glossary

- | **API exit.** A user-written program that monitors or
| modifies the function of an MQI call. For each MQI call
| issued by an application, the API exit is invoked before
| the *queue manager* starts to process the call and again
| after the queue manager has completed processing the
| call. The API exit can inspect and modify any of the
| parameters on the MQI call. The API exit is not
| supported on WebSphere MQ for z/OS.
- | **API-crossing exit.** A user written program that is
| similar in concept to an *API exit*. It is supported only
| by WebSphere MQ for z/OS and for CICS applications
| only.
- APPC.** See *advanced program-to-program communication (APPC)*.
- application-defined format.** In message queuing,
application data in a message, which has a meaning
defined by the user application. Contrast with *built-in
format*.
- | **application level security.** The security services that
| are invoked when an application issues an MQI call.
- application environment.** The software facilities that
are accessible by an application program. On the z/OS
platform, CICS and IMS are examples of application
environments.
- application log.** In Windows systems, a log that
records significant application events.
- application queue.** A queue used by an application.
- archive log.** See *recovery log*.
- ARM.** See *automatic restart manager (ARM)*
- ASID.** See *address space identifier (ASID)*
- | **asymmetric key cryptography.** A system of
| *cryptography* that uses two keys: a public key known to
| everyone and a private key known only to the receiver
| or sender of the message. Contrast with *symmetric key
| cryptography*. See also *public-private key*.
- asynchronous messaging.** A method of
communication between programs in which a program
places a message on a message queue, then proceeds
with its own processing without waiting for a reply to
its message. Contrast with *synchronous messaging*.
- attribute.**
1. One of a set of properties that defines the
characteristics of a WebSphere MQ object.
 - | 2. In *object oriented programming*, a property of an
| object or class that can be distinguished distinctly
| from any other properties. Attributes often describe
| state information.
- | **audit trail.** A chronological record of events within a
| computer system. An audit trail contains sufficient
| information to reconstruct and examine a given
| sequence of events. Audit trails are useful for managing
| security and for recovering lost transactions.
- | **authentication.** The *security service* that provides proof
| that a user of a computer system is genuinely who that
| person claims to be. Common mechanisms for
| implementing this service are passwords and digital
| signatures. Authentication is distinct from *authorization*;
| authentication is not concerned with granting or
| denying access to system resources.
- | **authority checks.** See *authorization checks*.
- | **authorization.** The process of granting or denying
| access to a system resource. Security systems use a two
| step process: after *authentication* has verified that users
| are who they say they are, authorization allows the
| users access to various resources based on their
| identity. Authorization is a component of the *access
| control security service*.
- | **authorization checks.** Security checks that are
| performed when a user or application attempts to
| access a system resource; for example, when an
| administrator attempts to issue a command to
| administer WebSphere MQ or when an application
| attempts to connect to a queue manager. Authorization
| checks are also known as authority checks and are
| performed as part of the *access control security service*.
- authorization file.** In WebSphere MQ on UNIX
systems, a file that provides security definitions for an
object, a class of objects, or all classes of objects.
- authorization service.** In WebSphere MQ on UNIX
systems and WebSphere MQ for Windows, a service
that provides authority checking of commands and
MQI calls for the user identifier associated with the
command or call.
- authorized program analysis report (APAR).** A report
of a problem caused by a suspected defect in a current,
unaltered release of a program.
- | **automatic restart manager (ARM).** A z/OS recovery
| function that can improve the availability of specific
| batch jobs or started tasks, and therefore result in faster
| resumption of productive work.

B

backout. An operation that reverses all the changes
made during the current unit of recovery or unit of
work. After the operation is complete, a new unit of
recovery or unit of work begins. Synonymous with
rollback. Contrast with *commit*.

bag. See *data bag*.

basic mapping support (BMS). An interface between
CICS and application programs that formats input and

output display data and routes multiple-page output messages without regard for control characters used by various terminals.

| **behavior.** In *object oriented programming*, the functionality embodied within a method.

BMS. See *basic mapping support (BMS)*.

Booch methodology. An object-oriented methodology that helps users design systems using the object-oriented paradigm.

bootstrap data set (BSDS). A VSAM data set that contains:

- An inventory of all active and archived log data sets known to WebSphere MQ for z/OS
- A wraparound inventory of all recent WebSphere MQ for z/OS activity

The BSDS is required to restart the WebSphere MQ for z/OS subsystem.

browse. In message queuing, to use the MQGET call to copy a message without removing it from the queue. See also *get*.

browse cursor. In message queuing, an indicator used when browsing a queue to identify the message that is next in sequence.

BSDS. See *bootstrap data set (BSDS)*.

buffer pool. An area of main storage used for WebSphere MQ for z/OS queues, messages, and object definitions. See also *page set*.

built-in format. In message queuing, application data in a message that has a meaning defined by the queue manager. Synonymous with *in-built format*. Contrast with *application-defined format*.

C

| **CA.** See *certification authority*.

call back. In WebSphere MQ, a requester message channel initiates a transfer from a sender channel by first calling the sender, then closing down and awaiting a call back.

CCF. See *channel control function (CCF)*.

CCSID. See *coded character set identifier (CCSID)*.

CDF. See *channel definition file (CDF)*.

| **certificate.** See *digital certificate*.

| **certificate authority.** See *certification authority*.

| **certificate revocation list (CRL).** A list of certificates that have been revoked before their scheduled

| expiration date. CRLs are maintained by the *certification authority* and used, during SSL handshaking, to ensure that the certificates involved have not been revoked.

| **certificate store.** Windows name for a *key repository*.

| **certification authority (CA).** A trusted third-party organization or company that issues the *digital certificates* that are digitally signed by the authority. A certification authority provides assurance that the individual identified in a digital certificate owns the public key contained in the certificate. A certification authority is also known as a *certificate authority*.

| **CFStruct.** A WebSphere MQ object used to describe the queue manager's use of a Coupling Facility list structure

channel. See *message channel*.

channel control function (CCF). A program to move messages from a transmission queue to a communication link, and from a communication link to a local queue, together with an operator panel interface to set up and control channels.

channel definition file (CDF). A file containing communication channel definitions that associate transmission queues with communication links.

| **channel event.** An event reporting conditions detected during channel operations, such as when a channel instance is started or stopped. Channel events are generated on the queue managers at both ends of the channel.

| **channel exit program.** A user-written program that is called from one of a defined number of places in the processing sequence of a *message channel agent (MCA)*.

channel initiator. A component of WebSphere MQ distributed queuing that monitors the initiation queue to see when triggering criteria have been met and then starts the sender channel.

channel listener. A component of WebSphere MQ distributed queuing that monitors the network for a startup request and then starts the receiving channel.

checkpoint. A time when significant information is written on the log. Contrast with *sync point*.
| Checkpoints are generated automatically and are used during the system restart process.

CI. See *control interval (CI)*

CICS transaction. In CICS, a unit of application processing, usually comprising one or more units of work.

| **CipherSpec.** The combination of encryption algorithm and hash function applied to an SSL message after authentication is complete.

