

WebSphere MQ



Migration Guide

Version 7.0

WebSphere MQ



Migration Guide

Version 7.0

Note

Before using this information and the product it supports, be sure to read the general information under notices at the back of this book.

Second edition (January 2009)

This edition of the book applies to the following products:

- IBM WebSphere MQ, Version 7.0
- IBM WebSphere MQ for z/OS, Version 7.0

and to any subsequent releases and modifications until otherwise indicated in new editions.

© **Copyright International Business Machines Corporation 2005, 2009.**

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

Contents

Figures	v	Chapter 5. Platform-specific information	29
Tables	vii	Migrating to WebSphere MQ for AIX Version 7.0	29
Chapter 1. Introduction to WebSphere MQ Version 7.0 migration	1	Migrating from an earlier version of WebSphere MQ for AIX client	30
Chapter 2. Migrating WebSphere MQ applications	3	Changes to the installation path	31
Definition of upward and downward compatibility	10	Removal of obsolete code pages from WebSphere MQ for AIX	32
Chapter 3. Coexistence.	13	Defining a transaction program.	32
Coexistence on WebSphere MQ for AIX	13	After you have migrated to WebSphere MQ Version 7.0.	33
Coexistence on WebSphere MQ for HP-UX	13	Migrating to WebSphere MQ for HP-UX Version 7.0	34
Uninstalling WebSphere MQ.	13	After you have migrated to WebSphere MQ Version 7.0.	35
Coexistence on WebSphere MQ for i5/OS	14	Migrating WebSphere MQ for i5/OS	35
Coexistence on WebSphere MQ for Linux	14	Migrating from WebSphere MQ V5.3 or later	35
Uninstalling Websphere MQ.	14	Preparing for migration	36
Coexistence on WebSphere MQ for Solaris	15	Quiescing WebSphere MQ	37
Uninstalling WebSphere MQ.	15	Performing a slip install	39
Coexistence on WebSphere MQ for Windows	16	Performing a side-by-side install	39
Coexistence on WebSphere MQ for z/OS	16	Migrating translated versions	40
Multiple queue manager versions in z/OS	16	Migrating WebSphere MQ classes for JMS and WebSphere MQ classes for Java on i5/OS	41
Multiple queue manager versions in a queue-sharing group	16	Verifying the migration	41
Operations and control panels	19	After migration	42
Application stubs	19	Upgrading to WebSphere MQ for Linux Version 7.0	42
Coexistence of WebSphere MQ classes for JMS	19	Upgrading to WebSphere MQ Version 7.0	43
Chapter 4. Migrating WebSphere MQ components to Version 7.0	21	Upgrading to WebSphere MQ Version 7.0 using RPM upgrade tools.	44
Migrating a queue manager cluster to WebSphere MQ Version 7.0.	21	After you have upgraded to WebSphere MQ Version 7	44
Migrating a queue manager cluster: Creating a plan	21	Migrating to WebSphere MQ for Solaris Version 7.0	45
Migrating a queue manager cluster: Creating a backout plan	24	After you have migrated to WebSphere MQ Version 7.0.	46
Migrating a queue manager cluster: Setting up a test environment.	24	Migrating to WebSphere MQ for Windows Version 7.0	46
Migrating the test system.	24	Choosing not to install WebSphere MQ Explorer and WebSphere Eclipse Platform	47
Migrating a queue manager cluster: Migrating the production system.	25	Migrating to WebSphere MQ Version 7.0 on Windows Vista or Windows Server 2008.	48
Migrating client-connection and server-connection channels	25	User Account Control (UAC) on Windows Vista and Windows Server 2008	48
Performance implications of sharing conversations on client-connection channels	26	Launchpad migration instructions	48
Migrating JMS Exception listeners	27	SPX support on Windows Vista and Windows Server 2008	49
Change in behavior of MQMessage.resizeBuffer in WebSphere MQ classes for Java.	27	WebSphere MQ service objects	49
WebSphere MQ Version 7.0 transport for SOAP	28	WebSphere MQ Explorer for Version 7.0 - error connecting to a migrated queue manager	50
MQCONN reason codes	28	Migrating an MSCS configuration	50
		Upgrading to WebSphere MQ for z/OS Version 7.0	51
		Migrating to WebSphere MQ for z/OS Version 7.0	52
		Migrating from Version 6.0	54
		Changing to full function WebSphere MQ	62
		Reverting to previous versions	63

Migrating from an earlier unsupported release of WebSphere MQ or MQSeries	63
Chapter 6. WebSphere MQ classes for JMS and classes for Java considerations	85
Chapter 7. Migration to publish/subscribe on WebSphere MQ V7.0	89
strmqbrk (Migrate WebSphere MQ Version 6.0 broker to Version 7.0)	89
Application migration	91
Delete Publication - Version 7 replacement	92
Deregister publisher - Version 7 replacement	93
Deregister subscriber - Version 7 replacement	94
Publish - Version 7 replacement	96
Register publisher - Version 7 replacement	97
Register subscriber - Version 7 replacement	98
Request Update - Version 7 replacement	101
New queue manager attributes for publish/subscribe	101
WebSphere MQ publish/subscribe topology migration.	102
Migrating a WebSphere MQ Version 6.0 publish/subscribe hierarchy to a Version 7.0 publish/subscribe cluster - all queue managers simultaneously	102
Migrating a WebSphere MQ Version 6.0 publish/subscribe hierarchy to a Version 7.0 publish/subscribe cluster - queue manager by queue manager.	103
Migrating a WebSphere MQ Version 6.0 two queue manager publish/subscribe hierarchy to a Version 7.0 hierarchy - parent first	104
Migrating a WebSphere MQ Version 6.0 two publish/subscribe queue manager hierarchy to a Version 7.0 hierarchy - child first	105
Using publish/subscribe with WebSphere MQ classes for JMS	105
Subscription name migration on the JMS client	108
Migration implications of mapping an alias queue to a topic object	109
Migrated topologies	110
Heterogeneous networks	110
Leaf node	110
Parent node	110
Chapter 8. Upgrading and migrating WebSphere MQ clients from WebSphere MQ Version 6.0 to WebSphere MQ Version 7.0	111
Upgrading a WebSphere MQ client from Version 6.0 to Version 7.0	111
Migration and client channel definition tables	112

Chapter 9. Migrating from WebSphere MQ Version 5.3 to WebSphere MQ Version 7.0	113
64-bit queue manager migration information	114
32-bit and 64-bit files	117
COBOL copybooks	119
Internet Protocol Version 6 (IPv6) migration	120
Key points in migrating to IPv6 and using WebSphere MQ Version 7.0.	120
Considerations when implementing IPv6 in a network	121
Migrating a queue manager to IPv6	121
Migrating a cluster to IPv6	123
Abbreviated migration scenarios	125
Migrating Windows Secure Sockets Layer (SSL) connections	129
Determining whether SSL connections have been set up	132
SSL migration steps	133
Additional migration information	144
Channels implemented as queue manager objects on distributed platforms	145
Setting the MQCD Version field	145
DCE support in WebSphere MQ	146
External Transaction Manager XA support.	146
Queue manager attributes SCHINIT and SCMDSERV	146
Starting and stopping services and listeners	146
UNIX directory permissions	147
High availability clustering.	147
Considerations for migrating from WebSphere MQ for Linux Version 5.3	147
Migrating from earlier versions of WebSphere MQ for Linux	147
WebSphere MQ Explorer for Version 7.0 - error connecting to a migrated queue manager	148
Considerations for migrating from WebSphere MQ for Windows Version 5.3.	148
Customizing your queue managers	148
Migrating from Windows NT	148
Installable features	149
WebSphere MQ service objects	149
WebSphere MQ Explorer - migrating from Version 5.3 to 7.0	150
Migrating WebSphere MQ Services	150
Publish/Subscribe users.	150
Saved views with WebSphere MQ Explorer	151
MCS migration issues	151
PL/I support in WebSphere MQ for Windows	152
Migrating from UDP	152
Java archive (JAR) file com.ibm.mqbind.jar	152
Notices	153
Index	157
Sending your comments to IBM	161

Figures

1.	WebSphere MQ application migration model	6	3.	Example job for migrating queue objects	85
2.	Example job for migrating channel objects	78			

Tables

1. Upward and downward compatibility	10	7. WebSphere MQ Version 7.0 on Windows Vista or Windows Server 2008: supported migration paths.	48
2. Compatibility of queue manager versions with operations and control panels versions	19	8.	102
3. Queue managers, repositories and cluster queues	22	9. Data type platform sizes for 32-bit and 64-bit applications (Bracketed figures are for 64-bit size).	116
4. SupportPacs superseded by WebSphere MQ Version 7.0.	29	10. Effects of CONNAME and LOCLADDR settings	126
5. SupportPacs superseded by WebSphere MQ Version 7.0.	35	11. Abbreviations used in system configurations	128
6. SupportPacs superseded by WebSphere MQ Version 7.0.	45	12. System configurations.	128

Chapter 1. Introduction to WebSphere MQ Version 7.0 migration

An introduction to the documentation if you are considering migrating to WebSphere® MQ Version 7.0.

The migration process is described and instructions given for all platforms. There is additional information for z/OS®, in the *WebSphere MQ for z/OS Program Directory*.

This collection of topics highlights the main areas that you must consider and gives guidance on the type of action that you need to take to continue 'business as usual'. It is not the intention of these topics to cover the migration of every possible combination of a WebSphere MQ Version 7.0 installation and associated applications.

Note:

Before upgrading any software or migrating WebSphere MQ components, make a full backup of your system to ensure that you can restore the system to its original working environment if a problem occurs.

For distributed platforms, information about how to back up your system begins with "Backing up and restoring WebSphere MQ" in Chapter 5 of *WebSphere MQ System Administration Guide*.

For z/OS, information about backups begins with "Planning for backup and recovery" in Chapter 4 of *WebSphere MQ for z/OS Concepts and Planning Guide*.

Some general considerations

The following lists some of the general issues to consider when making a migration or upgrade plan.

- Develop a backup plan in which you back up the relevant information on the queue manager and server.
- Read the latest README file for the product you are working with. You can find these files online for the relevant products starting from the IBM® WebSphere MQ Support Web page at <http://www.ibm.com/support/docview.wss...>
- If you need to have the queue manager active all the time, and you are unable to shut down the server running this queue manager, you might need to consider a different migration approach. This will mean performing the migration using the following general steps:
 - Copy all resources from the server concerned to another server.
 - Perform a migration on the duplicate server.
 - Switch over to the new server and queue manager at a convenient time.
 - Alternatively, in a queue-sharing environment on z/OS, you can migrate queue managers in a staged or sequential rolling manner to maintain 100% availability of shared queues.

- Be prepared to record a number of details about the existing system topology, including such things as the names of the queue managers and their queues, clients, channels and so on.
- Check whether there are any queues or queue managers that are no longer required.
- When you have upgraded from a previous version of WebSphere MQ to WebSphere MQ Version 7.0 on an individual computer or a z/OS image:
 - On distributed, then any queue manager on that computer is migrated to WebSphere MQ Version 7.0 when you start the queue manager. You might decide to keep instances of WebSphere MQ, and therefore their queue managers, on earlier versions of WebSphere MQ, and administer the queue managers from a computer on which you have upgraded to WebSphere MQ Version 7.0.
 - On z/OS, you will need to migrate the queue managers individually. You might decide to keep instances of WebSphere MQ, and therefore their queue managers, on earlier versions of WebSphere MQ, and administer the queue managers from a computer on which you have upgraded to WebSphere MQ Version 7.0

Migrating from a beta version

If you have previously installed a WebSphere MQ beta driver, you must uninstall this driver and remove all remaining files **before** you install the general availability version of WebSphere MQ Version 7.0.

Supported environments

The operating system on which you want to run a migrated WebSphere MQ Version 7.0 queue manager must be a supported WebSphere MQ Version 7.0 platform as documented on the Supported Environment Web page at: <http://www.ibm.com/software/integration/wmq/requirements/>. Supported environments are also listed in the Quick Beginnings guide for your operating system, or the *WebSphere MQ for z/OS Concepts and Planning Guide*.

Migration requirements can vary according to the environment. Check each of the platform-specific sections for details.

Queue managers from previous versions of WebSphere MQ (for example, WebSphere MQ Version 6.0) are supported, provided that they are on a platform supported by WebSphere MQ Version 7.0.

Note: Where WebSphere MQ Version 5.3 is stated, the information also applies to WebSphere MQ Version 5.3.1 for z/OS.

Chapter 2. Migrating WebSphere MQ applications

You do not need to migrate WebSphere MQ applications when a queue manager is upgraded. Applications built with earlier versions of WebSphere MQ continue to run when MQ is upgraded. When you choose to migrate an application, compile and link it with the new version of WebSphere MQ. Write your applications to be compatible with different versions of WebSphere MQ.

Application migration is defined as recoding, recompiling or relinking an application to be compatible with, or to take advantage of, new versions of underlying software. In general recoding, recompiling and relinking are termed *rebuilding*. If an application does not need to be rebuilt to run on a new version of its underlying software, then no application migration is required, and the application said to be upwardly compatible. More rarely, if an application is built against a later version of its underlying software and runs against an earlier version, the application is said to be downwardly compatible. A discussion of upward and downward compatibility is found in “Definition of upward and downward compatibility” on page 10.

Upward compatibility

Upward compatibility is a contract between the application and the underlying software. An upwardly compatible application is built using only functions that are supported in both present and future releases of the software, and the software guarantees to maintain the interfaces and behavior of those functions. Typically functions that will not be supported in future releases are *deprecated* one or two releases before they are withdrawn. An upwardly compatible application requires no rebuilding when underlying software is upgraded.

You do not need to re-code, recompile, or relink a WebSphere MQ application to run it on a new version of a queue manager. WebSphere MQ applications are *upwardly compatible*. If you upgrade a queue manager, you can continue to deploy applications built against earlier versions of the queue manager to the server running the upgraded queue manager. Likewise, if you upgrade a client installation, you do not need to re-code, recompile, or relink the client application, and you can continue to deploy client applications built against earlier versions of WebSphere MQ.

When you run an existing application with a new level of queue manager, is it upwardly compatible?

1. Can you run an existing WebSphere MQ server application, built with an earlier WebSphere MQ version, on a server with a later queue manager installed?
2. Can you run an existing WebSphere MQ client application, built with an earlier WebSphere MQ version, on a client platform with an earlier client installation when connected to the later queue manager?
3. Can you run an existing WebSphere MQ client application, built with an earlier WebSphere MQ version, on a client platform with a later client installation when connected to an earlier queue manager?
4. Can you exchange messages between an existing WebSphere MQ client or server application, connected to an earlier queue manager, with applications connected to a later queue manager?

The answer to all these questions is, *yes*.

You might make changes in the operating environment as a result of migrating to the new level of queue manager. The operating environment changes, rather than changes in WebSphere MQ itself, might require application change, recompilation, or relinking. Even then, you can sometimes choose to build the application with the tools shipped with the earlier WebSphere MQ version rather than migrate the development environment to the new level of WebSphere MQ. Nonetheless, migrating WebSphere MQ applications as part of migrating to a new level of queue manager is something you will do eventually, and requires planning, control and testing.

Upward version interoperability is a contract between WebSphere MQ clients and queue managers, and between queue managers. Earlier clients and queue managers can connect to, and exchange messages with, later versions. An application connected to an earlier version of the WebSphere MQ client and queue manager is not affected by upgrading the queue manager. WebSphere MQ client applications do not need to change to connect to new versions of a queue manager.

It also means that an application that exchanges messages with applications connected to other queue managers will continue to work if the other queue managers are upgraded to new versions, and the applications do not change. WebSphere MQ clients and queue managers are *upwardly version interoperable*.

Downward compatibility

Downward compatibility is a contract between the application and the underlying software. A downwardly compatible application is built using only functions that are supported in both present and earlier releases of the software, and the software guaranteed the constancy of the interfaces and behavior from release to release.

WebSphere MQ does not support downwardly compatible applications, though typically applications rebuilt against a later release, and redeployed to a queue manager running at an earlier release do work in practice, as long as they use no new function.

Downward version interoperability is a contract between WebSphere MQ applications, WebSphere MQ clients and queue managers, and between queue managers. Applications, as part of the contract, need to restrict the functions they use to those provided in earlier versions of WebSphere MQ. An application built with a later version of WebSphere MQ can interoperate with an application built with and connected to an earlier version of WebSphere MQ.

A client application that is built to a later version of the WebSphere MQ client is not affected by being connected to an earlier version of the queue manager. It can only use the functions available in the earlier queue manager, of course.

Any WebSphere MQ application that uses only functions provided by an earlier version of a queue manager can continue to send messages to the earlier version when the application is rebuilt and connected to a later version of WebSphere MQ.

When you migrate (build) an application without any changes to a new level of queue manager, is it downwardly compatible?

1. Can you run a WebSphere MQ server application, built with a later WebSphere MQ version, on a server with an earlier queue manager installed?

2. Can you run a WebSphere MQ client application, built with a later WebSphere MQ version, on a client platform with an earlier client installation when connected to the later queue manager?
3. Can you run a WebSphere MQ client application, built with a later WebSphere MQ version, on a client platform with the later client installation when connected to an earlier queue manager?
4. Can you exchange messages between a WebSphere MQ client or server application, connected to a later queue manager, with applications connected to an earlier queue manager?

The answer to questions 1 and 2 is *no*, and to 3 and 4 is *yes*. WebSphere MQ programs are *downwardly version compatible*; they are *not downwardly compatible*. However the applications in scenario 1 and 2 usually do run without any errors being diagnosed if you do not use any new function. These are scenarios you need to avoid as they are not guaranteed to work all the time. If you find you have been running an incompatible configuration and you encounter a problem, you need to rebuild your applications with the earlier level of WebSphere MQ before continuing with problem diagnosis.