Glossary

- | **CipherSuite.** The combination of authentication key exchange algorithm and CipherSpec used by SSL for secure exchange of data.
- | **ciphertext.** Data that has been encrypted. Cipher text is unreadable until it has been converted into *plaintext* (decrypted) with a key.
- circular logging.** In WebSphere MQ on UNIX systems and WebSphere MQ for Windows, the process of keeping all restart data in a ring of log files. Logging fills the first file in the ring and then moves on to the next, until all the files are full. At this point, logging goes back to the first file in the ring and starts again, if the space has been freed or is no longer needed. Circular logging is used during restart recovery, using the log to roll back transactions that were in progress when the system stopped. Contrast with *linear logging*.
- CL.** See *Command Language (CL)*.
- | **class.** In *object oriented programming*, an abstract model of behavior; a collection of methods. A class typically provides some unique behavior, in addition to other, common, behavior. The distinction between unique and common behavior is effected using either inheritance, or multiple interfaces.
- | **class hierarchy.** In *object oriented programming*, classes related by inheritance.
- | **class library.** In *object oriented programming*, a bundled collection of classes, usually related.
- client.** A runtime component that provides access to queuing services on a server for local user applications. The queues used by the applications reside on the server. See also *WebSphere MQ client*.
- client application.** An application, running on a workstation and linked to a client, that gives the application access to queuing services on a server.
- client connection channel type.** The type of MQI channel definition associated with a WebSphere MQ client. See also *server connection channel type*.
- CLUSRCVR.** See *cluster-receiver channel (CLUSRCVR)*.
- CLUSSDR.** See *cluster-sender channel (CLUSSDR)*.
- | **cluster.**
- | 1. In WebSphere MQ, a group or two or more queue managers on one or more computers, providing automatic interconnection, and allowing queues to be shared amongst them for load balancing and redundancy.
- | 2. In *Microsoft Cluster Server (MSCS)*, a group of computers, connected together and configured in such a way that, if one fails, MSCS performs a *failover*, transferring the state data of applications from the failing computer to another computer in the cluster and reinitiating their operation there.
- cluster queue.** A queue that is hosted by a cluster queue manager and made available to other queue managers in the cluster.
- cluster queue manager.** A queue manager that is a member of a cluster. A queue manager may be a member of more than one cluster.
- cluster-receiver channel (CLUSRCVR).** A channel on which a cluster queue manager can receive messages from other queue managers in the cluster, and cluster information from the repository queue managers.
- cluster-sender channel (CLUSSDR).** A channel on which a cluster queue manager can send messages to other queue managers in the cluster, and cluster information to the repository queue managers.
- cluster transmission queue.** A transmission queue that holds all messages from a queue manager destined for another queue manager that is in the same cluster. The queue is called SYSTEM.CLUSTER.TRANSMIT.QUEUE.
- | **coded character set identifier (CCSID).** The name of a coded set of characters and their code point assignments.
- command.** In WebSphere MQ, an administration instruction that can be carried out by the queue manager.
- command bag.** In the MQAI, a type of bag that is created for administering WebSphere MQ objects, but cannot change the order of data items or create lists within a message.
- | **Command Language (CL).** In WebSphere MQ for iSeries, a language that can be used to issue commands, either at the command line or by writing a CL program.
- command prefix (CPF).** In WebSphere MQ for z/OS, a character string that identifies the queue manager to which WebSphere MQ for z/OS commands are directed, and from which WebSphere MQ for z/OS operator messages are received.
- command processor.** The WebSphere MQ component that processes commands.
- command server.** The WebSphere MQ component that reads commands from the system-command input queue, verifies them, and passes valid commands to the command processor.
- commit.** An operation that applies all the changes made during the current unit of recovery or unit of work. After the operation is complete, a new unit of recovery or unit of work begins. Contrast with *backout*.
- completion code.** A return code indicating how an MQI call has ended.

| **confidentiality.** The *security service* that protects sensitive information from unauthorized disclosure. Encryption is a common mechanism for implementing this service.

| **configuration event.** Notifications about the attributes of an object. They are generated when it has been created, changed, or deleted and are also generated by explicit requests.

configuration file. In WebSphere MQ on UNIX systems and WebSphere MQ for iSeries, a file that contains configuration information for logs, communications, or installable services. Synonymous with *.ini file*. See also *stanza*.

| **connect.** The means by which an application gains access to a queue manager and its resources.

connection handle. The identifier or token by which a program accesses the queue manager to which it is connected.

| **constructor.** In *object oriented programming*, a special method used to initialize an object.

| **context.** Information about the originator of a message that is held in fields in the *message descriptor*. There are two categories of context information: identity context and origin context. The identity context fields contain information about the user of the application that put the message on the queue. The origin context fields contain information about the application itself and when the message was put on the queue. Context is also known as message context.

| **context security.** On z/OS[®], the authority checks that are performed when an application opens a queue and specifies that it will set the *context* in messages it puts on the queue, or pass the *context* from messages it has received to messages it puts on the queue.

control command. In WebSphere MQ on UNIX systems and WebSphere MQ for Windows, a command that can be entered interactively from the operating system command line. Such a command requires only that the WebSphere MQ product be installed; it does not require a special utility or program to run it.

control interval (CI). A fixed-length area of direct access storage in which VSAM stores records and creates distributed free spaces. The control interval is the unit of information that VSAM transmits to or from direct access storage.

controlled shutdown. See *quiesced shutdown*.

| **conversation.** See *LU 6.2 conversation*.

coupling facility. On z/OS, a special logical partition that provides high-speed caching, list processing, and locking functions in a parallel sysplex.

CPF. See *command prefix*.

| CRL. See *certificate revocation list (CRL)*.

cross systems coupling facility (XCF). Provides the z/OS coupling services that allow authorized programs in a multisystem environment to communicate with programs on the same or different z/OS systems.

| **cryptography.** Protecting information by transforming it (encrypting it) into an unreadable format, called *ciphertext*. Only those who possess a secret key can decipher (or decrypt) the message into *plaintext*.

D

DAE. See *dump analysis and elimination (DAE)*.

daemon. In UNIX systems, a program that runs unattended to perform a standard service. Some daemons are triggered automatically to perform their tasks; others operate periodically.

data bag. In the MQAI, a bag that allows you to handle properties (or parameters) of objects.

data conversion interface (DCI). The WebSphere MQ interface to which customer- or vendor-written programs that convert application data between different machine encodings and CCSIDs must conform. A part of the WebSphere MQ Framework.

data-conversion service. A service that converts application data to the character set and encoding required by applications on other platforms.

datagram. The simplest message that WebSphere MQ supports. This type of message does not require a reply.

| **data integrity.** The *security service* that detects whether there has been unauthorized modification of data, or *tampering*. The service only detects whether data has been modified; it does not restore data to its original state if it has been modified.

data item. In the MQAI, an item contained within a data bag. This can be an integer item or a character-string item, and a user item or a system item.

DCE. See *Distributed Computing Environment (DCE)*.

DCE principal. A user ID that uses the distributed computing environment.

DCI. See *data conversion interface (DCI)*.

| DCM. See *Digital Certificate Manager (DCM)*.

dead-letter queue (DLQ). A queue to which a queue manager or application sends messages that it cannot deliver to their correct destination.

Glossary

dead-letter queue handler. A WebSphere MQ-supplied utility that monitors a dead-letter queue (DLQ) and processes messages on the queue in accordance with a user-written rules table.

decryption. The process of decoding data that has been encrypted into a secret format. Decryption requires a secret key or password. Contrast with *encryption*.

default object. A definition of an object (for example, a queue) with all attributes defined. If a user defines an object but does not specify all possible attributes for that object, the queue manager uses default attributes in place of any that were not specified.

deferred connection. A pending event that is activated when a CICS subsystem tries to connect to WebSphere MQ for z/OS before it has started.

dequeue. To remove a message from a queue. Contrast with *enqueue*.

derivation. In *object oriented programming*, the refinement or extension of one class from another.

digital certificate. An electronic document used to identify an individual, server, company, or some other entity, and to associate a public key with the entity. A digital certificate is issued by a *certification authority* and is digitally signed by that authority. The purpose of a digital certificate is to provide assurance to a receiver of the certificate that the entity identified in the certificate owns the public key contained in the certificate.

Digital Certificate Manager (DCM). On OS/400 systems, enables you to manage *digital certificates* and use them in secure applications on the iSeries server. With Digital Certificate Manager, you can request and process digital certificates from Certification Authorities (CAs) or other third-parties. You can also act as a local *certification authority* to create and manage digital certificates for your users.

digital signature. A digital code that uniquely identifies the sender and that can be attached to a message. Like a written signature, the purpose of a digital signature is to guarantee that the individual sending the message really is who he or she claims to be. It also proves that the text of the message has not changed in transit.

Distinguished Name (DN). A set of name-value pairs (such as CN=person's name and C=country) that uniquely identify an entity in a *digital certificate*.

distributed application. In message queuing, a set of application programs that can each be connected to a different queue manager, but that collectively constitute a single application.

Distributed Computing Environment (DCE). Middleware that provides some basic services, making the development of distributed applications easier. DCE is defined by the Open Software Foundation (OSF).

distributed queue management (DQM). In message queuing, the setup and control of message channels to queue managers on other systems.

distributed queuing. See *distributed queue management (DQM)*.

distribution list. A list of queues to which a message can be put using a single MQPUT or MQPUT1 statement.

DLQ. See *dead-letter queue (DLQ)*.

DN. See *Distinguished Name*.

DQM. See *distributed queue management (DQM)*.

dual logging. A method of recording WebSphere MQ for z/OS activity, where each change is recorded on two data sets, so that if a restart is necessary and one data set is unreadable, the other can be used. Contrast with *single logging*.

dual mode. See *dual logging*.

dump analysis and elimination (DAE). A z/OS service that enables an installation to suppress SVC dumps and ABEND SYSUDUMP dumps that are not needed because they duplicate previously written dumps.

dynamic queue. A local queue created when a program opens a model queue object. See also *permanent dynamic queue* and *temporary dynamic queue*.

E

eavesdropping. A breach of communication security in which the information remains intact, but its privacy is compromised. See also *tampering* and *impersonation*.

encapsulation. In *object oriented programming*, the restriction whereby class behavior can only be observed using the methods of that class.

encryption. The translation of data into a secret code. To read an encrypted file, you must have access to a secret key or password that enables you to decrypt it. Unencrypted data is called *plaintext*; encrypted data is called *ciphertext*. Contrast with *decryption*.

end user verification. See *LUU 6.2 conversation level security*.

enqueue. To put a message on a queue. Contrast with *dequeue*.

environment. See *application environment*.

environment variable. One of a series of variables that control the way your operating system runs and what external devices it recognizes. You can define these variables in your system profile or override them temporarily with command-line commands.

ESM. See *external security manager (ESM)*.

ESTAE. See *extended specify task abnormal exit (ESTAE)*

| **event.** See *channel event, configuration event, instrumentation event, performance event, and queue manager event*.

event data. In an event message, the part of the message data that contains information about the event (such as the queue manager name, and the application that gave rise to the event). See also *event header*.

event header. In an event message, the part of the message data that identifies the event type of the reason code for the event.

event log. See *application log*.

event message. Contains information (such as the category of event, the name of the application that caused the event, and queue manager statistics) relating to the origin of an instrumentation event in a network of WebSphere MQ systems.

event queue. The queue onto which the queue manager puts an event message after it detects an event. Each category of event (queue manager, performance, or channel event) has its own event queue.

Event Viewer. A tool provided by Windows systems to examine and manage log files.

| **exclusive method.** In *object oriented programming*, a method that is not intended to exhibit polymorphism; one with specific effect.

extended specify task abnormal exit (ESTAE). A z/OS macro that provides recovery capability and gives control to the specified exit routine for processing, diagnosing an abend, or specifying a retry address.

external security manager (ESM). A security product that is invoked by the z/OS *System Authorization Facility (SAF)*. RACF® is an example of an ESM.

F

| **failover.** The operation where *Microsoft Cluster Server (MSCS)* detects a failure in an application on one computer in the cluster, and shuts down the disrupted application in an orderly manner, transfers its state data to the other computer, and re-initiates the application there.

FAP. See *Formats and Protocols (FAP)*

| **FDC.** See *First Failure Data Capture (FDC) files*.

FIFO. See *first-in-first-out (FIFO)*

| **firewall.** A means of preventing unauthorized traffic into and out of a secure network. A firewall acts as a gateway between a trusted internal network and an untrusted external network, such as the Internet. You can use a firewall to filter communications by source, content, or both. A firewall can be implemented by hardware, software, or a combination of both.

| **First Failure Data Capture (FDC) files.** Files containing product error diagnostic information (they have a suffix of .FDC).

first-in-first-out (FIFO). A queuing technique in which the next item to be retrieved is the item that has been in the queue for the longest time. (A)

forced shutdown. A type of shutdown of the CICS adapter where the adapter immediately disconnects from WebSphere MQ for z/OS, regardless of the state of any currently active tasks. Contrast with *quiesced shutdown*.

format. In message queuing, a term used to identify the nature of application data in a message. See also *built-in format* and *application-defined format*.

Formats and Protocols (FAP). The WebSphere MQ FAPs define how queue managers communicate with one another, and also how WebSphere MQ clients communicate with server queue managers.

Framework. In WebSphere MQ, a collection of programming interfaces that allow customers or vendors to write programs that extend or replace certain functions provided in WebSphere MQ products. The interfaces are:

- data conversion interface (DCI)
- message channel interface (MCI)
- name service interface (NSI)
- security enabling interface (SEI)
- trigger monitor interface (TMI)

| **friend class.** In *object oriented programming*, a class that is regarded as being derived from another, while this is not the case, for the purpose of accessing protected methods and instance data.

FRR. See *functional recovery routine (FRR)*

| **full repository.** A complete set of information about every queue manager in a cluster. This set of information is called the repository or sometimes the full repository and is usually held by two of the queue managers in the cluster. Contrast with *partial repository*.

function. A function call as supported by the C programming language.

Glossary

functional recovery routine (FRR). A z/OS recovery/termination manager that enables a recovery routine to gain control in the event of a program interrupt.

G

generalized trace facility (GTF). A z/OS service program that records significant system events, such as supervisor calls and start I/O operations, for the purpose of problem determination.

| **Generic Security Services Application Programming Interface (GSS API).** A common application programming interface (API) for accessing security services.

get. In message queuing, to use the MQGET call to remove a message from a queue.

global trace. A WebSphere MQ for z/OS trace option where the trace data comes from the entire WebSphere MQ for z/OS subsystem.

globally-defined object. On z/OS, an object whose definition is stored in the shared repository. The object is available to all queue managers in the queue-sharing group. See also *locally-defined object*.

| **GSKit.** The IBM Global Security Kit that enables you to use SSL security on UNIX and OS/400 systems.

| **GSS API.** See *Generic Security Services Application Programming Interface (GSS API)*.

GTF. See *generalized trace facility (GTF)*.

H

| **handshake.** The exchange of messages at the start of a *Secure Sockets Layer (SSL)* session that allows the client to authenticate the server using public key techniques (and, optionally, for the server to authenticate the client), then allows the client and server to cooperate in creating symmetric keys for encryption, decryption, and detection of tampering.

| **handle.** See *connection handle* and *object handle*.

hardened message. A message that is written to auxiliary (disk) storage so that the message is not lost in the event of a system failure. See also *persistent message*.

| **heartbeat.** The polling signal *Microsoft Cluster Server (MSCS)* sends and monitors over the private connections between computers in a cluster in order to know the state of each.

heartbeat flow. A pulse that is passed from a sending MCA to a receiving MCA when there are no messages to send. The pulse unblocks the receiving MCA, which

would otherwise remain in a wait state until a message arrived or the disconnect interval expired.

heartbeat interval. The time, in seconds, that is to elapse between heartbeat flows.

I

ICE. See *Intersystem Communications Environment (ICE)*

| **identification.** The *security service* that enables each user of a computer system to be identified uniquely. A common mechanism for implementing this service is to associate a user ID with each user.

IFCID. A trace event number.

ILE. See *Integrated Language Environment (ILE)*

immediate shutdown. In WebSphere MQ, a shutdown of a queue manager that does not wait for applications to disconnect. Current MQI calls are allowed to complete, but new MQI calls fail after an immediate shutdown has been requested. Contrast with *quiesced shutdown* and *preemptive shutdown*.

| **impersonation.** A breach of communication security in which the information is passed to a person posing as the intended receiver or information is sent by a person posing as someone else. See also *eavesdropping* and *tampering*.

inbound channel. A channel that receives messages from another queue manager. See also *shared inbound channel*.

in-built format. See *built-in format*.

index. In the MQAI, a means of referencing data items.

in-doubt unit of recovery. In WebSphere MQ, the status of a unit of recovery for which a syncpoint has been requested but not yet confirmed.

| **inheritance.** In *object oriented programming*, the ability of a class to include the behavior of another through refinement and extension; only refined and extended methods are defined in the derived class, thereby preserving encapsulation.

.ini file. See *configuration file*.

initialization input data sets. Data sets used by WebSphere MQ for z/OS when it starts up.

initiation queue. A local queue on which the queue manager puts trigger messages.

input parameter. A parameter of an MQI call in which you supply information when you make the call.

input/output parameter. A parameter of an MQI call in which you supply information when you make the call, and in which the queue manager changes the information when the call completes or fails.

insertion order. In the MQAI, the order that data items are placed into a data bag.

installable services. In WebSphere MQ on UNIX systems, WebSphere MQ for Windows, and MQSeries for OS/2 Warp, additional functionality provided as independent components. The installation of each component is optional: in-house or third-party components can be used instead. See also *authorization service* and *name service*.

instance. In *object oriented programming*, an object.

instance data. In *object oriented programming*, state information associated with an object.

instrumentation event. A facility that can be used to monitor the operation of queue managers in a network of WebSphere MQ systems. WebSphere MQ provides instrumentation events to monitor queue manager resource definitions, performance conditions, and channel conditions. Instrumentation events can be used by a user-written reporting mechanism in an administration application that displays the events to a system operator. They also allow applications acting as agents for other administration networks to monitor reports and create the appropriate alerts.

Integrated Language Environment® (ILE). In iSeries, a set of constructs and interfaces that provides common runtime environment and bindable application program interfaces (APIs) for all ILE-conforming high-level languages. This replaces the AS/400® Original Program Model (OPM).

interactive problem control system (IPCS). A component of z/OS that permits online problem management, interactive problem diagnosis, online debugging for disk-resident abend dumps, problem tracking, and problem reporting.