Suppose you decide to rebuild a *client* application. Can you deploy it to your production environment that contains some earlier versions of client and server platforms? The strict answer is no, you need to upgrade all the client machines you deploy to, at least to the version of the client you have built; you do not need to upgrade the queue managers. In practice the client should work as long as you have used no new function.

Suppose you deploy some WebSphere MQ queue managers at a new version level. You have an existing WebSphere MQ application that you use to send messages between the servers. Should you rebuild the application to deploy it onto the new servers, or should you deploy the old version onto the new servers? The answer is, either. You can continue to deploy the existing version of the application onto all your servers, or you can deploy the rebuilt application onto the new servers. Either configuration works. WebSphere MQ supports running the existing application on later servers (*upward compatibility*) and supports sending messages from later application versions to earlier ones (*downward version interoperability*). The one thing you cannot do is to rebuild the application and redeploy it onto both the earlier and newer servers, though that will probably work. The reason being that WebSphere MQ does not support *downward compatibility*.

The next section describes a model to help you understand the WebSphere MQ operating environment and think about how changes in one part of the environment might affect a WebSphere MQ application.

Operating environment

Figure 1 on page 6 shows two runtime operating system environments, with an WebSphere MQ server and server application installed in the environment called Server, and a WebSphere MQ client application in the environment called Client. The queue manager, *QM**, represents queue managers of various levels installed in other operating environments. The queue manager labelled *QM-n?* coexists on the same server as *QM*, but runs at a different release level. WebSphere MQ Migration information terms this a *coexistent* queue manager. Only z/OS supports multiple queue managers coexisting at different release levels in the same operating environment. The question-mark in the queue manager name indicates this

capability may not be present in your environment.

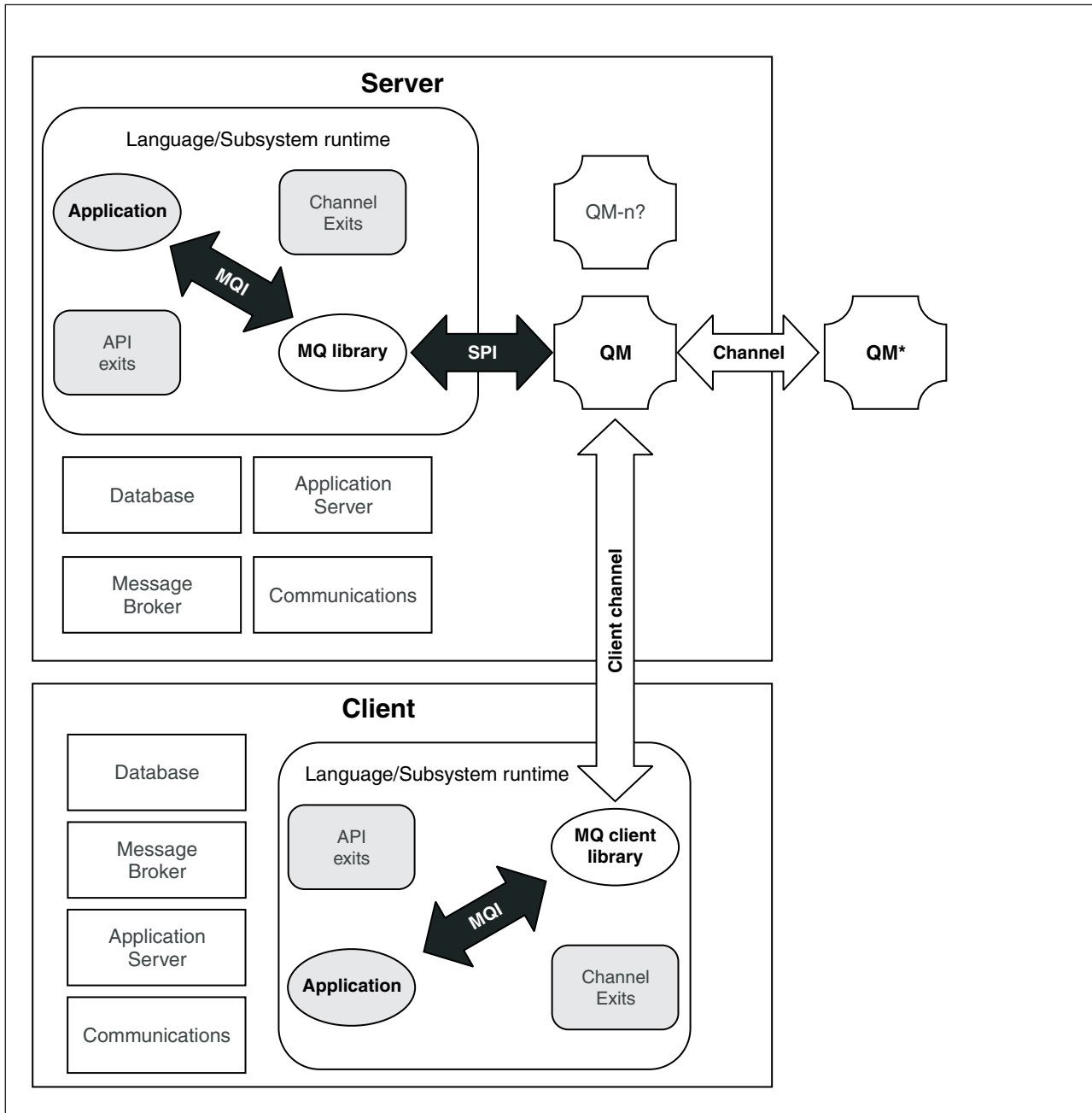


Figure 1. WebSphere MQ application migration model

WebSphere MQ has control over some, but not all, of interfaces shown in this model. It has complete control over an interface when it owns both sides of the interface, such as between an MQ channel and a queue manager. It shares control and use of the MQI interface and the exit interfaces with applications and it shares the control and use of operating environment interfaces with other software components. The diagram shows some major software components that share interfaces with WebSphere MQ, such the subsystem or language runtime environment, databases, application servers, communication subsystems and the message broker.

Using this model you can distinguish four types of migration that are discussed in the following sections.

Queue manager migration

Upgrading the Websphere MQ installation to a new version or release

WebSphere MQ client migration

Upgrading the Websphere MQ client installation to a new version or release. This is not the same as upgrading the client application itself.

Application migration

Any or all of relinking, recompiling or recoding a WebSphere MQ server or client application.

Environment migration

Upgrading the operating environment, or components in the environment such as installing a new level of JRE.

Queue manager migration

Queue manager migration takes place when you upgrade WebSphere MQ by installing a new release or version.

When you upgrade a queue manager, resources it owns such as the current log files are automatically migrated for you. The WebSphere MQ Migration information topic includes any additional migration tasks that you might need to do, particularly to exploit new functionality. Migration information also tells you about changes to default behavior, such as the introduction of channel sharing for WebSphere MQ client channels, which affects the way application client exit code operates.

Interfaces to WebSphere MQ resources that are shared between queue managers, such as channel interfaces, and the interfaces between WebSphere MQ libraries and the queue manager (labelled SPI in the diagram), are upward and downward compatible; as long as you change nothing else in the environment they require no migration. If do you choose to change something in the environment; for example to exploit the interface to IPv6 that was introduced in Version 6.0, there are migration tasks to perform that are explained in WebSphere MQ Migration information.

WebSphere MQ upgrades shared MQ libraries when the WebSphere MQ server is upgraded. Application linkages to the shared libraries do not normally change, so there is no need to relink an application to pick up the new libraries. Applications will pick up the new version of a library as soon as a new version of the shared libraries are loaded. Both ends of the SPI interface, the shared library and the queue manager are then at the new version level.

If you have copied the libraries to a different location, into an IDE for example, or if you have included the installation location of the libraries in a version control system, there is the possibility that application will continue to load the old rather than the new version of the libraries. As the SPI interface is upward and downward compatible the old libraries will probably continue to work and you might not notice a problem at first. Problems will become apparent either when you try to use new functions that are not supported in the earlier version, or when you apply fixes, and some fixes do not appear to work. Linking to libraries in anything other than their installed directory is an unsupported configuration, as is using server libraries at a different release level to the queue manager, though in practice, you might not notice any immediate problems.

WebSphere MQ client migration

WebSphere MQ client migration takes place when install a new release or version of the WebSphere MQ client. This is not the same as upgrading the client application itself.

WebSphere MQ clients are both upward and downward *version interoperable*, so there is little reason to upgrade a WebSphere MQ client, You might choose to do so when you are upgrading the client application, or when the WebSphere MQ client version goes out of service.

The WebSphere MQ client libraries (such as mqic.lib) are dynamic, and the application linkages to the libraries do not normally change. So there is no need to relink a client application to pick up new WebSphere client libraries. The client will pick up the library next time the library is loaded. Libraries should not be moved from their installed directory. Linking to libraries in anything other than their installed directory is an unsupported configuration.

Application migration

Application migration takes place when you rebuild your WebSphere MQ applications with a new version or release of the WebSphere MQ client or server tools. You might regard running an application against a new version of queue manager as application migration - but as long as the application has not been rebuilt with the new version of WebSphere MQ tools, this is termed queue manager migration.

Application migration is a two step process. The first step is taken when an application is rebuilt using a new version of the WebSphere MQ development tools. The tools contain source files, such as headers, and linkages to the WebSphere MQ libraries. The second step takes place when you decide to change the set of WebSphere MQ capabilities the application uses. The choice of capabilities a WebSphere MQ application uses is influenced not only by the common set of capabilities of the queue managers it connects with, but also indirectly by the capabilities of the queue managers it exchanges messages with.

When you upgrade a queue manager you can continue to build an application using existing development tools rather than choosing to migrate the application to the new WebSphere MQ version or release. When you take the step to build your application with the new version of WebSphere MQ, in rare cases you may need to make code changes to preserve the behavior of the application. For example, in Version 7.0 of WebSphere MQ the JMS classes throw different exceptions to those thrown in Version 6.0.

Typically for licensing, support, functionality or environment compatibility reasons you would expect to upgrade the development environment, to keep it in step with the latest installed server environment. Remember, you should not deploy client or server applications to a earlier version of the client or server platform, though you can write the application to be downward version interoperable if you want to communicate with earlier versions of the queue manager, or, in the case of client applications, to connect to an earlier version of the queue manager.

To help you write downward compatible applications that can exchange messages with earlier versions of WebSphere MQ, WebSphere MQ provides data type versioning. Data type versioning *assists* you in exchanging messages that are compatible with target queue managers. Even with data type versioning you can

still create messages that use new functions that are not compatible with all the queue managers an application interacts with.

You need to be aware of, and code to, the capabilities of all the queue managers the application interacts with. It requires planning and design to write an application that works with different versions of a queue manager. There is no API provided with WebSphere MQ to restrict an application to a function subset common to the set of queue managers it interacts with. To improve interoperability, some developers choose to provide an MQI wrapper layer, or use MQI API exits, to control the functionality programs use.

Client applications are more likely to connect to different queue managers than applications written for a specific server. Plan carefully when writing an application that is to connect to different versions of a queue manager, and even more so if it connects to queue managers on different platforms. The default values of some WebSphere constants, such as `PMO_SYNCPOINT`, `PMO_NO_SYNCPOINT` differ between platforms.

The following three tips are useful to consider when writing WebSphere MQ programs to run on multiple versions of the same platform.

- Use explicit MQSeries datatype versioning. Rather than use `datatype_CURRENT_VERSION`, set the version to use explicitly. You can then decide to upgrade to the new version of a datatype in line with the "lowest common denominator" queue manager.
- Review the default values provided by WebSphere MQ to initialize datatypes, and provide your own where appropriate.
- Restrict programmers to the earlier function set and consider continuing to use the tools from the earliest version of WebSphere MQ in operation.

Environment migration

Environment migration takes place when some aspect of the operating system environment is upgraded. Many changes in the operating environment do not affect WebSphere MQ. In some cases operating environment changes might require queue manager or even application migration. More commonly, changes in the operating environment create the *possibility* of exploiting new functionality in your applications and in WebSphere MQ.

To keep queue managers in step with current software environments, supported environments change with every major new release of WebSphere MQ. It is these changes, rather than changes in WebSphere MQ itself which may necessitate changes, recompilation or relinkage of your applications. You do not normally need to migrate an application to the new version of the queue manager to accommodate the changes in the software environment, and often you can continue to build it with an older version of the WebSphere MQ development tools until you are ready to migrate the application to exploit new capabilities. In practice, you might well choose to migrate the application without making any immediate functional changes to keep your development environment current.

A (rare) example where environment migration forces you to migrate applications without having a functional requirement to do so might be as follows. You have chosen to upgrade a queue manager for some operational reason. The new version of the queue manager requires a new version of the operating environment. The operating environment does not support the linkage to the old language runtime,

so applications have to be rebuilt with new linkages and use the new language runtime.

Summary

Application migration is rarely required when queue managers are upgraded.

- WebSphere MQ applications are *upwardly compatible*. They do not need to be rebuilt to work with later releases of queue managers.
- WebSphere MQ clients and queue managers are *upwardly and downwardly version interoperable*.

When you do migrate an application to a later version of queue manager, use MQ data type versioning to make your applications downwardly version interoperable. Read the Migration information topic to see if there are any changes in a new release of WebSphere MQ that affects the way your application behaves when you rebuild it. Remember, you should not deploy an application built at a later level of WebSphere MQ to a client or server platform running at an earlier release level.

Definition of upward and downward compatibility

Programs that are created on the current release and run on future releases of software are upwardly compatible, and those that also run on earlier releases are downward compatible.

Upward and downward compatibility are directional terms, and to avoid ambiguity it is important to get the direction clear. By saying application A is upwardly compatible with respect to B, we mean that A_1 works with B_1 and any higher numbered release of B. This implies B is downward compatible with A, that is B_n works with A_n , and any lower numbered version of the application. Table 1 illustrates the difference between upward and downward compatibility.

Table 1. Upward and downward compatibility

	B_1	B_2
A_1	-	A is upwardly compatible B is downwardly compatible
A_2	A is downwardly compatible B is upwardly compatible	-

If we say that WebSphere MQ supports upward compatibility, what do we mean? Probably a number of things; for example, that function calls in earlier releases will work in later releases, but not vice versa. In that case substitute function calls for A in Table 1, and WebSphere MQ queue managers for B. But we might also mean applications built against an earlier WebSphere MQ release work without rebuilding against a later release. In which case A is the application executable built using function calls from an earlier release of WebSphere MQ, and B is both the shared library (mqm.lib or mqic.lib for example) and the queue manager that A is run against.

The terms upward and downward compatibility are normally used in the context of the operating environment of a WebSphere MQ application, such as the question, "Can you connect an application built using one version of WebSphere MQ to a queue manager running at a different version?" The answer is basically "yes", but it depends on what the operating environment is. For example, for you need to consider the compatibility between a network client application and its

| remote server environment; such as between a WebSphere MQ client and the
| queue manager connected by a client channel, or between one queue manager and
| another, connected by a message channel. This form of compatibility is normally
| called *version interoperability* and you can talk about upward and downward
| version interoperability. Table 1 on page 10 describes the version interoperability of
| messages exchanged between queue managers, and between WebSphere MQ
| clients and queue managers.

| Consider the case of a WebSphere MQ client application built against the version 6
| of WebSphere MQ. You might upgrade both the WebSphere MQ server and client
| software to v7, or just upgrade the server, and leave the client unchanged. If you
| just upgrade the server, then the v6 WebSphere client library, such as mqic.lib,
| needs to be *upwardly version interoperable* with the server running a v7 queue
| manager. If you upgrade both the client and the server, then the application built
| against the v6 client library needs to be *upwardly compatible* to run against the v7
| client library without rebuilding. Fortunately both the upward compatibility
| requirement and the upward version interoperability requirements are met by
| WebSphere MQ clients.

Chapter 3. Coexistence

For the purposes of this section, coexistence is defined as the ability of two or more different versions of WebSphere MQ to function on the same computer. Two or more different versions of WebSphere MQ cannot coexist on the same computer, except for WebSphere MQ for z/OS where multiple different versions can coexist in a limited number of scenarios.

Coexistence on WebSphere MQ for AIX

WebSphere MQ for AIX® Version 7.0 does not coexist with previous versions of WebSphere MQ. You must migrate from either WebSphere MQ Version 5.3 or WebSphere MQ Version 6.0 to WebSphere MQ Version 7.0. You do not have to uninstall your current version before installing WebSphere MQ Version 7.0 because the installation process does it for you.

Coexistence on WebSphere MQ for HP-UX

WebSphere MQ for HP-UX Version 7.0 does not coexist with previous versions of WebSphere MQ. You must migrate from either WebSphere MQ Version 5.3 or WebSphere MQ Version 6.0 to WebSphere MQ Version 7.0. If you are migrating from a previous version of WebSphere MQ for HP-UX, you must uninstall your current version before installing WebSphere MQ Version 7.0.

Uninstalling WebSphere MQ

About this task

To uninstall (server or client), use the HP-UX `swremove` program.