Interactive System Productivity Facility (ISPF). An IBM licensed program that serves as a full-screen editor and dialog manager. Used to write application programs, it provides a means of generating standard screen panels and interactive dialogues between the application programmer and terminal user.

interface. In *object oriented programming*, an abstract model of behavior; a collection of functions or methods.

Internet Protocol (IP). In the Internet suite of protocols, a connectionless protocol that routes data through a network or interconnected networks and acts as an intermediary between the higher protocol layers and the physical network. See also *Transmission Control Protocol (TCP)* and *Transmission Control Protocol/Internet Protocol (TCP/IP)*.

interprocess communication (IPC). In UNIX systems, the process by which programs communicate data to each other and synchronize their activities. Semaphores, signals, and internal message queues are common methods of interprocess communication.

intersystem communication. In CICS, communication between separate systems by means of SNA networking facilities or by means of the application-to-application facilities of an SNA access method.

Intersystem Communications Environment (ICE). A family of Compaq-based software products that enables you to access a variety of applications on Compaq computers.

IP. See *Internet Protocol (IP)*

IPC. See *interprocess communication (IPC)*

IPCS. See *interactive problem control system (IPCS)*

ISC. See *intersystem communication*

ISPF. See *Interactive System Productivity Facility (ISPF)*

J

Java Runtime Environment (JRE). The minimum standard Java Platform for running Java programs. It contains the Java virtual machine, Java core classes, and supporting files.

journal. WebSphere MQ for iSeries uses the journals provided in OS/400 journaling support to control updates to local objects. Each queue manager library contains a journal for that queue manager.

JRE. See *Java Runtime Environment (JRE)*.

K

Kerberos. An authentication system developed at the Massachusetts Institute of Technology (MIT). Kerberos enables two parties to exchange private information over an otherwise open network. It works by assigning a unique key, called a ticket, to each user that logs on to the network. The ticket is then embedded in messages that are sent over the network. The receiver of a message uses the ticket to authenticate the sender.

key authentication. See *authentication*.

key database file. UNIX name for a *key repository*.

key repository. A store for *digital certificates* and their associated private keys. The specific store names on WebSphere MQ platforms are:

UNIX key database file

Windows

certificate store

Glossary

| z/OS key ring

| **key ring.** z/OS name for a *key repository*.

L

| **LDAP.** See *Lightweight Directory Access Protocol (LDAP)*.

| **Lightweight Directory Access Protocol (LDAP).** A set of protocols for accessing information directories. LDAP is based on the standards contained within the X.500 standard, but supports TCP/IP, enabling it to be used with any type of Internet access.

| **linear logging.** In WebSphere MQ on UNIX systems and WebSphere MQ for Windows, the process of keeping restart data in a sequence of files. New files are added to the sequence as necessary. The space in which the data is written is not reused. Contrast with *circular logging*.

| **link level security.** The *security services* that are invoked, directly or indirectly, by a *message channel agent (MCA)*, the communications subsystem, or a combination of the two working together.

| **listener.** In WebSphere MQ distributed queuing, a program that detects incoming network requests and starts the associated channel.

| **local definition of a remote queue.** A WebSphere MQ object belonging to a local queue manager. This object defines the attributes of a queue that is owned by another queue manager. In addition, it is used for queue-manager aliasing and reply-to-queue aliasing.

| **local queue.** A queue that belongs to the local queue manager. A local queue can contain a list of messages waiting to be processed. Contrast with *remote queue*.

| **local queue manager.** The queue manager to which a program is connected and that provides message queuing services to the program. A queue manager to which a program is not connected is called a *remote queue manager*, even if it is running on the same system as the program.

| **locale.** On UNIX systems, a subset of a user's environment that defines conventions for a specific culture (such as time, numeric, or monetary formatting and character classification, collation, or conversion). The queue manager CCSID is derived from the locale of the user ID that created the queue manager.

| **locally-defined object.** On z/OS, an object whose definition is stored on page set zero. The definition can be accessed only by the queue manager that defined it. Also known as a *privately-defined object*.

| **log.** In WebSphere MQ, a file recording the work done by queue managers while they receive, transmit, and deliver messages, to enable them to recover in the event of failure.

| **log control file.** In WebSphere MQ on UNIX systems and WebSphere MQ for Windows, the file containing information needed to monitor the use of log files (for example, their size and location, and the name of the next available file).

| **log file.** In WebSphere MQ on UNIX systems and WebSphere MQ for Windows, a file in which all significant changes to the data controlled by a queue manager are recorded. If the primary log files become full, WebSphere MQ allocates secondary log files.

| **logical unit (LU).** A component of the *Systems Network Architecture (SNA)* communications subsystem (stack) that enables users to gain access to network resources and communicate with each other. A user can be an application or an individual working at a workstation.

| **log record sequence number (LRSN).** An alternative technique to RBA for addressing log records.

| **LRSN.** See *log record sequence number (LRSN)*.

| **LU-LU verification.** See *session level authentication*.

| **luname.** The SNA name of the logical unit on your workstation.

| **LUWID.** Logical unit of work identifier.

| **LU 6.2.** In SNA, a type of *logical unit (LU)* that supports general communication between programs in a distributed processing environment.

| **LU 6.2 conversation.** In SNA, a logical connection between two transaction programs over an *LU 6.2 session* that enables them to communicate with each other.

| **LU 6.2 conversation level security.** In SNA, a conversation level security protocol that enables a partner *transaction program* to authenticate the transaction program that initiated the conversation. LU 6.2 conversation level security is also known as end user verification.

| **LU 6.2 session.** In SNA, a session between two logical units (LUs) of type 6.2.

M

| **marshalling.** In *object oriented programming*, the serialization of data.

| **MCA.** See *message channel agent (MCA)*.

| **MCI.** See *message channel interface (MCI)*.

| **message context.** See *context*.

| **message exit.** A type of *channel exit program*. Message exits usually work in pairs, one at each end of a channel. At the sending end of a channel, a message

exit is called after the *message channel agent (MCA)* has got a message from the transmission queue. At the receiving end of a channel, a message exit is called before the *message channel agent (MCA)* puts a message on its destination queue. A message exit can modify the contents of a message and change its length. A message exit can be called only on a message channel.

media image. In WebSphere MQ on UNIX systems and WebSphere MQ for Windows, the sequence of log records that contain an image of an object. The object can be recreated from this image.

message. In message queuing applications, a communication sent between programs. In system programming, information intended for the terminal operator or system administrator.

message channel. In distributed message queuing, a mechanism for moving messages from one queue manager to another. A message channel comprises two message channel agents (a sender at one end and a receiver at the other end) and a communication link. Contrast with *MQI channel*.

message channel agent (MCA). A program that transmits prepared messages from a transmission queue to a communication link, or from a communication link to a destination queue. See also *message queue interface (MQI)*.

message channel interface (MCI). The WebSphere MQ interface to which customer- or vendor-written programs that transmit messages between a WebSphere MQ queue manager and another messaging system must conform. A part of the WebSphere MQ Framework.

message descriptor. Control information describing the message format and presentation that is carried as part of a WebSphere MQ message. The format of the message descriptor is defined by the MQMD structure.

message flow control. A distributed queue management task that involves setting up and maintaining message routes between queue managers.

message format service (MFS). In IMS, an editing facility that allows application programs to deal with simple logical messages, instead of device-dependent data, thus simplifying the application development process. See *message input descriptor (MID)* and *message output descriptor (MOD)*.

message group. A group of logical messages. Logical grouping of messages allows applications to group messages that are similar and to ensure the sequence of the messages.

message input descriptor (MID). In IMS, the MFS control block that describes the format of the data presented to the application program. Contrast with *message output descriptor (MOD)*.

message output descriptor (MOD). In IMS, the MFS control block that describes the format of the output data produced by the application program. Contrast with *message input descriptor (MID)*.

message priority. In WebSphere MQ, an attribute of a message that can affect the order in which messages on a queue are retrieved, and whether a trigger event is generated.

message queue. Synonym for *queue*.

message queue interface (MQI). The programming interface provided by the WebSphere MQ queue managers. This programming interface allows application programs to access message queuing services.

message queuing. A programming technique in which each program within an application communicates with the other programs by putting messages on queues.

message-retry. An option available to an MCA that is unable to put a message. The MCA can wait for a predefined amount of time and then try to put the message again.

message segment. One of a number of segments of a message that is too large either for the application or for the queue manager to handle.

message sequence numbering. A programming technique in which messages are given unique numbers during transmission over a communication link. This enables the receiving process to check whether all messages are received, to place them in a queue in the original order, and to discard duplicate messages.

messaging. See *synchronous messaging* and *asynchronous messaging*.

method. In *object oriented programming*, a means of invoking a particular behavior in an object or class.

MFS. See *message format service (MFS)*.

Microsoft Cluster Server (MSCS). A way of achieving high availability by grouping computers into an MSCS cluster. If one of the computers in the cluster encounters any one of a range of problems, MSCS shuts down the disrupted application in an orderly manner, transfers its state data to another computer in the cluster, and re-initiates the application there.

Microsoft Transaction Server (MTS). Helps Windows users run business logic applications in a typical middle tier server. MTS divides work up into activities, which are (typically) short independent chunks of business logic.

model queue object. A set of queue attributes that act as a template when a program creates a dynamic queue.

Glossary

MQAI. See *WebSphere MQ Administration Interface (MQAI)*.

MQI. See *message queue interface (MQI)*.

MQI channel. Connects a WebSphere MQ client to a queue manager on a server system, and transfers MQI calls and responses. Contrast with *message channel*.

MQSC. See *WebSphere MQ script commands (MQSC)*.

MQSeries. A previous name for *WebSphere MQ*.

MSCS. See *Microsoft Cluster Server (MSCS)*.

MTS. See *Microsoft Transaction Server (MTS)*.

multi-hop. To pass through one or more intermediate queue managers when there is no direct communication link between a source queue manager and the target queue manager.

N

namelist. A WebSphere MQ object that contains a list of names, for example, queue names.

name service. In WebSphere MQ on UNIX systems and WebSphere MQ for Windows, the facility that determines which queue manager owns a specified queue.

name service interface (NSI). The WebSphere MQ interface to which customer- or vendor-written programs that resolve queue-name ownership must conform. A part of the WebSphere MQ Framework.

name transformation. In WebSphere MQ on UNIX systems and WebSphere MQ for Windows, an internal process that changes a queue manager name so that it is unique and valid for the system being used. Externally, the queue manager name remains unchanged.

nested bag. In the MQAI, a system bag that is inserted into another data bag

nesting. In the MQAI, a means of grouping information returned from WebSphere MQ.

NetBIOS. Network Basic Input/Output System. An operating system communications interface for application programs used on IBM-compatible personal computers that are attached to the IBM Token-Ring Network.

New Technology File System (NTFS). A Windows recoverable file system that provides security for files.

node. In MSCS, the computers in the cluster.

nonpersistent message. A message that does not survive a restart of the queue manager. Contrast with *persistent message*.

non-repudiation. This *security service* applies when data is transmitted electronically, such as in WebSphere MQ. The overall goal is to be able to prove that a particular message is associated with a particular individual. If the sender of a message denies sending it at some later date, the non-repudiation service can provide the recipient with evidence that the message was sent by that particular individual (proof of origin). If the receiver of a message denies receiving it at some later date, the non-repudiation service can provide the sender with evidence that the message was received by that particular individual (proof of delivery).

NSI. See *name service interface (NSI)*.

NTFS. See *New Technology File System (NTFS)*.

NTLM. See *Windows NT LAN Manager (NTLM)*.

null character. The character that is represented by X'00'.

O

OAM. See *object authority manager (OAM)*

object.

1. In WebSphere MQ, an object is a queue manager, queue, process definition, channel, namelist, authentication information object, or storage class (z/OS only).
2. In *object oriented programming*, an object is an instance of a class.

object authority manager (OAM). In WebSphere MQ on UNIX systems, WebSphere MQ for iSeries, and WebSphere MQ for Windows, the default authorization service for command and object management. The OAM can be replaced by, or run in combination with, a customer-supplied security service.

object descriptor. A data structure that identifies a particular WebSphere MQ object. Included in the descriptor are the name of the object and the object type.

object handle. The identifier or token by which a program accesses the WebSphere MQ object with which it is working.

object-oriented programming.

1. A method for structuring programs as hierarchically-organized classes describing the data and operations of objects that might interact with other objects.
2. A programming approach based on the concepts of data abstraction and inheritance. Unlike procedural

programming techniques, object-oriented programming concentrates not on how something is accomplished, but on what data objects comprise the problem and how they are manipulated.

Object-oriented programming languages include Java, C++, and Smalltalk.

off-loading. In WebSphere MQ for z/OS, an automatic process whereby a queue manager's active log is transferred to its archive log.

Open Systems Interconnect (OSI). A model of network architecture and a suite of protocols (a protocol stack) to implement it, developed by ISO in 1978 as a framework for international standards in heterogeneous computer network architecture.

Open Transaction Manager Access (OTMA). A transaction-based, connectionless client/server protocol. It functions as an interface for host-based communications servers accessing IMS TM applications through the z/OS cross systems coupling facility (XCF). OTMA lets you connect clients to servers so that the client can support a large network (or a large number of sessions) while maintaining high performance.

OPM. See *original program model (OPM)*.

original program model (OPM). On earlier versions of MQSeries for AS/400, an environment to help in writing application programs. On WebSphere MQ, we recommend using the *Integrated Language Environment (ILE)*.

OSI. See *Open Systems Interconnect (OSI)*.

OSI directory standard. X.500, the OSI directory standard, defines a comprehensive directory service, including an information model, a namespace, a functional model, and an authentication framework. X.500 also defines the Directory Access Protocol (DAP) used by clients to access the directory. DAP is a full OSI protocol that contains extensive functionality and that is significantly more complicated than the more prevalent TCP/IP stack implementations. The *Lightweight Directory Access Protocol (LDAP)* was designed to remove some of the burden of X.500 access from directory clients, making the directory available to a wider variety of machines and applications.

OTMA. See *Open Transaction Manager Access (OTMA)*.

outbound channel. A channel that takes messages from a transmission queue and sends them to another queue manager. See also *shared outbound channel*.

output log-buffer. In WebSphere MQ for z/OS, a buffer that holds recovery log records before they are written to the archive log.

output parameter. A parameter of an MQI call in which the queue manager returns information when the call completes or fails.

overloading. The existence of more than one flavor of method with the same name or operator, but with different signatures, within a class. The name or operator remains the same, but the method parameters differ, with each signature requiring a separate implementation. Such methods usually exhibit the same behavior, despite differences in signature.

P

page set. A VSAM data set used when WebSphere MQ for z/OS moves data (for example, queues and messages) from buffers in main storage to permanent backing storage (DASD).

parent class. In *object oriented programming*, a class from which another is derived.

partial repository. A partial set of information about queue managers in a cluster. A partial repository is maintained by all cluster queue managers that do not host a *full repository*.

PCF. See *programmable command format (PCF)*

PCF command. See *programmable command format (PCF)*.

pending event. An unscheduled event that occurs as a result of a connect request from a CICS adapter.

percolation. In error recovery, the passing along a preestablished path of control from a recovery routine to a higher-level recovery routine.

performance event. A category of event indicating that a limit condition has occurred.

performance trace. A WebSphere MQ trace option where the trace data is to be used for performance analysis and tuning.

permanent dynamic queue. A dynamic queue that is deleted when it is closed only if deletion is explicitly requested. Permanent dynamic queues are recovered if the queue manager fails, so they can contain persistent messages. Contrast with *temporary dynamic queue*.

persistent message. A message that survives a restart of the queue manager. Contrast with *non-persistent message*.

ping. In distributed queuing, a diagnostic aid that uses the exchange of a test message to confirm that a message channel or TCP/IP connection is functioning.

PKI. See *public key infrastructure (PKI)*.

plaintext. In *cryptology*, any message that is not encrypted. Contrast with *ciphertext*.

platform. In WebSphere MQ, the operating system under which a queue manager is running.