1. Stop all WebSphere MQ applications on the machine and the applications accessing remote machines.
2. Before starting to uninstall, end all WebSphere MQ activity.
 - a. Log in as a user in group `mqm`.
 - b. Use the `dspmqs` command to display the state of all the queue managers on the system.
 - c. Use the `endmqm` command to stop all running queue managers.
 - d. Stop any listeners associated with the queue managers, using the command:

```
endmq1sr -m QMgrName
```
 - e. To check that you have stopped all of them, enter the following:

```
ps -ef | grep mq
```
 - f. Check that there are no processes listed that are running command lines beginning `amq` or `runmq`. Ignore any that start with `amqi`.
3. Log in as root.
4. Enter `swremove MQSERIES` to uninstall the MQSERIES package.
 - a. Log in as root.
 - b. Enter `swremove MQSERIES` to uninstall the MQSERIES package.
5. After uninstalling WebSphere MQ, you can delete the `/var/mqm` directory tree, (this will destroy all queue managers and their associated data).
6. If no other products require the Global Security Kit Version 7, you can uninstall packages `gsk7bas` and `gsk7bas64` by typing:

```
swremove gsk7bas
```

or

```
swremove gsk7bas64
```

Coexistence on WebSphere MQ for i5/OS

WebSphere MQ for i5/OS[®] Version 7.0 does not coexist with previous versions of WebSphere MQ.

You must migrate from either WebSphere MQ Version 5.3 or WebSphere MQ Version 6.0 to WebSphere MQ Version 7.0. You cannot install two versions of WebSphere MQ or MQSeries[®] on the same logical partition. If you try to do this, WebSphere MQ assumes that a slip install (an upgrade on the same machine) is being performed. For further guidance on performing a slip install, see *WebSphere MQ for i5/OS Quick Beginnings*.

The WebSphere MQ Classes for Java[™] do not coexist with any previous versions of the WebSphere MQ Classes for Java or the MA88 SupportPac[™]: the same slip install process applies to this licensed program.

WebSphere MQ for i5/OS can coexist with any version of WebSphere MQ Classes for Java; you do not need to install a version of WebSphere MQ Classes for Java that matches the installed version of WebSphere MQ.

Coexistence on WebSphere MQ for Linux

WebSphere MQ for Linux[®] Version 7.0 does not coexist with previous versions of WebSphere MQ. You must migrate from either WebSphere MQ Version 5.3 or WebSphere MQ Version 6.0 to WebSphere MQ Version 7.0. If you are migrating from a previous version of WebSphere MQ for Linux, you must uninstall your current version before installing WebSphere MQ Version 7.0.

Uninstalling Websphere MQ

This chapter explains how to remove WebSphere MQ for Linux from your system.

1. Stop all WebSphere MQ applications on the machine and the applications accessing remote machines.
2. End all WebSphere MQ activity:
 - a. Log in as a user from group mqm.
 - b. Use the **dspmq** command to display the state of all the queue managers on the system.
 - c. Use the **endmqm** command to stop all running queue managers.
 - d. Stop any listeners associated with the queue managers, using the command:

```
endmqlsr -m QMgrName
```
 - e. To check that you have stopped all necessary WebSphere MQ activity, type:

```
ps -ef | grep mq
```

Check that there are no processes listed that are running command lines beginning amq or runmq. Ignore any that start with amqi.

3. Log in as root.
4. Before you uninstall WebSphere MQ for Linux you must find out the names of the packages (components) currently installed on your system. To list all the packages with their version information, enter the following:


```
rpm -qa | grep MQSeries
rpm -qa | grep ibm-java
rpm -qa | grep gsk
```

Note: If you have applied maintenance to the install, remove these maintenance packages before removing the base install.

5. Some of the installed packages are dependent on others. The **rpm** command will not remove a package if others are dependent on it. For this reason you must uninstall the packages in such an order that each one you uninstall has no dependencies from other packages. To list all of the packages on which a named package (for example MQSeriesServer) depends, enter the following:

```
rpm -q --requires MQSeriesServer
```

Alternatively, remove all the components at the same time by appending all the package names to the rpm command arguments. For example, to remove the runtime, Server and SDK components, enter the following:

```
rpm -ev MQSeriesRuntime MQSeriesServer MQSeriesSDK
```

6. After uninstalling , you might want to delete the contents of /var/mqm, which is not removed by the uninstallation. **Only do this if you do not need your queue manager data.** For example, do not delete the content if you plan to migrate to a later version of WebSphere MQ.

Results

You have now uninstalled WebSphere MQ.

Coexistence on WebSphere MQ for Solaris

WebSphere MQ for Solaris Version 7.0 does not coexist with previous versions of WebSphere MQ. You must migrate from either WebSphere MQ Version 5.3 or WebSphere MQ Version 6.0 to WebSphere MQ Version 7.0. If you are migrating from a previous version of WebSphere MQ for Solaris, you must uninstall your current version before installing WebSphere MQ Version 7.0.

Uninstalling WebSphere MQ

About this task

To uninstall WebSphere MQ (server or client), use the Solaris **pkgrm** program.

Note: If any updates have been applied, remove them first.

1. Stop all WebSphere MQ MQ applications on the machine and the applications accessing remote machines.
2. End all WebSphere MQ activity:
 - a. Log in as a user in group mqm.
 - b. Use the **dspm** command to display the state of all the queue managers on the system.
 - c. Use the **endmqm** command to stop all running queue managers.
 - d. Stop any listeners associated with the queue managers, using the command:

```
endmq1sr -m QMgrName
```
 - e. To check that you have stopped all of them, enter the following:

```
ps -ef | grep mq
```
 - f. Check that there are no processes listed that are running command lines beginning amq or runmq. Ignore any that start with amqi.

3. Log in as root.
4. Enter `pkgrm mqm` to uninstall the `mqm` package.

Note: If the product was successfully uninstalled, you can delete the files and directories contained in `/opt/mqm`. If for any reason the product was not properly uninstalled, files might still exist in the package.

5. After uninstalling WebSphere MQ, you can delete the `/var/mqm` directory tree, (this will destroy all queue managers and their associated data).
6. If no other products require the Global Security Kit Version 7, you can uninstall packages `gsk7bas` and `gsk7bas64`.

Coexistence on WebSphere MQ for Windows

WebSphere MQ for Windows[®] Version 7.0 does not coexist with previous versions of WebSphere MQ. You must migrate from either WebSphere MQ Version 5.3 or WebSphere MQ Version 6.0 to WebSphere MQ Version 7.0. You do not have to uninstall your current version before installing WebSphere MQ Version 7.0 because the installation process does it for you.

Coexistence on WebSphere MQ for z/OS

You must take a number of factors into consideration when WebSphere MQ Version 7.0 for z/OS coexists with previous versions of WebSphere MQ for z/OS.

Multiple queue manager versions in z/OS

There can be several WebSphere MQ subsystems in a z/OS image, and they can use different versions of WebSphere MQ, provided the WebSphere MQ early code modules are of the latest version being used. (These modules are loaded at z/OS IPL time and are shared among all the WebSphere MQ subsystems in the z/OS image.)

There can be multiple queue managers in a z/OS image at different levels of WebSphere MQ. This means that you can run one queue manager with Version 7.0 and another in the same image with an earlier version, provided that the early code is that of Version 7.0.

Each WebSphere MQ system has its own security settings, and can use mixed case RACF[®] class support if it is at the appropriate level of WebSphere MQ. Follow the migration steps in “Migration to mixed case security” on page 53.

Use STEPLIBs to control which level of WebSphere MQ is used.

Multiple queue manager versions in a queue-sharing group

A queue-sharing group can have Version 6.0 and Version 7.0 queue managers active and accessing shared queues and other shared objects.

The Version 6.0 queue managers must have a Version 7.0 coexistence PTF applied to enable them to run in a queue-sharing group with Version 7.0 queue managers.

After the Version 7.0 coexistence PTF has been applied the Version 6.0 queue manager must be started at least one time to allow the migration of MQ internal objects.

A Version 7.0 queue manager cannot coexist in a queue-sharing group with other queue managers running code earlier than Version 6.0. If there is a Version 5.3.1 queue manager in the queue-sharing group, or the queue-sharing group migration actions have not been performed (see “Migrating queue-sharing groups to Version 7.0” on page 65), you cannot start a Version 7.0 queue manager as a member of that queue-sharing group. After you have performed queue-sharing group migration actions, or if there are Version 7.0 queue managers active in the queue-sharing group, you cannot start queue managers at Version 5.3.1 or lower in the queue-sharing group. Version 6.0 queue managers will not start in such a queue-sharing group unless they have had the Version 7.0 coexistence PTF applied.

You are recommended to only have a mixed version queue-sharing group for the time it takes to migrate all queue managers to Version 7.0. While the queue-sharing group contains mixed version queue managers, WebSphere MQ for z/OS Version 7.0 allows prototyping with new Version 7.0 facilities on a Version 7.0 queue manager, and tolerates operation at the Version 6.0 level or above.

Function restrictions in a mixed queue-sharing group

You cannot alter a CF structure object from **CFLEVEL(3)** to **CFLEVEL(4)** until all queue managers in the queue-sharing group have been started at Version 6.0 or later level.

You cannot delete a CF structure object until all queue managers in the queue-sharing group have been started at Version 6.0 or later level.

Queue managers below Version 6.0 cannot connect to the Coupling Facility structure identified by the **CFLEVEL(4)** CF structure object, which means they can neither access the queues defined on it, nor messages stored on the queue.

You can define and alter objects with **QSGDISP(GROUP)** from a Version 7.0 queue manager. Those objects and their resulting copy objects are accessible on all the queue managers, but on older version queue managers the new Version 7.0 attributes and values are not available.

If you define a queue alias object with the properties **QSGDISP(GROUP)** and **TARGETYPE(TOPIC)** from a V7.0 queue manager, the **TARGETYPE** property is not recognized by the Version 6.0 queue manager. This will result in the Version 6.0 queue manager defining a copy object of the queue alias which is interpreted as referencing a queue instead of the intended topic.

On a Version 6.0 queue manager, MQSC commands using new Version 7.0 keywords and attribute values (but not new commands) can be entered for routing to a Version 7.0 queue manager using **CMDSCOPE**. Such commands, on whatever version queue manager, routed to Version 6.0 queue manager using **CMDSCOPE** will fail.

The operations and control panels work as detailed in “Operations and control panels” on page 19.

PCF commands can be entered only on a Version 6.0 or higher queue manager. If they use **MQCACF_COMMAND_SCOPE** they will fail on any queue manager they are routed to that is not at Version 6.0 or later.

Mixed case RACF support for multiple queue manager versions in a queue-sharing group

How you turn on mixed-case RACF support varies according to whether you have activated queue manager level checking or queue sharing group level checking, or both.

Mixed-case RACF security is supported in WebSphere MQ Version 7.0. Ensure that you have installed a level of the security product that supports mixed case security, that you have applied any updates required by WebSphere MQ, and that the new WebSphere MQ RACF® classes are installed and active. Because a queue-sharing group can contain a mix of WebSphere MQ Version 7.0 queue managers and WebSphere MQ Version 6.0 queue managers, you need to consider the following situations:

- “Mixed-case RACF support with queue manager level checking active”
- “Mixed-case RACF support with queue sharing group level checking active”
- “Mixed-case RACF support with queue manager and queue sharing group level checking active” on page 19

Mixed-case RACF support with queue manager level checking active: There are multiple queue managers in a queue-sharing group running on different versions of WebSphere MQ, and they all have queue manager only security checking active.

The process for turning on mixed-case RACF support for those queue managers that are running on WebSphere MQ Version 7.0 with mixed-case security is the same as when the queue managers are not in a queue-sharing group. Follow the migration steps in Migration to mixed-case security.

Mixed-case RACF support with queue sharing group level checking active: Multiple queue managers in a queue-sharing group are running on different versions of WebSphere MQ, and they all have queue-sharing group only security checking active.

To turn on mixed-case RACF support for those queue managers that are running on WebSphere MQ Version 7.0 with mixed-case security, see Migration to mixed-case security:

1. Carry out Steps 1-3.
2. Carry out Steps 4-6 for each queue manager that you are changing.
3. Carry out Step 7 for all queue managers in the queue-sharing group.
4. Carry out Step 8, but with the following additional consideration:

If you are defining a shared mixed-case resource, you can protect it with a mixed-case profile in a mixed-case class on those queue managers that are using that support. On queue managers that are not using that support, the resource can only be protected using a generic uppercase profile in an uppercase class.

In summary, queue managers using mixed-case support perform security checks against the mixed-case classes and queue managers using uppercase support (whatever the release of the product) perform security checks against the uppercase classes.

Carry out the above procedure if you have a shared mixed case resource, protected through a mixed case profile, but with some queue managers using SCYCASE(UPPER).

Mixed-case RACF support with queue manager and queue sharing group level checking active: There are multiple queue managers in a queue-sharing group running on different versions of WebSphere MQ, and they all have queue manager and queue-sharing group security checking active.

To turn on mixed-case RACF support for those queue managers that are running on WebSphere MQ Version 7.0 with mixed-case security, see “Mixed-case RACF support with queue sharing group level checking active” on page 18.

Operations and control panels

When using the operations and control panels, the WebSphere MQ libraries you use in ISPF must be compatible with those of the queue manager you are working with.

Table 2. Compatibility of queue manager versions with operations and control panels versions

Panel level	Version 7 queue manager	Version 6 queue manager
Version 7	X	X with restrictions and warnings
Version 6	X with warning, if migration & coexistence PTF applied	X

The panels at Version 7.0 level work with Version 7.0 queue managers. They also work, with some restrictions, with Version 6.0 queue managers, or with a queue-sharing group containing a mixture of queue managers. If you are using the panels with a queue manager earlier than Version 7, messages warn you about the restrictions. The Version 7.0 panels do not work with other levels of queue manager. If you have applied the migration & coexistence PTF, panels at the Version 6 level work as for Version 7.0; if they are used with a Version 7.0 queue manager they will give a warning, which can be ignored. If you have not applied the PTF, they will not work with a Version 7.0 queue manager. This is summarized in Table 2.

Application stubs

The stub modules that are link-edited with applications and exits (CSQASTUB, CSQBRSSI, CSQBRSTB, CSQBSTUB, CSQCSTUB, CSQQSTUB, and CSQXSTUB) might not work with earlier versions of the queue manager. For example, stubs supplied with Version 6 can be used by applications running on a Version 6 or Version 7 queue manager.

To take advantage of the new APIs introduced at WebSphere MQ V7, for example MQSUB and MQCB, you must link-edit your application with the appropriate V7 stub. Such an application will not run on an earlier version of the queue manager.

Coexistence of WebSphere MQ classes for JMS

The following are specific JMS coexistence considerations for WebSphere MQ Version 7.0.

Administered objects

WebSphere MQ Version 7.0 clients can read a WebSphere MQ Version 6.0 connection factory or destination object.

An object written under WebSphere MQ Version 7.0 can be used by a WebSphere MQ Version 6.0 client, but any of the new WebSphere MQ Version 7.0 properties that have been set are ignored.

For client migration information, see WebSphere MQ classes for JMS and classes for Java considerations.

Chapter 4. Migrating WebSphere MQ components to Version 7.0

If you are migrating WebSphere MQ components to Version 7.0, this section contains specific advice.

Migrating a queue manager cluster to WebSphere MQ Version 7.0.

This topic describes how to migrate a queue manager cluster.

About this task

You must consider the following issues, which relate to migrating a queue manager cluster:

- Minimizing application outages
- Measuring and verifying migration success and planning for backward migration if there are any migration problems
- Taking advantage of new WebSphere MQ features
- Managing the migration of a cluster in the context of the wider WebSphere MQ network and your organization's systems architecture

Techniques for migrating a cluster with minimal planned and unplanned outages are detailed in the following topics:

Migrating a queue manager cluster: Creating a plan

Before carrying out a migration, plan what you will do.

1. Consider general queue manager migration issues, issues specific to clustering, your system architecture, and your installation's change control policies.
2. Decide whether you will carry out a staged migration, or migrate all queue managers at the same time.
3. Document and test the plan before migrating production queue managers.

Migrating a queue manager cluster: Staged migration

In a staged migration, queue managers are migrated one at a time over an extended period.

Cluster queue managers can participate in clusters with other queue managers running at different versions, which is why a staged migration is possible.

When you migrate a cluster, it is good practice to stage the migration. You can leave several days between migrating each queue manager in the cluster, to test applications before wholly committing all queue managers to run at the new version.

It is advisable to migrate full repositories first. If all full repositories are migrated before partial repositories, all queue managers running at the new version can exploit features of the new version. Queue managers at the back-level version cannot take advantage of new features, and will appear in the repositories of new-version queue managers with the default values for any new attributes.

If full repositories are not migrated before partial repositories, the cluster continues to work but not all new-version queue managers are able to take advantage of new features. This is because of the manner in which cluster objects are published to the cluster, via full repositories, as shown in this example:

Table 3. Queue managers, repositories and cluster queues

Queue manager name	Full or partial repository	Cluster queues
QM1	Full	None
QM2	Full	None
QM3	Partial	Q1
QM4	Partial	Q1
QM5	Partial	None

Table 1 shows five queue managers that belong to the same cluster. All queue managers are currently running WebSphere MQ Version 6.0 but you are migrating the cluster to WebSphere MQ Version 7.0. You choose to migrate all the partial repositories (QM3, QM4, and QM5) first. You want to use a new feature of WebSphere MQ Version 7.0, referred to here as NEWFEATURE, so you alter the cluster-receiver channel on QM3 as follows:

```
ALTER CHL(TO.QM3) CHLTYPE(CLUSRCVR) NEWFEATURE(5)
```

When you run this command, QM3 publishes the change to the full repositories. Each full repository updates its local repository and forwards the original update from QM3 to queue managers that have subscribed to TO.QM3. QM5 has subscribed to TO.QM3, because applications connected to QM5 have previously put messages to Q1. At this point, the full repositories now hold QM3's changed cluster receiver record, but as the full repositories are running at Version 6.0, the record is a Version 6.0 record, and therefore does not contain the NEWFEATURE attribute.