Glossary

point of recovery. In WebSphere MQ for z/OS, the term used to describe a set of backup copies of WebSphere MQ for z/OS page sets and the corresponding log data sets required to recover these page sets. These backup copies provide a potential restart point in the event of page set loss (for example, page set I/O error).

| **polymorphism.** In *object oriented programming*, the characteristic whereby a method can be applied to a variety of classes, with consequent various effects: for example, an *open* method could be applied equally to *book* and *door* class objects.

preemptive shutdown. In WebSphere MQ, a shutdown of a queue manager that does not wait for connected applications to disconnect, or for current MQI calls to complete. Contrast with *immediate shutdown* and *quiesced shutdown*.

| **preferred computer.** An application under *Microsoft Cluster Server (MSCS)* control can be set to have one particular computer as its preferred computer. After a failover to another computer, MSCS monitors the preferred computer until it is repaired, and as soon as it is running correctly again, moves the application back to it.

principal. In WebSphere MQ on UNIX systems, WebSphere MQ for Windows, and MQSeries for OS/2 Warp, a term used for a user identifier. Used by the object authority manager for checking authorizations to system resources.

| **private key.** See *public-private key*.

| **private methods and instance data.** In *object oriented programming*, methods and instance data that are only accessible to the implementation of the same class.

privately-defined object. In z/OS, an object whose definition is stored on page set zero. The definition can be accessed only by the queue manager that defined it. Also known as a *locally-defined object*.

process definition object. A WebSphere MQ object that contains the definition of a WebSphere MQ application. For example, a queue manager uses the definition when it works with trigger messages.

programmable command format (PCF). A type of WebSphere MQ message used by:

- User administration applications, to put PCF commands onto the system command input queue of a specified queue manager
- User administration applications, to get the results of a PCF command from a specified queue manager
- A queue manager, as a notification that an event has occurred

Contrast with *WebSphere MQ script commands (MQSC)*.

program temporary fix (PTF). A solution or bypass of a problem diagnosed by IBM field engineering as the result of a defect in a current, unaltered release of a program.

| **protected methods and instance data.** In *object oriented programming*, methods and instance data that are only accessible to the implementations of the same or derived classes, or from friend classes.

PTF. See *program temporary fix (PTF)*.

| **public key.** See *public-private key*.

| **public key cryptography.** See *asymmetric key cryptography*.

| **public key infrastructure (PKI).** A system of digital certificates, certification authorities, and other registration authorities that verify and authenticate the validity of each party involved in an Internet transaction.

public methods and instance data. Methods and instance data that are accessible to all classes.

| **public-private key.** A system of *cryptography* that uses a unique pair of keys: a public key that is known to everyone, and a private key that is known only to the owner and is kept in the secure environment of the owner. The public and private keys are related in such a way that, if one of the keys is used to encrypt data, only the other key can be used to decrypt the data. A cryptographic system that uses public and private keys is called *asymmetric key cryptography*.

| **Publish/Subscribe.** A WebSphere MQ base product extension supplied in SupportPac MA0C. In a Publish/Subscribe system, there are two types of application: publisher and subscriber. A publisher supplies information in the form of messages that it sends to a broker. A subscriber specifies which topics it is interested in by sending a subscription request to a broker in the form of a message. A broker receives published messages from publishers, subscription requests from subscribers, and routes the published messages to the subscribers. A subscriber is sent messages only about those topics to which it has subscribed.

Q

queue. A WebSphere MQ object. Message queuing applications can put messages on, and get messages from, a queue. A queue is owned and maintained by a queue manager. Local queues can contain a list of messages waiting to be processed. Queues of other types cannot contain messages; they point to other queues, or can be used as models for dynamic queues.

queue manager.

1. A system program that provides queuing services to applications. It provides an application programming interface so that programs can access messages on the queues that the queue manager owns. See also *local queue manager* and *remote queue manager*.
2. A WebSphere MQ object that defines the attributes of a particular queue manager.

queue manager event. An event that indicates:

- An error condition has occurred in relation to the resources used by a queue manager. For example, a queue is unavailable.
- A significant change has occurred in the queue manager. For example, a queue manager has stopped or started.

| **queue manager cluster.** See *cluster*.

| **queue manager level security.** In WebSphere MQ for z/OS, the *authorization checks* that are performed using RACF profiles specific to a queue manager.

queue-sharing group. In WebSphere MQ for z/OS, a group of *queue managers* in the same sysplex that can access a single set of object definitions stored in the shared repository, and a single set of shared queues stored in the coupling facility. See also *shared queue*.

| **queue-sharing group level security.** In WebSphere MQ for z/OS, the *authorization checks* that are performed using RACF profiles that are shared by all *queue managers* in a *queue-sharing group*.

queuing. See *message queueing*.

quiesced shutdown.

1. In WebSphere MQ, a shutdown of a queue manager that allows all connected applications to disconnect. Contrast with *immediate shutdown* and *preemptive shutdown*.
2. A type of shutdown of the CICS adapter where the adapter disconnects from WebSphere MQ, but only after all the currently active tasks have been completed. Contrast with *forced shutdown*.

quiescing. In WebSphere MQ, the state of a queue manager before it stops. In this state, programs are allowed to finish processing, but no new programs are allowed to start.

| **quorum disk.** The disk accessed exclusively by *Microsoft Cluster Server (MSCS)* to store the cluster recovery log, and to determine whether a server is up or down. Only one server can own the quorum disk at a time. Servers in the cluster can negotiate for the ownership.

R

| **RACF.** See *resource access control facility (RACF)*.

| **RAID.** See *redundant array of independent disks (RAID)*.

RBA. See *relative byte address (RBA)*.

reason code. A return code that describes the reason for the failure or partial success of an MQI call.

| **receive exit.** A type of *channel exit program*. For an explanation, see *send exit*.

receiver channel. In message queuing, a channel that responds to a sender channel, takes messages from a communication link, and puts them on a local queue.

recovery log. In WebSphere MQ for z/OS, data sets containing information needed to recover messages, queues, and the WebSphere MQ subsystem. WebSphere MQ for z/OS writes each record to a data set called the *active log*. When the active log is full, its contents are off-loaded to a DASD or tape data set called the *archive log*. Synonymous with *log*.

recovery termination manager (RTM). A program that handles all normal and abnormal termination of tasks by passing control to a recovery routine associated with the terminating function.

| **redundant array of independent disks (RAID).** A device that looks like a single disk drive to the operating system, but internally contains two or more hard disks with redundant space, and monitoring firmware. If part of a disk fails, the internal firmware can remap another working part to that address.

reference message. A message that refers to a piece of data to be transmitted. The reference message is handled by message exit programs, which attach and detach the data from the message so allowing the data to be transmitted without having to be stored on any queues.

Registry. In Windows, a secure database that provides a single source for system and application configuration data.

Registry Editor. In Windows, the program item that allows the user to edit the Registry.

Registry Hive. In Windows, the structure of the data stored in the Registry.

relative byte address (RBA). The displacement in bytes of a stored record or control interval from the beginning of the storage space allocated to the data set to which it belongs.

remote queue. A queue belonging to a remote queue manager. Programs can put messages on remote

Glossary

queues, but they cannot get messages from remote queues. Contrast with *local queue*.

remote queue manager. To a program, a queue manager that is not the one to which the program is connected.

remote queue object. See *local definition of a remote queue*.

remote queuing. In message queuing, the provision of services to enable applications to put messages on queues belonging to other queue managers.

reply message. A type of message used for replies to request messages. Contrast with *request message* and *report message*.

reply-to queue. The name of a queue to which the program that issued an MQPUT call wants a reply message or report message sent.

report message. A type of message that gives information about another message. A report message can indicate that a message has been delivered, has arrived at its destination, has expired, or could not be processed for some reason. Contrast with *reply message* and *request message*.

repository. A collection of information about the queue managers that are members of a cluster. This information includes queue manager names, their locations, their channels, what queues they host, and so on.

repository queue manager. A queue manager that hosts the *full repository* of information about a cluster.

| **requester channel.** In message queuing, a channel that
| can be started locally to initiate operation of a *server*
| *channel*.

request message. A type of message used to request a reply from another program. Contrast with *reply message* and *report message*.

RESLEVEL. In WebSphere MQ for z/OS, an option that controls the number of CICS user IDs checked for API-resource security in WebSphere MQ for z/OS.

resolution path. The set of queues that are opened when an application specifies an alias or a remote queue on input to an MQOPEN call.

resource. Any facility of the computing system or operating system required by a job or task. In WebSphere MQ for z/OS, examples of resources are buffer pools, page sets, log data sets, queues, and messages.

| **resource access control facility (RACF).** In z/OS, a
| licensed program that provides access control by
| identifying users to the system, verifying users of the
| system, authorizing access to protected resources,

| logging detected, unauthorized attempts to enter the
| system, and logging detected accesses to protected
| resources.

resource manager. An application, program, or transaction that manages and controls access to shared resources such as memory buffers and data sets. WebSphere MQ, CICS, and IMS are resource managers.

Resource Recovery Services (RRS). A z/OS facility that provides two-phase syncpoint support across participating resource managers.

responder. In distributed queuing, a program that replies to network connection requests from another system.

resynch. In WebSphere MQ, an option to direct a channel to start up and resolve any in-doubt status messages, but without restarting message transfer.

return codes. The collective name for completion codes and reason codes.

return-to-sender. An option available to an MCA that is unable to deliver a message. The MCA can send the message back to the originator.

rollback. Synonym for *backout*.