QM5 holds QM3's changed cluster receiver record, which includes the NEWFEATURE attribute, because QM5 is running at Version 7.0. Although QM5 received the updated record via the back-level full repositories, it has received the Version 7.0 attribute data. Full repositories are able to forward publications, so they can publish records with attributes that they themselves do not support. Executing DIS CLUSQMGR(*) NEWFEATURE on QM5 returns:

```
QM1 NEWFEATURE(0)
QM2 NEWFEATURE(0)
QM3 NEWFEATURE(5)
QM4 NEWFEATURE(0)
QM5 NEWFEATURE(0)
```

As this example shows, all the partial repositories can be upgraded before the full repositories and still take advantage of the new function.

Now you add a new Version 7.0 queue manager (QM6) to the cluster, and an application connects to it and opens Q1. At this point QM6 asks the full repositories about Q1 and where it is hosted. The full repositories publish information for Q1 and the queue managers that host it (QM3 and QM4) to QM6. Since the full repositories hold only Version 6.0 records, the records they send to QM6 do not contain NEWFEATURE attributes. At this point, executing DIS CLUSQMGR(*) NEWFEATURE on QM6 would return:

QM1 NEWFEATURE(0)
QM2 NEWFEATURE(0)
QM3 NEWFEATURE(0)
QM4 NEWFEATURE(0)
QM6 NEWFEATURE(0)

The NEWFEATURE value shown for QM3 (0) does not match the value on the TO.QM3 object, which is 5. This mismatch will be resolved the next time QM3 publishes its cluster receiver, as the full repositories will then forward the publication (including Version 7.0 attribute data) from QM3 straight on to QM6.

Mismatches like this, caused by the back-level full repositories publishing direct from their repositories, can also be caused by opening a queue for the first time or by issuing the REFRESH CLUSTER on a partial repository.

Migrating one cluster queue manager

This is an example of a basic migration plan for a cluster queue manager. Determine the order in which you need to migrate the queue managers, then use it as a basis for a procedure that fits your own requirements. Your procedure might have more or fewer steps:

1. Suspend the queue manager that you want to migrate from the cluster:
 - a. Issue the MQSC command `SUSPEND QMGR CLUSTER(cluster name)`
 - b. Monitor traffic to the suspended queue manager. The cluster workload algorithm can choose a suspended queue manager if there are no other valid destinations available or an application has affinity with a particular queue manager. You might have to close the application submitting messages to this queue manager.
2. Save a record of all cluster objects known by this queue manager. This data will be used after migration to check that objects have been migrated successfully.
 - a. Issue the command `DISPLAY CLUSQMGR(*)` to view cluster queue managers.
 - b. Issue the command `DISPLAY QC(*)` to view cluster queues.
3. Save a record of the full repositories' view of the cluster objects owned by this queue manager. This data will be used after migration to check that objects have been migrated successfully.
 - a. Issue the command `DISPLAY CLUSQMGR(migrated queue manager name)` on the full repositories.
 - b. Issue the command `DISPLAY QC(*) WHERE(CLUSQMGR EQ migrated queue manager name)` on the full repositories
4. Stop the queue manager.
5. Take a backup of the queue manager, as described in *WebSphere MQ System Administration Guide*.
6. Install the new version of WebSphere MQ.
7. Restart the queue manager.
8. Ensure that all cluster objects have been migrated successfully.
 - a. Issue the command `DISPLAY CLUSQMGR(*)` to view cluster queue managers, and check the output against the data saved before migration.
 - b. Issue the command `DISPLAY QC(*)` to view cluster queues and check the output against the data saved before migration.
9. Ensure that the queue manager is communicating with the full repositories correctly. Check that cluster channels to full repositories can start.

10. Ensure that the full repositories still know about the migrated cluster queue manager and its cluster queues.
 - a. Issue the command `DISPLAY CLUSQMGR(migrated_queue_manager_name)` on the full repositories and check the output against the data saved before migration.
 - b. Issue the command `DISPLAY QC(*) WHERE (CLUSQMGR EQ migrated_queue_manager_name)` on the full repositories and check the output against the data saved before migration.
11. Test that applications on other queue managers can put messages to the migrated cluster queue manager's queues.
12. Test that applications on the migrated queue manager can put messages to the other cluster queue managers' queues.
13. Resume the queue manager.
 - a. Issue `RESUME QMGR CLUSTER(cluster name)`
14. Closely monitor the queue manager and applications in the cluster for a period of time.

Migrating a queue manager cluster: Creating a backout plan

Before performing a migration, decide on a backout plan in case of failure.

The backout plan should detail what constitutes a successful migration, the conditions that trigger the backout procedure, and the backout procedure itself. The procedure could involve removing or suspending the queue manager from the cluster, backward migration, or keeping the queue manager offline until an external issue is resolved.

Migrating a queue manager cluster: Setting up a test environment

When planning to migrate a cluster, set up a test environment.

When you make any system changes, it is good practice to test the changes in a test or quality-assurance environment before implementing the changes in a production system.

Ideally, execute an identical migration plan in test and production environments to maximize the chance of finding potential problems in test rather than in production. In practice, test and production environments are unlikely to be configured identically or to have the same workloads, so the migration steps carried out in test will not exactly match those carried out in production. Whether the plans and environments for test and production differ or not, it is always possible to find problems when migrating the production cluster queue managers. Do not assume that a problem-free test migration indicates a problem-free production migration.

Migrating the test system

Migrate each queue manager in the test system.

Before you begin

For each queue manager in the test system, in the order defined in the migration plan you developed in "Migrating a queue manager cluster: Creating a plan" on page 21, migrate and test the queue manager.

Migrating a queue manager cluster: Migrating the production system

Migrate each queue manager in the production system.

Before you begin

For each queue manager in the production system, in the order defined in the migration plan you developed in “Migrating a queue manager cluster: Creating a plan” on page 21, migrate and test the queue manager.

Migrating client-connection and server-connection channels

This topic describes issues relating to migrating client-connection or server-connection channels to WebSphere MQ Version 7.0.

The default for client channels is to use sharing conversations, described in “Using sharing conversations” in *WebSphere MQ Clients*. On client channels only that use sharing conversations, heartbeats can flow across the channel (from both ends) at any time, depending on channel activity and the heartbeat interval. On versions of WebSphere MQ earlier than Version 7.0, or on WebSphere MQ Version 7.0 channel instances which are configured to not allow sharing conversations, heartbeats can flow on client channels only when an MQGET call is outstanding. This can affect whether you set heartbeating on client channels and what value you set it to.

The default value on a migrated client-connection or server-connection channel for the SHARECNV channel attribute is 10, and the default WebSphere MQ V7.0 sharing MQCNO value for an existing client application is MQCNO_ALL_CONVS_SHARE. Conversations are, therefore, shared on TCP/IP channel instances by default when you migrate to WebSphere MQ Version 7.0. This is not apparent, unless you have send, receive, or security exits written to perform actions that are not generally expected in send, receive or security exits. In particular:

- If send or receive exits alter the MQCD structure on an MQXR_INIT call, the effect of these exits will differ, depending on whether they are on the first, or subsequent conversations on a channel instance with sharing conversations:
 - If the MQCXP SharingConversations field is set to FALSE, this exit instance applies to the first conversation on the channel instance. No other exit can be changing the MQCD at the same time and changes that are made to the MQCD can affect the way that the channel runs.
 - If the MQCXP SharingConversations field is set to TRUE, this exit instance applies to a conversation that is sharing the channel instance with other conversations. Changes made to the MQCD in the exit instance are retained in the MQCD but cannot affect the way the channel runs.
- If send, receive, or security exit instances alter the MQCD, when the MQCXP SharingConversations field is set to TRUE, exit instances on other conversations can be changing the MQCD at the same time and updates can be overwritten; this does not affect the way the channel runs, but if you use the data in the MQCD for a different purpose that requires the integrity of the data to be preserved, then it might be necessary to serialize access to the MQCD across these different exit instances.
- For further information on the MQCXP SharingConversations field, see “Using sharing conversations” in *WebSphere MQ Clients*.
- If send or receive exits on the server-connection side of the channel instance need to perform MQI calls on the connection with which they are associated,

they use the connection handle provided in the MQCXP Hconn field. You must be aware that client-connection send and receive exits cannot make MQI calls.

Performance implications of sharing conversations on client-connection channels

The SHARECNV channel attribute specifies the maximum number of conversations that share each TCP/IP channel instance. Sharing conversations has performance implications when migrating client-connection channels to WebSphere MQ Version 7.0.

A SHARECNV channel attribute of 0 specifies no sharing of conversations over a TCP/IP channel instance. The channel instance runs in a mode similar to that of WebSphere MQ Version 6.0 and inhibits the following WebSphere MQ Version 7.0 features:

- Administrator stop-quiet
- Heartbeating
- Read ahead
- Client asynchronous consume

Established client applications that do not need to use the WebSphere MQ Version 7.0 features can use SHARECNV(0). Using SHARECNV(0), processing of messages is, on average, 3 percent slower than under WebSphere MQ Version 6.0.

A SHARECNV channel attribute value of 1 specifies no sharing of conversations over a TCP/IP channel instance but makes the WebSphere MQ Version 7.0 features available.

A SHARECNV channel attribute value greater than 1 specifies sharing of conversations over a TCP/IP channel instance and makes the WebSphere MQ Version 7.0 features available.

The default value on a migrated client-connection channel for the SHARECNV channel attribute is 10, and the default WebSphere MQ V7.0 sharing MQCNO value for an existing client application is MQCNO_ALL_CONVS_SHARE. Conversations are, therefore, shared on TCP/IP channel instances by default when you migrate to WebSphere MQ Version 7.0.

All of the conversations on a socket are received by the same thread. High SHARECNV limits have the advantage of reducing queue manager thread usage. However, if a large number of conversations sharing a socket are all busy, there is a possibility of delays as the conversations contend with one another to use the receiving thread. In this situation a lower SHARECNV value is better.

On average, processing of messages from client applications is 15 percent slower when using SHARECNV(10), compared with SHARECNV(0); this is the cost of providing the client function that is new in WebSphere MQ Version 7.0. However, performance improvements introduced in WebSphere MQ Version 7.0 on distributed platforms and the availability of WebSphere MQ Version 7.0 performance features such as read ahead and client asynchronous consume, can mitigate this performance difference.

You must bear in mind the preceding points when optimizing the SHARECNV value for your enterprise.

Migrating JMS Exception listeners

JMS exception listeners in WebSphere MQ Version 7.0 behave slightly differently than they did in previous releases.

In versions of WebSphere MQ earlier than 7.0, an exception listener was called only to inform the application of any error condition that occurred asynchronously to the application's execution, for example, during the processing of a message for an asynchronous consumer where the application might have no other means to discover the exception. This could include connection broken events or if an attempt was made to process an unreadable message.

APAR IY81774 introduced a system property `activateExceptionListener`. If this property was set, all exceptions resulting from a broken connection were sent to the exception listener, regardless of the context in which they occurred. Therefore, in addition to being called for exceptions that occurred asynchronously to the application's execution, the exception listener was also passed any exceptions that indicated a connection broken event that were discovered during a normal synchronous JMS API call. For example, if a connection broken event occurred during a call to "receive", then the exception would not only be thrown from the receive method, but would also be passed back to the exception listener.

In WebSphere MQ Version 7.0, if the `ASYNC_EXCEPTIONS` property of the factory object is set to its default value of `ASYNC_EXCEPTIONS_ALL`, the exception listener is called for all broken connection exceptions and all exceptions that arise outside the scope of a synchronous JMS API call. Setting `ASYNC_EXCEPTIONS` property to `ASYNC_EXCEPTIONS_ALL` provides the same behavior as system property `activateExceptionListener` that was introduced in APAR IY81774. The property `activateExceptionListener` is therefore deprecated.

If the `ASYNC_EXCEPTIONS` property is set to `ASYNC_EXCEPTIONS_CONNECTIONBROKEN`, only exceptions indicating a broken connection are sent to the exception listener. These exceptions include connection broken exceptions occurring both synchronously and asynchronously, but do not include any other asynchronous errors such as for unreadable messages. In this mode, if the exception listener is triggered, the connection can be considered to have failed and it is no longer possible to send or receive messages.

Your applications might therefore receive more or fewer exceptions than previously. Ensure that the applications take appropriate action, for example, by attempting to remake the connection, when such exception listener calls are made.

For information about handling exceptions, see *WebSphere MQ Using Java*.

Change in behavior of `MQMessage.resizeBuffer` in WebSphere MQ classes for Java

In earlier releases of WebSphere MQ, attempting to resize a buffer to a negative size using `MQMessage.resizeBuffer` in a Java program resulted in a `NegativeArraySizeException`. In WebSphere MQ Version 7.0, it results in an `IllegalArgumentException`.

WebSphere MQ Version 7.0 transport for SOAP

This topic provides information about migrating to WebSphere MQ Version 7.0 transport for SOAP.

The level of Apache Axis has been updated from Version 1.1 to Version 1.4 and the supported level of the Microsoft® .NET Framework has been updated from Version 1.1 to Version 2.0. The MA0V (Asynchronous support in WebSphere MQ transport for SOAP) SupportPac has been updated correspondingly for the asynchronous function.

MQCONN reason codes

WebSphere MQ Version 7.0 provides some more specific reason codes for failed MQCONN calls.

Situations in which a MQCONN call failed to connect to a queue manager in WebSphere MQ Version 6.0, with reason code MQRC_Q_MGR_NOT_AVAILABLE, can still output reason code MQRC_Q_MGR_NOT_AVAILABLE in WebSphere MQ Version 7.0, or one of the following more specific reason codes:

- MQRC_CHANNEL_NOT_AVAILABLE
- MQRC_HOST_NOT_AVAILABLE
- MQRC_CHANNEL_CONFIG_ERROR
- MQRC_UNKNOWN_CHANNEL_NAME

If you have a WebSphere MQ Version 6.0 application that takes specific actions based on the reason code, you might need to change the application to take account of the above additional reason codes. All of the reason codes are listed in the MQCONN parameters in the *WebSphere MQ Application Programming Reference*.

Chapter 5. Platform-specific information

This collection of topics describes migration information specific to each platform.

Migrating to WebSphere MQ for AIX Version 7.0

Instructions for migrating to WebSphere MQ Version 7.0.

About this task

If you want to migrate to WebSphere MQ Version 7.0, complete this task.

If you migrate from a previous level of this product without first backing up your system, you **cannot** revert to your previous level, so back up your system **before** you install WebSphere MQ Version 7.0. This will enable you to back out the upgrade if necessary. If you back out the upgrade, however, you cannot recover any work, such as changes to messages and objects, performed by WebSphere MQ Version 7.0.

1. Stop all MQ applications on the machine and the applications accessing remote machines.
2. End all WebSphere MQ activity.
 - a. Log in as a user in group mqm.
 - b. Use the **dspmq** command to display the state of all the queue managers on the system.
 - c. Use the **endmqm** command to stop all running queue managers.
 - d. Stop any listeners associated with the queue managers, using the command:
`endmqlsr -m QMgrName`
 - e. To check that you have stopped all of them, enter the following:
`ps -ef | grep mq`

Check that there are no processes listed that are running command lines beginning amq or runmq. Ignore any that start with amqi.

3. Log in as root.
4. Function supplied by the SupportPacs in the table below has been superseded by function in WebSphere MQ Version 7.0. Remove these SupportPacs before installing WebSphere MQ Version 7.0.

Table 4. SupportPacs superseded by WebSphere MQ Version 7.0

SupportPac Number	Description
MA0Y	WebSphere MQ Bridge for HTTP

Note: If you installed SupportPac MACS, remove the directory `/usr/mqm/inc64` and its contents.
Review any other installed SupportPacs for their applicability to WebSphere MQ Version 7.0.

5. You do not have to uninstall your current version before installing WebSphere MQ Version 7.0. However, before installing you must ensure that you manually remove the filesets which have been retired from WebSphere MQ Version 7.0.

- Remove the following filesets before migrating from WebSphere MQ Version 5.3 to WebSphere MQ Version 7.0:

```
mqm.Client.Bnd
mqm.Server.Bnd
mqm.dce.samples
mqm.dce.server
mqm.msg.De_DE
mqm.msg.Es_ES
mqm.msg.Fr_FR
mqm.msg.It_IT
```

- Remove the following filesets before migrating from WebSphere MQ Version 6.0 to WebSphere MQ Version 7.0:

```
mqm.msg.De_DE
mqm.msg.Es_ES
mqm.msg.Fr_FR
mqm.msg.It_IT
```

6. Install WebSphere MQ Version 7.0 by following the tasks set out in *WebSphere MQ Quick Beginnings for AIX*.
7. Restart WebSphere MQ.

Migrating from an earlier version of WebSphere MQ for AIX client

If you want to migrate to a WebSphere MQ for AIX, Version 7.0 client from an earlier version of the WebSphere MQ client, you must perform certain tasks.

You must first end all WebSphere MQ activity on the target machine.

This migration procedure applies only to migration from an earlier version of a WebSphere MQ for AIX client to IBM WebSphere MQ for AIX Version 7.0 clients. If you are migrating from an earlier version of WebSphere MQ or MQSeries for AIX, you are advised to uninstall your current version before installing the IBM WebSphere MQ for AIX Version 7.0 client.

Migration from an earlier version of WebSphere MQ for AIX involves updating any currently installed filesets, and installing any new filesets that might be required.

To update currently installed filesets:

1. Go into SMIT for root authority. From the shell, enter:
smit
2. Select the device appropriate for your installation using the following sequence of windows:

```
Software Installation and Maintenance
Install and Update Software
Update Installed Software to Latest Level (Update All)
```

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

```
smitty update_latest
```

3. Select the **List** button to display the Single Select List window.
4. Select **/dev/cd0 (CD Drive)**.

5. Select **OK** to display the parameters for **Update All**.
6. Update all previously installed software for WebSphere MQ by selecting the **_update_all** option in the **Software to update** field.
7. Press Enter.
8. Select **OK** in the confirmation window to start updating the software.