RRS. See *Resource Recovery Services (RRS)*.

| **RSA.** Public-key encryption technology developed by
| RSA Data Security, Inc, and used in our implementation
| of *Secure Sockets Layer (SSL)*.

RTM. See *recovery termination manager (RTM)*.

rules table. A control file containing one or more rules that the dead-letter queue handler applies to messages on the DLQ.

S

SAF. See *System Authorization Facility (SAF)*.

Scalable Parallel 2 (SP2®). IBM's parallel UNIX system: effectively parallel AIX systems on a high-speed network.

| **SDK.** See *software development kit (SDK)*.

SDWA. See *system diagnostic work area (SDWA)*

| **secret key cryptography.** See *symmetric key*
| *cryptography*.

| **Secure Sockets Layer (SSL).** A protocol developed by
| Netscape for transmitting data securely over the
| Internet.

security enabling interface (SEI). The WebSphere MQ interface to which customer- or vendor-written programs that check authorization, supply a user

identifier, or perform authentication must conform. A part of the WebSphere MQ Framework.

security exit. A *channel exit program* that is called immediately after the initial data negotiation has completed on channel startup. Security exits normally work in pairs and can be called on both *message channels* and *MQI channels*. The primary purpose of the security exit is to enable the *message channel agent (MCA)* at each end of a channel to authenticate its partner.

security identifier (SID). On Windows systems, a supplement to the user ID that identifies the full user account details on the Windows security account manager database where the user is defined.

security mechanism. A technical tool or technique that is used to implement a security service. A mechanism might operate by itself, or in conjunction with others, to provide a particular service. Examples of security mechanisms include *access control list (ACL)s*, *cryptography*, and *digital signatures*.

security message. When *security exits* are called at both ends of a channel, they communicate with each other by sending security messages. The format of a security message is not defined and is determined by the user.

security service. A service within a computer system that protect its resources. Access control is an example of a security service.

Security Support Provider Interface (SSPI). The means for networked applications to call one of several security support providers (SSPs) to establish authenticated connections and to exchange data securely over those connections. It is available for use on Windows systems.

SEI. See *security enabling interface (SEI)*.

selector. Used to identify a data item. In the MQAI there are two types of selector: a user selector and a system selector.

semaphore. In UNIX systems:

1. An entity used to control access to system resources. Processes can be locked to a resource with semaphores if the processes follow certain programming conventions.
2. A general method of communication between two processes that extends the features of signals.

sender channel. In message queuing, a channel that initiates transfers, removes messages from a transmission queue, and moves them over a communication link to a receiver or requester channel.

send exit. A type of *channel exit program*. A send exit at one end of a channel and a receive exit at the other end usually work in pairs. A send exit is called just before a

message channel agent (MCA) issues a communications send to send a unit of data over a communications connection. A receive exit is called just after the *message channel agent (MCA)* has regained control following a communications receive and has received a unit of data from a communications connection. Send and receive exits can modify the contents of each unit of data and change its length. Send and receive exits can be called on both *message channels* and *MQI channels*.

Sequenced Package Exchange transmission protocol (SPX). A transport layer protocol used in Novell Netware networks. The SPX layer provides connection-oriented services between two nodes on the network, and is used primarily by client/server applications.

sequence number wrap value. In WebSphere MQ, a method of ensuring that both ends of a communication link reset their current message sequence numbers at the same time. Transmitting messages with a sequence number ensures that the receiving channel can reestablish the message sequence when storing the messages.

serialization. In *object oriented programming*, writing data in a sequential fashion to a communications medium from program memory.

server. (1) In WebSphere MQ, a queue manager that provides queue services to client applications running on a remote workstation. (2) The program that responds to requests for information in the particular two-program, information-flow model of client/server. See also *client*.

server channel. In message queuing, a channel that responds to a requester channel, removes messages from a transmission queue, and moves them over a communication link to the requester channel.

server connection channel type. The type of MQI channel definition associated with a server. See also *client connection channel type*.

service interval. A time interval, against which the elapsed time between a put or a get and a subsequent get is compared by the queue manager in deciding whether the conditions for a service interval event have been met. The service interval for a queue is specified by a queue attribute.

service interval event. An event related to the service interval.

session. In *Systems Network Architecture (SNA)*, a logical connection between two logical units (LUs) which can be activated and deactivated as required. A session acts as a conduit through which data moves between the two LUs.

session ID. In WebSphere MQ for z/OS, the CICS-unique identifier that defines the communication

Glossary

link to be used by a message channel agent when moving messages from a transmission queue to a link.

| **session level authentication.** In *Systems Network Architecture (SNA)*, a *session* level security protocol that enables two logical units (LUs) to authenticate each other while they are activating a session. Session level authentication is also known as LU-LU verification.

| **session level cryptography.** In *Systems Network Architecture (SNA)*, session level cryptography encrypts and decrypts data that flows on a *session* between two logical units (LUs).

shared inbound channel. In WebSphere MQ for z/OS, a channel that was started by a listener using the group port. The channel definition of a shared channel can be stored either on page set zero (private) or in the shared repository (global).

shared outbound channel. In WebSphere MQ for z/OS, a channel that moves messages from a shared transmission queue. The channel definition of a shared channel can be stored either on page set zero (private) or in the shared repository (global).

shared queue. In WebSphere MQ for z/OS, a type of local queue. The messages on the queue are stored in the *coupling facility* and can be accessed by one or more queue managers in a *queue-sharing group*. The definition of the queue is stored in the *shared repository*.

shared repository. In WebSphere MQ for z/OS, a shared DB2 database that is used to hold object definitions that have been defined globally.

| **shell.** In the UNIX environment, a software interface between a user and the operating system of a computer. Shell programs interpret commands and communicate them to the operating system.

shutdown. See *immediate shutdown*, *preemptive shutdown*, and *quiesced shutdown*.

| **SID.** See *security identifier (SID)*.

| **signal.** In UNIX systems a simple method of communicating between two processes. One process can inform the other when an event occurs.

signaling. In WebSphere MQ for z/OS and WebSphere MQ for Windows, a feature that allows the operating system to notify a program when an expected message arrives on a queue.

| **signature.** In *object oriented programming*, a distinct combination of method name or operator, and parameters.

single logging. A method of recording WebSphere MQ for z/OS activity where each change is recorded on one data set only. Contrast with *dual logging*.

single-phase backout. A method in which an action in progress must not be allowed to finish, and all changes that are part of that action must be undone.

single-phase commit. A method in which a program can commit updates to a queue without coordinating those updates with updates the program has made to resources controlled by another resource manager. Contrast with *two-phase commit*.

SIT. See *system initialization table (SIT)*.

SMF. See *system management facility (SMF)*.

SNA. See *Systems Network Architecture (SNA)*.

| **software development kit (SDK).** A set of tools, APIs, and documentation to help you develop software in a specific computer language or for a particular operating environment.

source queue manager. See *local queue manager*.

SPX. See *Sequenced Package Exchange transmission protocol (SPX)*.

SP2. See *Scalable Parallel 2 (SP2)*.

| **SSL.** See *Secure Sockets Layer (SSL)*.

| **SSPI.** See *Security Support Provider Interface (SSPI)*.

stanza. A group of lines in a configuration file that assigns a value to a parameter modifying the behavior of a queue manager, client, or channel. In WebSphere MQ on UNIX systems a configuration (.ini) file can contain a number of stanzas.

star-connected communications network. A network in which all nodes are connected to a central node.

storage class. In WebSphere MQ for z/OS, a storage class defines the page set to hold the messages for a particular queue. The storage class is specified when the queue is defined.

store and forward. The temporary storing of packets, messages, or frames in a data network before they are retransmitted toward their destination.

| **streaming.** In *object oriented programming*, marshalling class information and object instance data.

| **subsystem.** In z/OS, a service provider that performs one or many functions, but does nothing until asked. For example, each WebSphere MQ for z/OS queue manager is a z/OS subsystem.

supervisor call (SVC). A z/OS instruction that interrupts a running program and passes control to the supervisor so that it can perform the specific service indicated by the instruction.

SVC. See *supervisor call (SVC)*.

switch profile. In WebSphere MQ for z/OS, a RACF profile used when WebSphere MQ starts up or when a refresh security command is issued. Each switch profile that WebSphere MQ detects turns off checking for the specified resource.

symmetric key cryptography. A system of *cryptography* in which the sender and receiver of a message share a single, common, secret key that is used to encrypt and decrypt the message. Contrast with *asymmetric key cryptography*. See also *public-private key*.

symptom string. Diagnostic information displayed in a structured format designed for searching the IBM software support database.

synchronous messaging. A method of communication between programs in which a program places a message on a message queue and then waits for a reply before resuming its own processing. Contrast with *asynchronous messaging*.

sync point. An intermediate or end point during processing of a transaction at which the transaction's protected resources are consistent. At a syncpoint, changes to the resources can safely be committed, or they can be backed out to the previous syncpoint.

sysplex. A multiple z/OS-system environment that allows multiple-console support (MCS) consoles to receive console messages and send operator commands across systems.