When all previously installed filesets have been updated to the latest level, you can install any additional filesets. See the installation instructions in *WebSphere MQ for AIX Quick Beginnings* for more information.

Changes to the installation path

The current version of the WebSphere MQ for AIX client installs into a different directory to versions earlier than Version 5.3. If you are migrating from an early version, you need to take action.

Changes in AIX LPP Version 4 packaging mean that the IBM WebSphere MQ for AIX, Version 5.3 and later clients install into directory **/usr/mqm**. Versions previous to WebSphere MQ for AIX, V5.3 of the product installed into directory **/usr/lpp/mqm**.

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

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

or

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

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

If you uninstall your existing client and either delete or move your existing file system, you can then install the IBM WebSphere MQ for AIX client as described in the installation instructions.

However, if you move the old file system to the new installation path, you should then perform the migration installation described in “Migrating from an earlier version of WebSphere MQ for AIX client” on page 30.

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

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

Removal of obsolete code pages from WebSphere MQ for AIX

IBM-850 code pages are no longer supported.

WebSphere MQ Version 7.0 requires a base AIX operating system at level 5.3 or higher and, from this level of operating system, the locales supporting IBM-850 code pages are no longer supported. Therefore, the following filesets containing message catalogs for IBM-850 code pages are not included in WebSphere MQ Version 7.0 for AIX:

mqm.msg.De_DE	(German)
mqm.msg.Es_ES	(Spanish)
mqm.msg.Fr_FR	(French)
mqm.msg.It_IT	(Italian)

If you have any of WebSphere MQ filesets installed that contain message catalogs encoded in the IBM-850 code page, you must delete them manually before you install WebSphere MQ Version 7.0. You can use the **lspp** command to display them.

The **ifreqs** command is set up in the packages to ensure that you do not install the product with a mixture of WebSphere MQ Version 7.0, and earlier, message catalogs.

To find out which language is currently in use, use the **locale** command. If you are using any of the withdrawn locales, you need to change the setting of the LANG environment variable, or LC_MESSAGES environment variable, or both, to see messages in your desired language; see WebSphere MQ for AIX Quick Beginnings for information about the LANG environment variable.

Defining a transaction program

WebSphere MQ allows you to use the Communications Server for AIX V5 graphical interface to configure transaction programs.

If you are migrating from a previous version of MQSeries, you should delete any existing Communications Server definitions of transaction programs that can be invoked by WebSphere MQ using the following commands:

1. Type

```
snaadmin delete_tp_load_info.tp_name=xxxxx
```

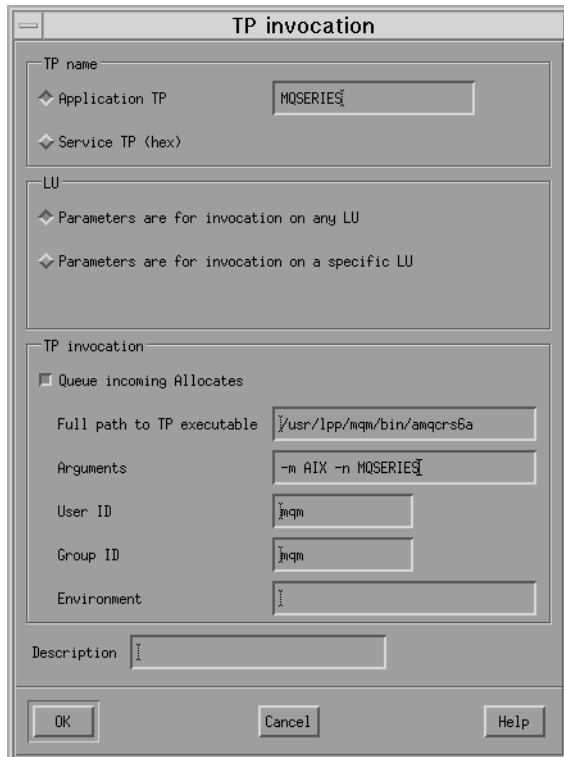
2. Then type

```
snaadmin delete_tp.tp_name=xxxxx
```

An attempt to invoke a previously defined transaction program results in a SNA sense code of 084B6031. In addition, error message AMQ9213 is returned. See WebSphere MQ Messages for more information about this and other WebSphere MQ messages.

You can then re-create the transaction program definition using the following instructions

From the main window, click **Services, APPC, and Transaction programs ...** The following panel is displayed:



1. Type **TP name** (6) in the **Application TP** field.
2. Clear the **Queue incoming Allocates** check box.
3. Type the **Full path to executable** (7).
4. Type **-m Local queue manager** in the **Arguments** field.
5. Type **mqm** in the **User ID** and **Group ID** fields.
6. Enter environment variables **APPCLU=local LU** (4) and **APPCTPN=Invokable TP** (6) separated by the pipe character in the **Environment** field.
7. Click **OK**.

After you have migrated to WebSphere MQ Version 7.0

Perform these tasks immediately after you migrate to WebSphere MQ Version 7.0.

About this task

Immediately after you migrate to WebSphere MQ Version 7.0, you must start your queue managers at least once to migrate your file system structure before you start any WebSphere MQ listeners. Otherwise, you will not be able to start WebSphere MQ listeners after migration.

If you cannot connect to a migrated queue manager using MQ Explorer and receive the error message "SYSTEM.MQEXPLORER.REPLY.MODEL not defined", run the following command on that queue manager:

```
strmqm -c
```

This command refreshes existing system objects with default values (for example, setting the MCAUSER attribute of a channel definition to blanks), and creates the queue required by the WebSphere MQ Explorer.

Extended Shared Memory (EXTSHM) environment variable

You might need to use the extended shared memory environment variable to increase the number of shared storage segments.

On AIX, 32-bit applications can attach only 10 shared storage segments concurrently. When an application connects to WebSphere MQ, typically two storage segments are attached. However, the number of shared storage segments needed might increase depending on the workload of the queue manager. If, when connecting to WebSphere MQ, your 32-bit application fails to connect because it cannot attach all of the shared storage segments, consider setting the environment variable EXTSHM=ON in the environment of the queue manager. Set EXTSHM=ON before running the `strmqm` command and in the environment of your 32-bit applications. Alternatively you can connect to the queue manager using isolated bindings, which use only a single shared storage segment when connecting to the queue manager. Do this by replacing MQCONN calls with MQCONNX calls and specifying MQCNO_ISOLATED_BINDING in the MQCNO parameter of the MQCONNX calls.

There is virtually no limit to the number of shared storage segments attached by 64-bit applications.

Migrating to WebSphere MQ for HP-UX Version 7.0

Follow this information to migrate from previous versions of WebSphere MQ for HP-UX to WebSphere MQ for HP-UX Version 7.0.

About this task

If you want to migrate to WebSphere MQ Version 7.0, complete this task.

If you migrate from a previous level of this product without first backing up your system, you **cannot** revert to your previous level, so back up your system **before** you install WebSphere MQ Version 7.0. This will enable you to back out the upgrade if necessary. If you back out the upgrade, however, you cannot recover any work performed by WebSphere MQ Version 7.0.

1. Stop all Websphere MQ applications on the machine and any applications accessing remote machines. This includes any vendor products used to monitor or control WebSphere MQ.
2. End all WebSphere MQ activity.
 - a. Log in as a WebSphere MQ administrator.
 - b. Use the **endmqm** command to stop all running queue managers.
 - c. Stop any listeners associated with the queue managers, using the command:
`endmq1sr -m QMgrName`
 - d. To check that you have stopped all of them, enter the following:
`ps -ef | grep mq`

Check that there are no processes listed that are running command lines beginning `amq` or `runmq`. Ignore any that start with `amqi`.

3. Login as root.
4. Function supplied by the SupportPacs in the table below has been superseded by function in WebSphere MQ Version 7.0. Remove these SupportPacs before installing WebSphere MQ Version 7.0.

Table 5. SupportPacs superseded by WebSphere MQ Version 7.0

SupportPac Number	Description
MA0Y	WebSphere MQ Bridge for HTTP
MA6P	WebSphere MQ - XA libraries for use with TX-series on HP-UX for IPF

Note: If you installed SupportPac MACS, remove the directory `/usr/mqm/inc64` and its contents.

Review any other installed SupportPacs for their applicability to WebSphere MQ for HP-UX Version 7.0.

5. Uninstall any WebSphere MQ service updates and then uninstall WebSphere MQ.
6. Install WebSphere MQ Version 7.0 by following the tasks set out in *Quick Beginnings for HP-UX*.

After you have migrated to WebSphere MQ Version 7.0

Perform these tasks immediately after you migrate to WebSphere MQ Version 7.0.

About this task

Immediately after you migrate to WebSphere MQ Version 7.0, you must start your queue managers at least once to migrate your file system structure before you start any WebSphere MQ listeners. Otherwise, you will not be able to start WebSphere MQ listeners after migration.

If you cannot connect to a migrated queue manager using MQ Explorer and receive the error message “SYSTEM.MQEXPLORER.REPLY.MODEL not defined”, run the following command on that queue manager:

```
strmqm -c
```

This command refreshes existing system objects with default values (for example, setting the MCAUSER attribute of a channel definition to blanks), and creates the queue required by the WebSphere MQ Explorer.

Migrating WebSphere MQ for i5/OS

This section gives information on considerations for migrating WebSphere MQ for i5/OS.

Migrating from WebSphere MQ V5.3 or later

There are two major types of upgrade:

- The upgrade takes place on the same machine, optionally accompanied by a hardware upgrade. This is referred to as a *slip install*.
- The upgrade takes place on a different machine. This is referred to as a *side-by-side install*.

Whichever type of install you perform, you must first:

1. Prepare your system
2. Quiesce your queue managers

Preparing for migration

About this task

Before performing a slip install or side-by-side install, carry out the following procedure:

1. Sign on to the system with a user profile that has *ALLOBJ special authority, for example QSECOFR.
2. Stop all applications that are using the existing version of WebSphere MQ. (On WebSphere MQ version 5.3 and later, use the command WRKMQM, option 22 "Work with queue manager jobs", to help find them).
3. End all channels for all queue managers on the system. To do this, use the WRKMQMCHL command and select option 15.
4. On each queue manager, end the command server. To do this, enter the command:

```
ENDMQMCSVR MQMNAME(QMGRNAME) OPTION(*IMMED)
```

where *QMGRNAME* is the name of the queue manager.

5. Remove any unwanted messages from your queues.
6. Resolve any in-doubt messages that are held by sender or server channels. To do this, use the WRKMQMCHST command and select option 17.
7. On each queue manager, save the latest media recovery checkpoint. To do this enter the following command:

```
RCDMQMIMG OBJ(*ALL) OBJTYPE(*ALL) MQMNAME(QMGRNAME) DSPJRNDTA(*YES)
```

8. Quiesce the product. To do this, see "Quiescing WebSphere MQ" on page 37.
9. End the QMQM subsystem by entering the command:

```
ENDSBS SBS(QMQM)
```

10. Save your WebSphere MQ data:

- a. Create a save file for every queue manager library on your system. To do this, issue the command:

```
CRTSAVF FILE(QGPL/queue-manager-library)
```

where the *queue-manager-library* name consists of the name of the queue manager preceded by QM.

- b. Save your queue manager libraries into the save files. To do this, issue the commands:

```
SAVLIB LIB(queue-manager-library) DEV(*SAVF)  
SAVF(QGPL/queue-manager-library)
```

- c.

Remove all unwanted FDC data from directory:

```
QIBM/UserData/mqm/errors
```

Remove old FDC files with the command:

```
RMVLNK OBJLNK('/QIBM/UserData/mqm/errors/*.FDC')
```

This cleans up all files with an extension of 'FDC' in the IFS.

Remove old JOB files with the command:

```
RMVLNK OBJLNK('/QIBM/UserData/mqm/errors/*.JOB')
```

This cleans up all files with an extension of 'JOB' in the IFS.

- d.

Remove all unwanted trace data from directory:

QIBM/UserData/mqm/trace

(or remove the whole directory).

Remove all trace files with the command:

```
RMVLNK OBJLNK('/qibm/userdata/mqm/trace/*')
```

e. Create a save file for MQ IFS data. To do this, issue the command:

```
CRTSAVF FILE(QGPL/QMUSERDATA)
```

f. Save your MQ IFS data, using the command:

```
SAV DEV('/QSYS.LIB/QGPL.LIB/QMUSERDATA.FILE') OBJ('/QIBM/UserData/mqm')
```

11. If you are moving from running WebSphere MQ on any machine to WebSphere MQ on a new machine, transfer the save files to the new machine.

Quiescing WebSphere MQ

About this task

The orderly shutdown of WebSphere MQ is called *quiescing*. You need to quiesce WebSphere MQ to upgrade to a newer version.

To quiesce one or more queue managers:

1. Sign on to a new interactive i5/OS session, ensuring that you are not accessing any WebSphere MQ objects.
2. Ensure that you have:
 - a. *ALLOBJ authority, or object management authority for the QMQM library.
 - b. Sufficient authority to use the ENDSBS command.
3. Warn all users that you are going to stop WebSphere MQ.
4. Quiesce all queue managers:-
 - a. On Version 5.3, use the ENDMQM command:

```
ENDMQM MQMNAME(*ALL) OPTION(*CNTRLD) ENDCCTJOB(*YES) TIMEOUT(15)
```

Where *15* is a timeout value in seconds.

If the ENDMQM command has not completed within a reasonable period (at least 10 minutes), use the WRKMQM command to identify the queue managers that are still ending, and force each one in turn to stop by issuing:

```
ENDMQM MQMNAME(QMGRNAME) OPTION(*IMMED)
```

Where *QMGRNAME* is the name of the queue manager.

Complete the tidying up of shared memory by issuing the command:

```
ENDMQM MQMNAME(*ALL) OPTION(*IMMED) ENDCCTJOB(*YES) TIMEOUT(15)
```

b. On Version 6.0 or later, use the ENDMQM command:

```
ENDMQM MQMNAME(*ALL) OPTION(*CNTRLD) ENDCCTJOB(*YES) RCDMQMIMG(*YES) TIMEOUT(15)
```

Where *15* is a timeout value in seconds.

If the ENDMQM command has not completed within a reasonable period (at least 10 minutes), use the WRKMQM command to identify the queue managers that are still ending, and force each one in turn to stop by issuing:

```
ENDMQM MQMNAME(QMGRNAME) OPTION(*IMMED)
```

Where *QMGRNAME* is the name of the queue manager.

Complete the tidying up of shared memory by issuing the command:

```
ENDMQM MQMNAME(*ALL) OPTION(*IMMED) ENDCCTJOB(*YES) RCDMQMIMG(*NO)
TIMEOUT(15)
```

5. If the command in the previous step does not complete, end the subsystem immediately by issuing:
ENDSBS SBS(QMQM) OPTION(*IMMED)
6. If this also fails, use the operating system command ENDJOB to end all jobs in the subsystem QMQM, as described below.

Note: Do not use ENDJOBABN unless you intend to IPL the machine before starting WebSphere MQ. Ending WebSphere MQ jobs using ENDJOBABN can lead to damaged semaphores, which in turn can prevent your queue manager from starting.

- a. If a QMGR(s) must be shut down manually, the recommended order of ending jobs (ENDJOB) is shown below (if the job exists) - wait a few minutes for AMQA* or AMQZ* jobs to tidy up.

- 1) RUNMQLSR - TCP listener (multi-threaded)
- 2) AMQCLMAA - TCP listener (single-threaded)
- 3) AMQRMPPA - Channel process pooling job
- 4) RUNMQCHI - channel initiator
- 5) AMQCRSTA - receiving MCA jobs
- 6) RUNMQCHL - sending MCA jobs
- 7) AMQCRS6B - LU62 receiver channel
- 8) AMQPCSEA - command server
- 9) RUNMQTRM - Application trigger monitor
- 10) RUNMQDLQ - Dead letter queue handler
- 11) AMQFCXBA - Broker Worker Job
- 12) AMQFQPUB - Queued Publish/Subscribe Daemon
- 13) AMQZMUC0 ('0' is a zero) - Utility Manager
- 14) AMQZMUF0 ('0' is a zero) - Utility Manager
- 15) AMQZMUR0 ('0' is a zero) - Utility Manager
- 16) AMQZMGR0 ('0' is a zero) - Process Controller
- 17) AMQRRMFA - cluster repository manager
- 18) AMQZDMAA - deferred message manager
- 19) AMQALMPX - Log Manager
- 20) AMQZFUMA - Object Authority Manager
- 21) AMQZLAS0 ('0' is a zero) - LQM agents
- 22) AMQZLAA0 ('0' is a zero) - LQM agents
- 23) AMQZXMA0 ('0' is a zero) - Execution Controller

- b. Issue the following command:

```
ENDMQM MQMNAME(QMGRNAME) OPTION(*IMMED)
```

- c. Issue the following command:

```
ENDMQM MQMNAME(*ALL) OPTION(*CNTRLD) ENDCCTJOB(*YES) RCDMQMIMG(*NO)
TIMEOUT(05)
```

Where 05 is a timeout value in seconds.

- d. Manually clean up shared memory. Issue the following command:

```
EDTF '/QIBM/UserData/mqm/qmgrs'
```

then:

- 1) Take option 5 for &SYSTEM and check that the following sub-dirs are empty: isem, esem, msem, ssem, and shmem.
- 2) Take option 5 for QMGRNAME and check that the following sub-dirs are empty :- isem, esem, msem, ssem, and shmem.

- 3) Take option 5 for `&ipcc` in the QMGRNAME directory and check that the following sub-dirs are empty :- isem, esem, msem, ssem, and shmem.
- 4) Take option 5 for `&qmpersist` in the QMGRNAME directory and check that the following sub-dirs are empty :- isem, esem, msem, ssem, and shmem.
- 5) Take option 5 for `&app` and check that the following sub-dirs are empty: isem, esem, msem, ssem, and shmem.

Performing a slip install

This lists the tasks that you perform to perform a slip install, when you upgrade and migrate WebSphere MQ for i5/OS on a computer where an earlier version is installed.

About this task

To perform a slip install, carry out the following tasks:

1. "Preparing for migration" on page 36
2. iq10390_.dita
3. iq10410_.dita
4. "Verifying the migration" on page 41
5. iq10420_.dita

Performing a side-by-side install

This lists the tasks that you perform a side-by-side install, when you upgrade and migrate WebSphere MQ for i5/OS on a different computer.

About this task

To perform a side-by-side install, carry out the following tasks:

1. "Preparing for migration" on page 36
2. iq10390_.dita
3. iq10410_.dita
4. "Verifying the migration" on page 41
5. iq10420_.dita
6. Restore the queue manager libraries for every queue manager, using the command:

```
RSTLIB SAVLIB(queue-manager-library) DEV(*SAVF) (*PRINT)
SAVF(QGPL/queue-manager-library)
```

where the *queue-manager-library* name consists of the name of the queue manager preceded by QM.

7. Restore the WebSphere MQ IFS data, using the command:

```
RST DEV('/QSYS.LIB/QGPL.LIB/QMUSERDATA.FILE') OBJ('/QIBM/UserData/mqm') (*PRINT)
```
8. To associate the journal receivers, issue the command WRKJRN on the journal AMQAJRN in each queue manager library, by pressing *PF4* and selecting option 9.
9. If you want to set up your work management environment, job descriptions, and pools, see the WebSphere MQ System Administration Guide for guidance. Otherwise, use the default setup.

Migrating translated versions

Follow these instructions to migrate a translated version.

About this task

Previous editions of this book recommended installing translated versions that are not on the CD by: installing a language that is on the CD, and then copying the QSYS29xx files into QSYS. This is no longer recommended because those objects will not be removed at a subsequent upgrade or reinstall. The new recommended procedure is described in Installing translated versions.

If you are upgrading now, and, when you installed your existing translated version, you followed the previous advice and copied objects into QSYS, then follow the steps below to ensure that all the old objects are removed at an upgrade or reinstall of MQ. This is a one-time resolution.

To remove old objects from QSYS:

1. To uninstall the previous version of MQ, issue the command:

```
DLTLICPGM 5724B41
```

where 5724B41 is the component id of the previous version of MQ.

2. Create a temporary user profile:

```
CRTUSRPRF QMQMx
```

where QMQMx is any new temporary usrprf.

3. Delete the previous QMQM profile, and change the ownership of its objects to the temporary profile:

```
DLTUSRPRF USRPRF(QMQM) OWNNOBJOPT(*CHGOWN QMQMx)
```

4. Install the new version of WebSphere MQ for i5/OS, V7.0 base product and language:

```
RSTLICPGM LICPGM(5724H72) DEV(install device) RSTOBJ(*LNG) LNG(language ID)
```

See Installing translated versions for full details on this command, and what to do if your required language is not on the CD.

5. To preserve the required objects, change the ownership of the IFS /QIBM/UserData/mqm and its subdirs back to QMQM using the command:

```
CHGOWN OBJ('/QIBM/UserData/mqm') NEWOWN(QMQM)
```

6. To preserve the required objects in the queue manager libraries, change their ownership back to QMQM using commands such as:

```
CHGOBJOWN OBJ(qmgr_lib/qmgr_object) OBJTYPE(*object_type) NEWOWN(QMQM)
```

where:

- *qmgr_lib* is the queue manager library,
- *qmgr_object* is one of the objects there,
- *object_type* is the type of the object.

Repeat the command for every object, and every queue manager.

7. Delete the temporary user profile QMQMx (and all its owned objects):

```
DLTUSRPRF USRPRF(QMQMx) OWNNOBJ(*DLT)
```

8. Follow the steps in Verifying the installation

Migrating WebSphere MQ classes for JMS and WebSphere MQ classes for Java on i5/OS

If you have WebSphere MQ Java SupportPac MA88 installed, you must uninstall it first.

About this task

If you have WebSphere MQ Java V5.3, or WebSphere MQ Java V6.0 installed, and want to install the WebSphere MQ Java V7.0 product on i5/OS, you can install the new version without uninstalling the old one.

If you do have MA88 installed, and try to install WebSphere MQ Java V7.0 anyway, the installation fails with a warning requesting you to uninstall the old client.

To uninstall the previous Java client:

1. Delete the QMQMJAVA library and the /QIBM/ProdData/mqm/java directory, by issuing the command:

```
DLTLICPGM LICPGM(5648C60) OPTION(*ALL)
```
2. If the step above failed to delete the IFS directory /QIBM/ProdData/mqm/java and its subdirectories, use the EDTF command, for example:

```
EDTF STMF('/QIBM/ProdData/mqm')
```

and select option 9 against the java directory.

Verifying the migration

After you have verified the installation, perform these steps.

Before you begin

Before you start, make sure you have performed the steps in *Verifying the installation* in *WebSphere MQ Intercommunications*.

About this task

To verify that you have migrated to WebSphere MQ for i5/OS, V7.0 successfully:

1. Make QMQMADM either the primary or a secondary group profile for your user profile. To do this, issue one of the following commands:

```
CHGUSRPRF USRPRF(YOUR PROFILE) GRPPRF(QMQMADM)  
CHGUSRPRF USRPRF(YOUR PROFILE) SUPGRPPRF(QMQMADM)
```
2. Start the WebSphere MQ subsystem with the command:

```
STRSBS SBS(D(QMQM/QMQM))
```

(If it is already running, you will get error message CPF1010 which you can safely ignore).
3. Check that your queue managers are accessible by issuing the command:

```
WRKMQM
```

Use option 14 against each queue manager to start it.
Use option 5 against each queue manager to check its attributes.
4. You can use the other options to check your queue manager objects, for example check your queues using option 18, check your channels using option 20, and so on.

5. Take a fresh media recovery checkpoint, using the following command:
- ```
RCDMQMIMG OBJ(*ALL) OBJTYPE(*ALL) MQMNAME(QMGRNAME) DSPJRNDTA(*YES)
```

Where *QMGRNAME* is the name of the queue manager.

## Results

Refer to the documentation for more information about these commands and how to use them. See either *Command Reference* or *Documentation in Delivered components* in *WebSphere MQ Intercommunications*.

## After migration

This lists the tasks that you perform after upgrading and migrating WebSphere MQ for i5/OS.

1. When you are satisfied that the migration has completed successfully, you can delete the saved data in the save files in QGPL. (This data was saved in Preparing for migration.)
2. See the tasks in After installation.

## Results

---

## Upgrading to WebSphere MQ for Linux Version 7.0

This section provides information about considerations for upgrading WebSphere MQ for Linux.

If you are upgrading from a previous version of WebSphere MQ for Linux, you must uninstall your current version before installing WebSphere MQ Version 7.0.

Because you cannot therefore revert to your previous level, back up your system before you install WebSphere MQ Version 7.0. This enables you to back out the upgrade if necessary. If you back out the upgrade, however, you cannot recover any work, such as changes to messages and objects, that WebSphere MQ Version 7.0 has done.

### WebSphere MQ for Linux Version 7.0 - zSeries® platform 32-bit and 64-bit

You cannot upgrade from WebSphere MQ for Linux V6.0 (zSeries platform) (32-bit) to WebSphere MQ for Linux zSeries s390x V7.0 (64-bit).

### WebSphere MQ for Linux, (x86 and x86-64 platforms) and 32-bit and 64-bit queue managers

You can install WebSphere MQ for Linux (x86 platform) on 32-bit and 64-bit versions of the Linux operating system. If you currently have 32-bit WebSphere MQ for Linux (x86 platform) Version 5.3 or Version 6.0 installed on a 64-bit version of the Linux operating system, you can install WebSphere MQ for Linux (x86 platform) or WebSphere MQ for Linux (x86-64) when you upgrade to WebSphere MQ for Linux, Version 7.0.

If you install the 64-bit version of WebSphere MQ for Linux (x86-64 platform), existing queue managers are migrated to 64-bit when they are started. When a queue manager is 64-bit, you **cannot** start it by using a 32-bit version of WebSphere MQ.

Upgrading from WebSphere MQ for Linux (x86 platform), Version 7.0 to WebSphere MQ for Linux (x86-64 platform) Version 7.0 is not supported.

Upgrading from WebSphere MQ for Linux (x86-64 platform) to WebSphere MQ for Linux (x86 platform) is not supported for any version.

The supported upgrade paths for previous versions of WebSphere MQ for Linux (x86 and x86-64 platforms) to WebSphere MQ for Linux Version 7.0 are as follows:

| From:                                                 | To:                                                                                                         |
|-------------------------------------------------------|-------------------------------------------------------------------------------------------------------------|
| WebSphere MQ for Linux, Version 5.3 (x86 platform)    | WebSphere MQ for Linux, Version 7.0 (x86 platform)<br>WebSphere MQ for Linux, Version 7.0 (x86-64 platform) |
| WebSphere MQ for Linux, Version 6.0 (x86 platform)    | WebSphere MQ for Linux, Version 7.0 (x86 platform)<br>WebSphere MQ for Linux, Version 7.0 (x86-64 platform) |
| WebSphere MQ for Linux, Version 6.0 (x86-64 platform) | WebSphere MQ for Linux, Version 7.0 (x86-64 platform)                                                       |

See the following topics for information about how to migrate to WebSphere MQ for Linux Version 7.0.

## Upgrading to WebSphere MQ Version 7.0

Complete this task to migrate to WebSphere MQ Version 7.0, while retaining your WebSphere MQ data, and optionally upgrade your operating system to one supported by WebSphere MQ Version 7.0.

### Before you begin

Review the upgrading limitations that are described in “Upgrading to WebSphere MQ for Linux Version 7.0” on page 42.

Review the WebSphere MQ Version 7.0 prerequisites that are described in *WebSphere MQ Quick Beginnings for Linux*. If your system version does not have the prerequisites, you must install a version that does. Operating system installation is an optional step in the procedure later in this topic.

1. Stop all WebSphere MQ applications on the computer.
2. End all WebSphere MQ activity:
  - a. Log in as a user in group `mqm`.
  - b. Display the state of all the queue managers on the system, by using the command `dspmq`.
  - c. Stop all running queue managers, by using the command `endmqm`.
  - d. Stop any listeners associated with the queue managers, by using the command:

```
endmq1sr -m QMgrName
```
  - e. To check that you have stopped all necessary WebSphere MQ activity, type:

```
ps -ef | grep mq
```

If any processes are listed that are running command lines beginning `amq` or `runmq`, stop them. Ignore any that start with `amqi`.

3. Log in as root.
4. Remove the SupportPacs listed in the table below. They have been superseded by function in WebSphere MQ Version 7.0.

| SupportPac Number | Description                  |
|-------------------|------------------------------|
| MA0Y              | WebSphere MQ Bridge for HTTP |

5. Uninstall all WebSphere MQ service updates, then uninstall WebSphere MQ.
6. Optional: If you need to upgrade your operating system, complete the following steps:
  - a. Copy the `/var/mqm` directory and save it to a location that will not be erased by the operating system upgrade.
  - b. Upgrade your operating system by following the manufacturer's instructions.
  - c. Copy the `/var/mqm` directory that you saved in step 6.a back into your file system.
7. Install WebSphere MQ Version 7.0 as described in
8. Restart WebSphere MQ. Queue managers are migrated to WebSphere MQ Version 7.0 when you start them.

## Upgrading to WebSphere MQ Version 7.0 using RPM upgrade tools

The use of RPM upgrade tools, using the `-U` or `-F` flags, to upgrade is not supported.

### About this task

If you have already tried to upgrade WebSphere MQ by using `rpm -U` or `rpm -F`, you might have deleted your old WebSphere MQ package entries from the RPM database without removing the product from your system. You might also have partially installed WebSphere MQ Version 7.0.

To continue upgrading to WebSphere MQ Version 7.0, follow these steps:

1. Find out which WebSphere MQ packages still have entries in your RPM database by using `rpm -qa | grep MQSeries`
2. Remove all remaining WebSphere MQ packages from your system by using `rpm -e <package-name>`
3. Remove the `/opt/mqm` directory by using `rm -rf /opt/mqm`
4. Install WebSphere MQ Version 7.0 using the instructions given in "Installing a WebSphere MQ server" in *WebSphere MQ Quick Beginnings for Linux*, or "Installing a WebSphere MQ client" in *WebSphere MQ Quick Beginnings for Linux*.

## After you have upgraded to WebSphere MQ Version 7

You must perform certain tasks immediately after upgrading to V7.

## About this task

Immediately after you upgrade to Websphere MQ Version 7.0, you must start your queue managers at least once to migrate your file system structure before you start any Websphere MQ listeners. Otherwise, you will not be able to start Websphere MQ listeners after upgrading.

If you cannot connect to a migrated queue manager by using MQ Explorer, and receive the error message `SYSTEM.MQEXPLORER.REPLY.MODEL not defined`, run the following command on that queue manager:

```
strmqm -c
```

This command refreshes existing system objects with default values (for example, setting the `MCAUSER` attribute of a channel definition to blanks), and creates the queue required by WebSphere MQ Explorer. The queue manager is stopped when the command finishes.

---

## Migrating to WebSphere MQ for Solaris Version 7.0

Follow this information to migrate from previous versions of WebSphere MQ for Solaris to WebSphere MQ for Solaris Version 7.0.

### About this task

If you want to migrate to WebSphere MQ Version 7.0 complete this task.

If you migrate from a previous level of this product without first backing up your system, you **cannot** revert to your previous level, so back up your system **before** you install WebSphere MQ Version 7.0. This will enable you to back out the upgrade if necessary. If you back out the upgrade, however, you cannot recover any work performed by WebSphere MQ Version 7.0.

1. Stop all MQ applications on the machine and the applications accessing remote machines.
2. End all WebSphere MQ activity.
  - a. Log in as MQ administrator.
  - b. Use the **endmqm** command to stop all running queue managers.
  - c. Stop any listeners associated with the queue managers, using the command:

```
endmq1sr -m QMgrName
```
  - d. To check that you have stopped all of them, enter the following:

```
ps -ef | grep mq
```

Check that there are no processes listed that are running command lines beginning `amq` or `runmq`. Ignore any that start with `amqi`.

3. Log in as root.
4. Function supplied by the SupportPacs in the table below has been superseded by function in WebSphere MQ Version 7.0. Remove these SupportPacs before installing WebSphere MQ Version 7.0.

Table 6. SupportPacs superseded by WebSphere MQ Version 7.0

| SupportPac Number | Description                     |
|-------------------|---------------------------------|
| MA0C              | MQSeries Publish/subscribe      |
| MA0R              | WebSphere MQ transport for SOAP |

Table 6. SupportPacs superseded by WebSphere MQ Version 7.0 (continued)

| SupportPac Number | Description                                                                 |
|-------------------|-----------------------------------------------------------------------------|
| MA88              | MQSeries Classes for Java and WebSphere MQ classes for Java Message Service |
| MACR              | MQSeries Client libraries for Sun Solaris (64-bit)                          |

Review any other installed SupportPacs for their applicability to WebSphere MQ Version 7.0.

5. Uninstall any WebSphere MQ service updates and then uninstall WebSphere MQ.
6. Install WebSphere MQ Version 7.0 by following the tasks set out in *Quick Beginnings for Solaris* and *Quick Beginnings for Solaris*.

## After you have migrated to WebSphere MQ Version 7.0

Perform these tasks immediately after you migrate to WebSphere MQ Version 7.0.

### About this task

Immediately after you migrate to WebSphere MQ Version 7.0, you must start your queue managers at least once to migrate your file system structure before you start any WebSphere MQ listeners. Otherwise, you will not be able to start WebSphere MQ listeners after migration.

If you cannot connect to a migrated queue manager using MQ Explorer and receive the error message “SYSTEM.MQEXPLORER.REPLY.MODEL not defined”, run the following command on that queue manager:

```
strmqm -c
```

This command refreshes existing system objects with default values (for example, setting the MCAUSER attribute of a channel definition to blanks), and creates the queue required by the WebSphere MQ Explorer.

---

## Migrating to WebSphere MQ for Windows Version 7.0

Follow this information to migrate from previous versions of WebSphere MQ for Windows to WebSphere MQ for Windows Version 7.0.

Before you upgrade to WebSphere MQ Version 7.0, you must consider this information.

If you migrate from a previous level of this product without first backing up your system, you **cannot** revert to your previous level, so back up your system **before** you install WebSphere MQ Version 7.0. This enables you to back out the upgrade if necessary. If you back out the upgrade, however, you cannot recover any work, such as changes to messages and objects, for example, performed by WebSphere MQ Version 7.0. For distributed platforms, information about how to back up your system begins with Backing up and restoring WebSphere MQ in the WebSphere MQ *System Administration Guide*.

The installation process for WebSphere MQ for Windows detects whether this is a new installation or an update from a previous level of this product. If you migrate from a previous level, all the objects that you previously created (for example, your



queue managers) can be maintained. The components that were previously installed are preselected in the feature options when you install the new level. If you leave these components selected, you can keep them or reinstall them. However, if you deselect any of these components, the installation process uninstalls them.

You can also migrate to WebSphere MQ V7.0 on a system where a previous version has been uninstalled but where the queue manager data has been retained.