System Authorization Facility (SAF). A z/OS facility through which WebSphere MQ for z/OS communicates with an external security manager such as RACF.

system bag. A type of data bag that is created by the MQAI.

system control commands. Commands used to manipulate platform-specific entities such as buffer pools, storage classes, and page sets.

system diagnostic work area (SDWA). Data recorded in a SYS1.LOGREC entry, which describes a program or hardware error.

system initialization table (SIT). A table containing parameters used by CICS on startup.

system item. A type of data item that is created by the MQAI.

system management facility (SMF). A z/OS facility that collects and records system and job-related information that you can then use to create reports for use in managing your installation.

system selector. In the MQAI, used to identify a system item. A system selector is included in the data bag when it is created.

Systems Network Architecture (SNA). The description of the logical structure, formats, protocols, and operational sequences for transmitting information units through, and controlling the configuration and operation of, networks.

SYS1.LOGREC. A service aid containing information about program and hardware errors.

T

tampering. A breach of communication security in which information in transit is changed or replaced and then sent on to the recipient. See also *eavesdropping* and *impersonation*.

target library high-level qualifier (thlqual). High-level qualifier for z/OS target data set names.

target queue manager. See *remote queue manager*.

task control block (TCB). A z/OS control block used to communicate information about tasks within an address space that are connected to a z/OS subsystem such as WebSphere MQ for z/OS or CICS.

task switching. The overlapping of I/O operations and processing between several tasks. In WebSphere MQ for z/OS, the task switcher optimizes performance by allowing some MQI calls to be executed under subtasks rather than under the main CICS TCB.

TCB. See *task control block (TCB)*.

TCP. See *Transmission Control Protocol (TCP)*.

TCP/IP. See *Transmission Control Protocol/Internet Protocol (TCP/IP)*.

temporary dynamic queue. A dynamic queue that is deleted when it is closed. Temporary dynamic queues are not recovered if the queue manager fails, so they can contain nonpersistent messages only. Contrast with *permanent dynamic queue*.

teraspace. A form of shared memory introduced in OS/400 V4R4.

termination notification. A pending event that is activated when a CICS subsystem successfully connects to WebSphere MQ for z/OS.

this. In *object oriented programming*, the reserved word that represents a pointer to the current object.

thlqual. See *target library high-level qualifier (thlqual)*.

thread. In WebSphere MQ, the lowest level of parallel execution available on an operating system platform.

time-independent messaging. See *asynchronous messaging*.

Glossary

TMI. See *trigger monitor interface (TMI)*.

trace. In WebSphere MQ, a facility for recording WebSphere MQ activity. The destinations for trace entries can include *generalized trace facility (GTF)* and the *system management facility (SMF)*.

tranid. See *transaction identifier*.

transaction. See *unit of work* and *CICS transaction*.

transaction identifier. In CICS, a name that is specified when the transaction is defined, and that is used to invoke the transaction.

transaction manager. A software unit that coordinates the activities of resource managers by managing global transactions and coordinating the decision to commit them or roll them back. WebSphere MQ is a transaction manager.

| **transaction program.** An application that is used for
| cooperative processing within a *Systems Network*
| *Architecture (SNA)* network.

Transmission Control Protocol (TCP). A communications protocol used in the Internet and in any network that follows the Internet Engineering Task Force (IETF) standards for internetwork protocol. TCP provides a reliable host-to-host protocol between hosts in packet-switched communications networks and in interconnected systems of such networks. It uses the Internet Protocol (IP) as the underlying protocol.

Transmission Control Protocol/Internet Protocol (TCP/IP).

1. The Transmission Control Protocol and the Internet Protocol, which together provide reliable end-to-end connections between applications over interconnected networks of different types.
2. The suite of transport and application protocols that run over the Internet Protocol.

transmission program. See *message channel agent (MCA)*.

transmission queue. A local queue on which prepared messages destined for a remote queue manager are temporarily stored.

trigger event. An event (such as a message arriving on a queue) that causes a queue manager to create a trigger message on an initiation queue.

triggering. In WebSphere MQ, a facility allowing a queue manager to start an application automatically when predetermined conditions on a queue are satisfied.

trigger message. A message containing information about the program that a trigger monitor is to start.

trigger monitor. A continuously-running application serving one or more initiation queues. When a trigger message arrives on an initiation queue, the trigger monitor retrieves the message. It uses the information in the trigger message to start a process that serves the queue on which a trigger event occurred.

trigger monitor interface (TMI). The WebSphere MQ interface to which customer- or vendor-written trigger monitor programs must conform. A part of the WebSphere MQ Framework.

two-phase commit. A protocol for the coordination of changes to recoverable resources when more than one resource manager is used by a single transaction. Contrast with *single-phase commit*.

| **type.** In *object oriented programming*, a fundamental data type of computer architecture, including for example character string and integer.

U

UDP. See *User Datagram Protocol (UDP)*.

| **unauthorized access.** Gaining access to resources
| within a computer system without permission.

undelivered-message queue. See *dead-letter queue (DLQ)*.

undo/redo record. A log record used in recovery. The redo part of the record describes a change to be made to a WebSphere MQ object. The undo part describes how to back out the change if the work is not committed.

unit of recovery. A recoverable sequence of operations within a single resource manager. Contrast with *unit of work*.

unit of work. A recoverable sequence of operations performed by an application between two points of consistency. A unit of work begins when a transaction starts or after a user-requested syncpoint. It ends either at a user-requested syncpoint or at the end of a transaction. Contrast with *unit of recovery*.

user bag. In the MQAI, a type of data bag that is created by the user.

User Datagram Protocol (UDP). In the Internet suite of protocols, a protocol that provides unreliable, connectionless datagram service. It enables an application program on one machine or process to send a datagram to an application program on another machine or process. UDP uses the Internet Protocol (IP) to deliver datagrams.

user item. In the MQAI, a type of data item that is created by the user.

user selector. In the MQAI, used to identify a user item. For the administration of WebSphere MQ objects, valid user selectors are already defined.

utility. In WebSphere MQ, a supplied set of programs that provide the system operator or system administrator with facilities in addition to those provided by the WebSphere MQ commands.

V

value. Value of a data item. This can be an integer, a string, or a handle of another bag.

- | **virtual method.** In *object oriented programming*, a method that exhibits polymorphism.

W

WebSphere MQ. A family of IBM licensed programs that provides message queuing services.

WebSphere MQ Administration Interface (MQAI). A programming interface to WebSphere MQ.

- | **WebSphere MQ channel protocol flows.** See *Formats and Protocols (FAP)*.

WebSphere MQ client. Part of a WebSphere MQ product that can be installed on a system without installing the full queue manager. The WebSphere MQ client accepts MQI calls from applications and communicates with a queue manager on a server system.

- | **WebSphere MQ Internet pass-thru.** A WebSphere MQ base product extension supplied in SupportPac MS81. It enables two queue managers to exchange messages, or a WebSphere MQ client application to connect to a queue manager, over the Internet without requiring a direct TCP/IP connection.

WebSphere MQ script (MQSC) commands . Human readable commands, uniform across all platforms, that are used to manipulate WebSphere MQ objects.

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

- | **Windows NT Challenge/Response (NTLM).** See *Windows NT LAN Manager (NTLM)*.

- | **Windows NT LAN Manager (NTLM).** Windows NT Challenge/Response (NTLM) is the authentication protocol used on networks that include Windows NT systems and on standalone systems. NTLM stands for

- | Windows NT LAN Manager to distinguish the challenge/response-based protocol from its predecessor, LAN Manager.

- | **wiretapping.** The act of gaining access to information that is flowing along a wire or any other type of conductor used in communications. The objective of wiretapping is to gain unauthorized access to information without being detected.

X

- | **X.500.** See *OSI directory standard*.

X/Open XA. The X/Open Distributed Transaction Processing XA interface. A proposed standard for distributed transaction communication. The standard specifies a bidirectional interface between resource managers that provide access to shared resources within transactions, and between a transaction service that monitors and resolves transactions.

XCF. See *cross systems coupling facility (XCF)*.

Glossary

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.

SC34-6113-02