If an earlier version of WebSphere MQ is installed on the computer:

- Ensure that no queue managers, listeners, MQ applications or remote applications that use queue managers are running and that the WebSphere MQ Service is stopped. To stop the WebSphere MQ Service, right-click on the WebSphere MQ icon in the system tray, then click Stop WebSphere MQ.
- When the installation of WebSphere MQ Version 7.0 completes, the Prepare WebSphere MQ Wizard launches automatically. Use this wizard to reenter domain, user ID and password information.

You can subsequently use the DCOMCNFG.EXE tool, shipped with Windows systems, to change the domain, user ID, and password information. This tool is described in *WebSphere MQ System Administration Guide*

**For Windows Vista and Windows Server 2008 users with UAC enabled only:** if you do not complete the Prepare WebSphere MQ Wizard directly after WebSphere MQ installs or if for any reason your machine is rebooted between completing WebSphere MQ installation and completing the Prepare WebSphere MQ Wizard, ensure the wizard is run with Administrator privilege, otherwise the wizard might fail.

- WebSphere MQ Version 7.0 supports migration from WebSphere MQ Version 5.3 and WebSphere MQ Version 6.0 only.
- WebSphere MQ Version 7.0 does not coexist with previous versions of WebSphere MQ. You must migrate from either WebSphere MQ Version 5.3 or WebSphere MQ Version 6.0 to WebSphere MQ Version 7.0.
- By default, a typical WebSphere MQ Version 6.0 to Version 7.0 migration installs only the same features that were installed in the previous version installation. For example, if WebSphere MQ Explorer was not installed in the Version 6.0 installation, it is not installed in a Version 7.0 installation. If you want WebSphere MQ Explorer, select a custom installation of WebSphere MQ Version 7.0 and select the MQ Explorer feature on the Features panel. If you are migrating from WebSphere Version 5.3 to Version 7.0, WebSphere MQ Explorer is installed by default. If you do not want WebSphere MQ Explorer, uninstall the WebSphere MQ Explorer feature by selecting a custom installation of WebSphere MQ Version 7.0. Then deselect the MQ Explorer feature on the Features panel. For more information on how to deselect features, see Launchpad instructions in *Quick Beginnings for Windows*.

## Choosing not to install WebSphere MQ Explorer and WebSphere Eclipse Platform

You can choose not to install the WebSphere MQ Explorer feature, during a custom installation.

By default, a typical migration installation enables the WebSphere MQ Explorer feature. WebSphere MQ Explorer has a prerequisite of WebSphere Eclipse Platform V3.3.

If you do not want to install WebSphere Eclipse Platform V3.3, select a custom installation then deselect the WebSphere MQ Explorer feature from the Features panel.

If you installed WebSphere MQ Explorer on WebSphere MQ Version 6.0 but do not want to install it on WebSphere MQ Version 7.0, select a custom installation and then deselect the WebSphere MQ Explorer feature. The migration process removes the Version 6.0 WebSphere MQ Explorer feature.

For more information on how to deselect features, see Launchpad instructions in *Quick Beginnings for Windows*.

## Migrating to WebSphere MQ Version 7.0 on Windows Vista or Windows Server 2008

You can migrate to WebSphere MQ Version 7.0 on Windows Vista or Windows Server 2008 using either of the following supported migration paths:

*Table 7. WebSphere MQ Version 7.0 on Windows Vista or Windows Server 2008: supported migration paths*

| From:                                                                                        | To:                                                                     |
|----------------------------------------------------------------------------------------------|-------------------------------------------------------------------------|
| WebSphere MQ Version 6.0 Client plus FixPack 6.0.2.1 on Windows Vista or Windows Server 2008 | WebSphere MQ Version 7.0 Client on Windows Vista or Windows Server 2008 |
| WebSphere MQ Version 7.0 on Windows XP SP2                                                   | WebSphere MQ Version 7.0 on Windows Vista or Windows Server 2008        |

## User Account Control (UAC) on Windows Vista and Windows Server 2008

Windows Vista and Windows Server 2008 introduce a User Account Control (UAC) feature, which restricts the actions users can perform on certain operating system facilities, even if they are members of the Administrators group.

UAC is enabled by default on Windows Vista and Windows Server 2008. This means installing WebSphere MQ on Windows Vista or Windows Server 2008 differs from installing WebSphere MQ on previous versions of Windows. At certain points during installation, migration, and uninstallation, you must manually accept the Windows UAC prompt to allow processes to run with elevated authority. During silent installation and uninstallation, you must invoke the process from an elevated command prompt. The points when you have to accept the Windows prompt for UAC or invoke processes from an elevated command prompt have been flagged in the specific topics affected.

## Launchpad migration instructions

### About this task

These instructions cover migrating WebSphere MQ to Version 7.0 using the launchpad.

1. Stop all existing WebSphere MQ processes.
  - a. Stop execution of all WebSphere MQ applications on the local machine, and on any connected remote machines.
  - b. Stop execution of all queue managers
  - c. Stop execution of all channels

- d. Stop execution of all listeners
  - e. Right click WebSphere MQ on the system tray and select Stop WebSphere MQ. Wait for all services to stop.
  - f. Right click WebSphere MQ on the system tray and select Exit. Some previous versions of WebSphere MQ might not have an Exit option. If there is not, use the task manager to end process *amqmsrvn.exe*.
2. Insert the WebSphere MQ Version 7.0 Server CD into the CD-ROM drive.
  3. If autorun is enabled, the launchpad starts automatically. If not, run setup.exe from the CD-ROM.  
If you are migrating on Windows Vista or Windows Server 2008 and UAC is enabled, accept the Windows prompt to allow the launchpad to run as elevated. During migration you might also see Open File - Security Warning dialog boxes that list International Business Machines Limited as the publisher. Click **Run** to allow the migration to continue.
  4. Click the **Software Requirements** option to check if prerequisites are met. You might need to install or upgrade to WebSphere Eclipse Platform Version 3.3 if you want to use WebSphere MQ Explorer in WebSphere MQ Version 7.0.
  5. Click the **Network Configuration** option and select from the three choices. Click **More Information** to launch the relevant WebSphere MQ documentation.
  6. Click the **Migrating SSL Certificates** option and select from the choices. This option will only appear if you are migrating from WebSphere MQ Version 5.3, otherwise it will not be shown.
  7. Click the **WebSphere MQ Installation** option, check the pre-installation status, select language, and click Launch IBM WebSphere MQ Installer.
    - a. Read and accept the license agreement to proceed.
    - b. Select **update** or **custom**. Update installs WebSphere MQ Version 7.0 with the same features as the previous version. Custom allows you to add or remove components for the new version.
    - c. If **custom** was selected, choose components. Unselecting a component will remove the component from the machine if it is already installed. WebSphere MQ Explorer requires WebSphere Eclipse Platform 3.3 to have been installed.
    - d. Confirm details and install. Follow any on-screen instructions.
    - e. Restart computer on completion of installation if you are prompted to do so.
    - f. WebSphere MQ will launch some final configuration steps, allowing you to review network configuration options, and select components to start up when finished.
  8. WebSphere MQ Version 7.0 is now installed, and all persistent items have been migrated from the previous version of WebSphere MQ. You can use the launchpad to view the Quick Beginnings, Release Notes, or Product Tour before finishing.

## SPX support on Windows Vista and Windows Server 2008

The Sequenced Package Exchange protocol (SPX) is not supported on Windows Vista or Windows Server 2008. SPX is supported on Windows XP and Windows 2003 only.

## WebSphere MQ service objects

This section gives information about service objects in WebSphere MQ for Windows.

In WebSphere MQ for Windows, services are now managed by service objects. The listener, command server, trigger monitor, and channel initiator services defined using the MMC plugin in earlier versions of WebSphere MQ will be migrated to queue manager objects as follows:

- The startup property of the command server (automatic or manual) is migrated to the SCMDSERV attribute of the queue manager.
- The startup property of the channel initiator (automatic or manual) is migrated to the SCHINIT attribute of the queue manager.
- For each listener defined in the WebSphere MQ Explorer, a queue manager LISTENER object is created.
- For each trigger monitor defined in the WebSphere MQ Explorer, a queue manager SERVICE object is created.

## WebSphere MQ Explorer for Version 7.0 - error connecting to a migrated queue manager

If you cannot connect to a migrated queue manager and see an error like "SYSTEM.MQEXPLORER.REPLY.MODEL not defined", run:

```
strmqm -c
```

on that queue manager.

This command refreshes existing system objects (for example, setting the MCAUSER attribute of a channel definition to blanks), and creates the queue required by WebSphere MQ Explorer.

## Migrating an MSCS configuration

Migrate one node at a time, following these instructions.

### About this task

These steps are required for a rolling upgrade with a minimum amount of downtime. You should always upgrade a Passive node with no Active WebSphere MQ resources. In an Active/Active setup, convert the node being migrated to a Passive node with no running MQ resources under MSCS control.

1. Modify the possible owners of the WebSphere MQ resource to encompass only the Active node or nodes. This prevents the WebSphere MQ resource from experiencing an unexpected failover to the node currently being migrated.
2. Ensure that the group containing the WebSphere MQ resource is currently on one of the nodes defined as a possible owner. This should include any applications connecting to the queue manager resource.
3. Stop the cluster service on the node being migrated. This clears the MSCS cache of any WebSphere MQ DLLs that have been registered.
4. Migrate the Passive node by following the standard instructions in "Migrating to WebSphere MQ for Windows Version 7.0" on page 46. Apply the required maintenance level.
5. Start the cluster service on the Passive node.
6. On the next node to be migrated, ensure that the WebSphere MQ resources are offline.
7. Remove this node from the list of possible owners. For clusters with more than 2 nodes, see the additional considerations later in this topic.

8. Move the group containing the WebSphere MQ resource to one of the possible owners and bring it online.
9. Repeat steps 3-8 as necessary for any remaining nodes.

### What to do next

**Additional considerations in an MSCS setup with more than 2 nodes:** When you modify the list of possible owners for the WebSphere MQ resources in a poly-node setup, ensure that the list of possible owners (at any given time) does not contain both migrated and non-migrated nodes. As each node is migrated, modify the list of possible owners to the remaining nodes that have not yet been migrated. When half of the nodes have been migrated, the list of possible owners should include only migrated nodes. Modify the list of possible owners at this point.

If you are migrating from WebSphere MQ Version 5.3, see “MSCS migration issues” on page 151

---

## Upgrading to WebSphere MQ for z/OS Version 7.0

This chapter describes the things that you must consider if you are migrating a single queue manager from a previous version of WebSphere MQ.

First class migration support is provided to WebSphere MQ for z/OS Version 7.0 from WebSphere MQ for z/OS Version 6.0. This means you can restart a queue manager at WebSphere MQ Version 7.0 by replacing the executable code (for example, the STEPLIB data sets) with equivalent Version 7.0 libraries and then restarting the queue manager. When a backward migration PTF has been applied to Version 6.0 code, it is possible to fall back from Version 7.0 operation to Version 6.0.

Follow these steps to migrate a production queue manager from WebSphere MQ Version for z/OS Version 6.0 to WebSphere MQ Version for z/OS Version 7.0:

1. Install Version 7.0 early code. Refresh the early code level of all queue managers in the LPAR.
2. Apply the Version 7.0 coexistence and backward migration PTFs to the Version 6.0 code.
3. Restart production WebSphere MQ systems with this PTF-updated Version 6.0 code. Verify correct function before proceeding and review any special actions detailed in the PTFs. If you require fall back at this stage use normal maintenance procedures to revert to the Version 6.0 code before PTF application.
4. Restart production WebSphere MQ systems with Version 7.0 code. Verify correct function of existing applications before using any new Version 7.0 functions. If problems become apparent at this stage, fall back to the Version 6.0 code with backward migration PTFs that have been previously validated for your environment.

For a queue-sharing group, apply the Version 6.0 coexistence and backward migration PTFs to all members before you migrate any member to Version 7.0.

For releases earlier than Version 6.0, no backward migration support from Version 7.0 is provided. You can migrate a queue manager from a release earlier than Version 6.0 directly to Version 7.0, but if problems arise there is no good fall back capability. This is not good practice for production systems, but it might be

sufficient for a test queue manager. See section “Migrating from an earlier unsupported release of WebSphere MQ or MQSeries” on page 63 for further information. Ensure that production systems are fully migrated to Version 6.0 and stability is achieved before migration to Version 7.0.

## Migrating to WebSphere MQ for z/OS Version 7.0

You might need to perform some additional tasks when migrating from a previous version, depending on which of the new features you want to use, and which level of WebSphere MQ you are migrating from. Generally, the earlier the version of WebSphere MQ you are migrating from, the more tasks you need to perform.

The new functions for this release are described in the *WebSphere MQ for z/OS Concepts and Planning Guide*. Consider which of these functions you want to use before customizing WebSphere MQ because you might not need to perform all the migration tasks.

When you migrate from a previous version of WebSphere MQ for z/OS or MQSeries for OS/390®, you can continue to use your existing queue managers with the new version, including their page sets, log data sets, object definitions, and initialization input data sets. You can continue to use your existing queues, including system queues such as the SYSTEM.CHANNEL.SYNCQ.

**Do not cold start your queue managers when migrating from a previous version. A cold start deletes and redefines log files and page sets. This action destroys all messages and other information such as channel state.**

If you use the Client Attachment feature, you must install and use the V7 feature. Previous versions of this feature will not work with WebSphere MQ Version 7.0.

If you are using IMS™ (other than the WebSphere MQ IMS bridge), restart the IMS control region after the migration to ensure that you use the new version of any changed files.

### Additional considerations when you migrate to Version 7.0

#### Change in behavior of client connections to WebSphere MQ for z/OS:

In WebSphere MQ Version 7.0 you can attach a maximum of five client connections to WebSphere MQ for z/OS without having the Client Attachment feature installed.

In previous releases of WebSphere MQ you needed to install the optional Client Attachment feature (CAF) on z/OS in order to attach clients.

If you have the CAF installed you can attach a maximum of 999 999 999 clients.

If you do not have the CAF installed you can attach a maximum of five clients only by setting the MAXINST attribute on the server-connection channel to a value between zero and five. A value greater than five is interpreted as zero without the CAF installed.

If you decide to attach a limited number of clients without the CAF you must secure the server-connection channel; see WebSphere MQ Clients for information about server-connection security.

### Considerations when using the MQ Explorer to manage WebSphere MQ for z/OS queue managers:

If you use the MQ Explorer to administer your z/OS queue manager then after migration between releases the following actions must be performed on the MQ Explorer.

- Start the MQ Explorer.
- Right click the Queue Managers folder icon in the Navigator pane.
- Remove the migrated z/OS queue manager from the list of managed queue managers using the Hide option within the Hide/Show Queue Managers function.
- Add the migrated z/OS queue manager back into the list by using the Show option within the Hide/Show Queue Managers function.

This must be done when migrating from Version 6 to Version 7 or when migrating back from Version 7 to Version 6. These actions ensure that the MQ Explorer's cached view of the queue manager attributes, including the command level are correctly refreshed after the migration.

### Migrating WebSphere MQ for z/OS RACF security profiles:

This topic describes how to migrate RACF security profiles to WebSphere MQ Version 7.0.

You must ensure that you have addressed the following points when you migrate your RACF profiles to WebSphere MQ Version 7.0:

- External Security Manager software is at the correct version and level and that all of your prerequisite software is installed.
- WebSphere MQ RACF classes have been updated to include the new mixed case classes. For more information, see the RACF information in *WebSphere MQ z/OS Concepts and Planning Guide*
- Enterprise has migrated to mixed case security - see Migration to mixed case security.

### Migration to mixed case security:

If you want to use the mixed case security profile support follow these steps:

1. Ensure that you have installed a level of the security product that supports mixed case security, and that you have applied any updates required by WebSphere MQ.
2. Ensure that the new WebSphere MQ RACF classes are installed and active.
3. Copy all your existing profiles and access levels from the uppercase classes to the equivalent mixed case RACF class:
  - a. MQADMIN to MXADMIN
  - b. MQPROC to MXPROC
  - c. MXLIST to MXNLIST
  - d. MQQUEUE to MXQUEUE
4. Start your queue manager with the queue manager SCYCASE attribute set to UPPER.
5. Change the value of the SCYCASE attribute to MIXED by issuing the following command:

```
ALTER QMGR SCYCASE(MIXED)
```

6. Activate your existing security profiles by issuing the following command:  
REFRESH SECURITY(\*) TYPE(CLASSES)
7. Test that your security profiles are working correctly.
8. Review your object definitions and create new mixed case profiles as appropriate, using REFRESH SECURITY as required to activate the profiles.

### Exploiting 64-bit storage:

WebSphere MQ Version 7 uses z/OS memory objects above the bar for some functions. You must therefore allow the queue manager to access storage above the bar.

Your installation might have customized the SMFPRMxx member of SYS1.PARMLIB or the IEFUSI exit to provide a default limit for jobs using virtual storage above the 2 GB bar.

If your installation does not have a default limit for storage above the bar, or if you want to use a different limit for your queue manager, you can provide a queue manager-specific restriction on the amount of virtual storage available above the bar for memory objects by coding a MEMLIMIT parameter on the JCL of the queue manager stored procedure, xxxxMSTR, for example:

```
//PROCSTEP EXEC PGM=CSQYASCP,REGION=0M,MEMLIMIT=2G
```

**Migrating from AMI:** AMI is not supplied as part of WebSphere MQ Version 6.0 and later. If you have previously used AMI in WebSphere MQ Version 5.3.1, you can continue to use it by including the Version 5.3.1 SCSQLOAD file in your STEPLIB concatenation, as shown below:

```
//STEPLIB DD DISP=SHR,DSN=PP.MQM.V600.SCSQANLE V6 Messages
// DD DISP=SHR,DSN=PP.MQM.V600.SCSQAUTH V6 API MODULES
// DD DISP=SHR,DSN=PP.MQM.V531.SCSQLOAD AMI CODE
```

If you have removed WebSphere MQ Version 5.3.1 but need AMI functionality, you can install the AMI SupportPac MA0F. This is available from the IBM Web site: <http://www.ibm.com/software/integration/wmq/support/>

## Migrating from Version 6.0

If you are migrating from Version 6.0 to Version 7.0, you must consider the following when you customize your new version.

### Software levels

If you are migrating from WebSphere MQ for z/OS Version 5.3.1 to Version 7.0, check that you have the correct levels of prerequisite and corequisite software.

The minimum levels for some of the items of software required to use WebSphere MQ have changed. Check that you have the correct levels from the list in the *WebSphere MQ for z/OS Concepts and Planning Guide*.

### Testing for distributed queuing

You can use the supplied installation verification program, CSQ4IVPX, to confirm that distributed queuing is operational.



## CICS users

If you are using CICS® Transaction Server for z/OS Version 3.2 or later, the connection between WebSphere MQ and CICS is provided by CICS libraries. See the CICS Transaction Server for z/OS Version 3.2 Information Center at: <http://publib.boulder.ibm.com/infocenter/cicsts/v3r2/index.jsp> for further information.

If you are using a CICS version lower than V3.2, ensure you restart that CICS region after updating the WebSphere MQ libraries to V7 so that CICS uses new versions of any changed files.

## WebSphere Event Broker and WebSphere Message Broker users

WebSphere MQ Version 7.0 provides a queued publish/subscribe interface to support the migration of publish/subscribe applications from a WebSphere MQ Version 6.0 environment (where publish/subscribe is performed by the broker) to an environment where publish/subscribe is supported by function in WebSphere MQ Version 7.0.

The queue manager attribute PSMODE controls the queued publish/subscribe interface. If you set PSMODE to COMPAT, the interface is not started. Therefore, if you want to use WebSphere Event Broker Version 6.0 or WebSphere Message Broker Version 6.0 with WebSphere MQ Version 7.0, ensure PSMODE is set to COMPAT.

If you set PSMODE to ENABLED, the queued publish/subscribe interface is started and processes any broker control messages on the SYSTEM.BROKER.CONTROL.QUEUE.

In WebSphere MQ Version 7.0 the endmqbrk command can no longer be used.

## Migrating queue-sharing groups to Version 7.0

This might affect Tasks 9, 10 and 16 in Chapter 1, "Customizing your queue managers" in the *WebSphere MQ for z/OS System Setup Guide*.

The steps outlined below are designed to let you migrate one or more existing queue-sharing groups containing Version 6.0 queue managers to Version 7.0. The sequence has been designed so that at no stage is an outage of the entire queue-sharing group required. Version 6.0 and Version 7.0 queue managers can coexist within a queue-sharing group, however, some functions are not supported on the Version 6.0 queue managers and some operations are not available until all queue managers in the queue-sharing group have been migrated to Version 7.0. See "Multiple queue manager versions in a queue-sharing group" on page 16.

When all queue managers in the queue-sharing group have been migrated to Version 7.0 you can take full advantage of new Version 7.0 function.

### Applying WebSphere MQ for z/OS Version 7.0 migration & coexistence PTFs:

**Note:** This step can be performed at any suitable time in preparation for a migration to WebSphere MQ for z/OS Version 7.0 or as part of normal maintenance. It is not dependent on Version 7.0 being available.

You cannot add a Version 7.0 queue manager to a queue-sharing group, or start an existing queue manager in a queue-sharing group at Version 7.0 level, until all the

queue managers in a queue-sharing group within the DB2® data-sharing group have had a migration & coexistence PTF applied. This is because Version 7.0 requires new DB2 tables and additional changes to existing DB2 tables.

Similarly, after a Version 7.0 queue manager has been started in a queue-sharing group you cannot start a Version 6.0 queue manager as a member of the group unless it has a migration & coexistence PTF applied.

You must take the following steps:

1. Apply the PTF to the Version 6.0 libraries.
2. The PTF changes some of the DB2 operations performed by the Version 6.0 queue manager so that it is compatible with WebSphere MQ for z/OS Version 7.0. This means that the PTF contains some replacement DBRMs. Bind these DBRMs into new plans with a 600 version number (as detailed in the job supplied in the HOLDDATA of the PTF). The new plan names are as listed in the sample program CSQ45BPL, which is supplied with the PTF.  
This means that you have two sets of plans: one set for queue managers without the PTF and the other set for queue managers with the PTF applied. Module CSQ5PLAN (and its aliases) also changes the DB2 plans to be used by WebSphere MQ in the PTF names.
3. Bind the DBRMs supplied in the PTF into new version plans using the job supplied with the PTF.
4. Grant execute authority on new DB2 plans to the same user IDs as for existing Version 6.0 plans, using the job supplied with the PTF.
5. By turn, stop each queue manager and restart it so that it picks up the new code level.
6. Perform testing of the new code level.

#### **Migrating DB2 tables:**

**Note:** This step cannot be performed until all queue managers defined in a queue-sharing group within the DB2 data-sharing group have been started with the migration & coexistence PTF applied.

If you have queue managers defined in the queue-sharing group that cannot be started with the PTF, they can be removed from the queue-sharing group using the CSQ5PQSG utility.

If these jobs fail because of a DB2 locking problem, it is probably because of contention for a DB2 resource, especially if the system is being heavily used. Resubmit the job later, preferably when the system is lightly used or quiesced.

*To migrate the tables and all queue-sharing groups:*

Upgrading from WebSphere MQ V6 requires additional steps to migrate DB2 tables.

1. Customize and run the CSQ45ATB sample JCL in thlqual.SCSQPROC supplied with WebSphere MQ for z/OS Version 7.0. This performs the following steps:
  - a. Bind the new DB2 plan for the CSQ5PQSG utility.
  - b. Grant execute authority to the DB2 plan.
  - c. Check that the data-sharing group is in a state suitable for migration, using the MIGRATE DSG function.

- d. Modify the existing tables, and create the new tables required for WebSphere MQ for z/OS Version 7.0.
2. Bind the Version 7.0 DBRMs into plans and grant execute authority to them using the supplied jobs CSQ45BPL and CSQ45GEX, as described in Task 9: Set up the DB2 environment.

*To migrate one queue-sharing group at a time:*

These are the tasks that you perform to migrate a single queue-sharing group within the DB2 data-sharing group.

1. Customize the CSQ45ATB sample JCL in thlqual.SCSQPROC supplied with WebSphere MQ for z/OS Version 7.0, editing the step that executes the MIGRATE QSG function and specifying the name of the first queue-sharing group that is to be migrated. Run the job. This performs the following steps:
  - a. Bind the new DB2 plan for the CSQ5PQSG utility.
  - b. Grant execute authority to the DB2 plan.
  - c. Check that the queue-sharing group is in a state suitable for migration, using the MIGRATE QSG function.
  - d. Modify the existing tables, and create the new tables required for WebSphere MQ for z/OS Version 7.0.
2. Bind the Version 7.0 DBRMs into plans and grant execute authority to them using the supplied jobs CSQ45BPL and CSQ45GEX, as described in zs10420\_.dita.
3. To migrate subsequent queue-sharing groups, use the MIGRATE QSG function of CSQ5PQSG for each queue-sharing group to be migrated. It is not necessary to run CSQ45ATB for any queue-sharing group other than the first to be migrated.

## Setting up a new queue-sharing group

This might affect Tasks 6, 9, 10, 13, 16, and 17 in Chapter 1, "Customizing your queue managers" in the *WebSphere MQ for z/OS System Setup Guide*.

If you want to set up a new queue-sharing group or data-sharing group in WebSphere MQ Version 7.0, follow these steps:

1. Review your DB2 data-sharing requirements.
 

A single DB2 data-sharing group can be used to support multiple WebSphere MQ queue-sharing groups. If you intend to add this new queue-sharing group to a DB2 data-sharing group that already supports WebSphere MQ queue-sharing groups, you need to ensure that the migration & coexistence PTF has been applied, so that the DB2 tables used by WebSphere MQ can support Version 7.0 queue managers (see "Applying WebSphere MQ for z/OS Version 7.0 migration & coexistence PTFs" on page 55).
2. Customize the CSQ45ATB sample JCL in thlqual.SCSQPROC supplied with WebSphere MQ for z/OS Version 7.0, deleting or bypassing the step that executes the MIGRATE QSG function. Run the job.
3. Set up the Coupling Facility. See .
4. Add the WebSphere MQ entries to the DB2 data-sharing group using the CSQ5PQSG program. See .
5. Tailor the system parameter module to add DB2 data-sharing group and WebSphere MQ queue-sharing group information. See .
6. Customize and include the initialization input sample thlqual.SCSQPROC (CSQ4INSS) in the CSQINP2 data set.

See the *WebSphere MQ for z/OS System Administration Guide* for more information about managing queue-sharing groups.

## **CF structures**

This might affect Task 10 in Chapter 1, "Customizing your queue managers" in the *WebSphere MQ for z/OS System Setup Guide*.

WebSphere MQ local queue objects defined with QSGDISP(SHARED) have their messages stored on a Coupling Facility (CF) list structure so that they can be accessed by any other queue managers within the queue-sharing group.

As with the CFSTRUCT attribute of queues, the name is specified without the initial four-character queue-sharing group name that forms the name used by z/OS.

WebSphere MQ Version 6.0 and later support large messages on shared queues (up to 100 MB) and use DB2 to store data from those messages. CF structures supporting large shared queue messages must have the new CFLEVEL(4).

### **CFLEVEL(4) function:**

Shared queues defined on a CFLEVEL(4) CF structure can hold messages of up to 100 MB. Shared queues of a lower CFLEVEL can only hold messages of up to 63 KB.

A CFSTRUCT with CFLEVEL(4) can only be backed up or recovered by a Version 6.0 or later queue manager. CF structure objects defined with CFLEVEL(4) are only usable by Version 6.0 or later queue managers.

### **CFLEVEL(3) functions:**

Queues defined on a CFLEVEL(3) CF structure can have the INDXTYPE(GROUPID) attribute.

Persistent messages can be stored on a queue defined on a CF structure with CFLEVEL(3) and the RECOVER(YES) attribute. The MQSC commands BACKUP CFSTRUCT and RECOVER CFSTRUCT are provided to support recovery. Ensure that regular backups of such structures are taken.

To use the RECOVER CFSTRUCT command, all the active and archive log data sets, and bootstrap data sets of each queue manager in a queue-sharing group must be accessible to each other. You must ensure that your security setup allows this, and use Access Method Services to change the SHAREOPTIONS of these data sets to be (2 3), for example:

```
ALTER '++hlq++.logcopy.ds01.data' SHAREOPTIONS (2 3)
```

For information about other CFLEVELs, see the DEFINE CFSTRUCT command in WebSphere MQ Script (MQSC) Command Reference.

### **Recommendations:**

Once all queue managers in the queue-sharing group have been migrated to Version 7.0, alter all CF structure objects to have at least CFLEVEL(3). This gives greater resilience in the unlikely event of a Coupling Facility structure failure, as Version 7.0 queue managers can tolerate the failure of a CFLEVEL(3) CF structure.

## Channel initiators

This might affect Task 7 in Chapter 1, "Customizing your queue managers" in the *WebSphere MQ for z/OS System Setup Guide*.

### Channel initiator parameters:

The channel initiator parameters, which used to be specified in a channel initiator parameter module using CSQ6CHIP, are now queue manager attributes. To migrate your parameters, use the XPARM function of the WebSphere MQ utility program, CSQUTIL, which converts your parameter module into an **ALTER QMGR MQSC** command. The PARM parameter on START CHINIT is now obsolete. A warning message will be issued if this parameter is used, but the command will continue.

The minimum value of the TRAXTBL attribute has changed from 0 to 2 (2 MB). If your value of TRAXTBL is less than 2, CSQUTIL will change it to 2 during migration.

The initial value of the ADOPTMCA attribute has changed from NO to ALL. To retain the current behavior, use the ALTER QMGR command to change it to NO before starting the channel initiator.

For Version 5.3 users, there are new attributes RCVTIME, RCVTTYPE and RCVTMIN.

### TCP/IP Interface:

Only one type of TCP/IP interface is now supported, which is that formerly specified by TCPTYPE=OESOCKET in the channel initiator parameters.

*Migration from IUCV to OESOCKET:*

#### MQ Steps

1. Use the ALTER QMGR command to set the TCPNAME attribute to be the name of the OpenEdition stack for TCP/IP if this is not the same name as the TCP/IP address space.

#### Interlink Steps

1. Ensure you are using Interlink V5.2 or greater.
2. Create an OE segment for your channel initiator's user ID.
3. Check your BPXPARMs to ensure that MAXSOCKET is set high enough (the Interlink documentation suggests MAXSOCKETS of 4096).
4. It is no longer necessary to run an IUCV job.
5. For specific details of configuration steps for UNIX<sup>®</sup> System Services, check the Interlink documentation.

*Migration from SNSTCPACCESS to OESOCKET:*

#### MQ Steps

1. Change the channel initiator started task procedure (xxxxCHIN) to use hlq.SCSQMVR1 instead of hlq.SCSQMVR2.
2. Use the ALTER QMGR command to set the TCPNAME attribute to be the name of the OpenEdition stack for TCP/IP.

## Interlink Steps

1. Ensure you are using Interlink V5.2 or greater.
2. Create an OE segment for your channel initiator's user ID.
3. Check your BPXPARMs to make ensure that MAXSOCKET is set high enough (the Interlink book suggests MAXSOCKETS of 4096).
4. For specific details of configuration steps for UNIX System Services, check the Interlink documentation.

## Enable or disable channel event messages:

The queue manager attributes CHLEV, BRIDGEEV and SSLEV are introduced. These attributes enable you to control whether to generate channel events, IMS Bridge events and SSL events. By default, all these attributes are set to "ENABLED" to retain the behavior seen in previous releases. For more information about setting these attributes, see WebSphere MQ Script (MQSC) Command Reference.

## Channel attributes:

Heartbeat interval is now supported for server-connection channels. By default, existing such channels will have heartbeats set on with an interval of 300 seconds.

Although message retry is now supported for receiving channels, existing such channels will have message retry set to "no retry". However, if these channels are subsequently deleted and redefined they will have message retry set on.

## Commands

### PCF:

PCF commands, identified by a message format of MQFMT\_ADMIN, are now supported.

MQSC commands put to the system command input queue with a message format of MQFMT\_ADMIN were accepted in previous versions. In Version 6, such messages will be treated as PCF commands, and so will be rejected.

Applications currently using PCF will need changing to use PCF on z/OS, to use the necessary extended command formats and handle the extended response formats. See WebSphere MQ Programmable Command Formats and Administration Interface for details.

### Security profiles:

The MQSC command DISPLAY DQM has been renamed DISPLAY CHINIT, although the old command still works as a synonym. If you have a security profile for the old command, you should rename it.

There are several new MQSC commands, and several commands now allowed outside the queue manager initialization data sets, for which you should add security profiles:

- DISPLAY CONN
- ALTER/DEFINE/DELETE PSID
- ALTER/DEFINE/DELETE BUFFPOOL
- DEFINE LOG

Most MQSC commands now have PCF equivalents. These generally use the same security profiles but there are some which need new profiles, for example Change Queue. For further information see Profiles for command security.

If you are using the WebSphere MQ Version 6.0 Explorer or later, you need appropriate security profiles for the queues SYSTEM.MQEXPLORER.REPLY.MODEL and AMQ.EXPLORER.\*. For further information see Profiles used to control access to WebSphere MQ resources.

### **Channel Initiator:**

Channel initiator commands (other than START CHINIT) are no longer allowed in the queue manager initialization input data set CSQINP2. They should be moved to the channel initiator initialization input data set CSQINPX. Output from these commands is written to the output data set CSQOUTX. For more details, see Task 13: Customize the initialization input data sets in Chapter 1, "Customizing your queue managers" in the *WebSphere MQ for z/OS System Setup Guide*.

The START TRACE, DISPLAY TRACE, ALTER TRACE, and STOP TRACE commands have been changed to handle channel initiator traces better. For information about WebSphere MQ commands, see the WebSphere MQ Script (MQSC) Command Reference manual.

### **Initialization input data sets CSQINP1 and CSQINP2**

This affects Tasks 6 and 13 in Chapter 1, "Customizing your queue managers" in the *WebSphere MQ for z/OS System Setup Guide*.

The supplied samples have been extended to provide extra object definitions. There are more samples and the contents of some have been moved to other samples. Review them and update the customized versions you are currently using as required.

In WebSphere MQ Version 7.0 the ordering of definitions becomes important. Member CSQ4MSTR of SCSQPROC provides a recommended ordering of supplied definitions in the CSQINP2 concatenation. If you are migrating from a previous release, the following ordering is important for correct publish/subscribe operation:

1. Storage class SYSLNGLV must be defined mapping to a defined pageset.
2. SYSTEM.DURABLE.SUBSCRIBER.QUEUE (which uses SYSLNGLV) is defined
3. SYSTEM.DEFAULT.SUB (a durable subscription) is defined

### **DB2 plan names**

Plan names for accessing the DB2 tables containing the WebSphere MQ object descriptions have changed. You need to bind the new plans in order to access the tables. Member CSQ45BPL of hlq.SCSQPROC contains the plan names required for WebSphere MQ Version 7.0. Tailor this member to your environment, using your DB2 subsystem names and data set names, and submit it.

If this job fails because of a DB2 locking problem, it is probably due to contention for a DB2 resource, especially if the system is being heavily used. Resubmit the job later, preferably when the system is lightly used or quiesced.

## Clustering

The cluster cache can now be dynamic. See the description of CLCACHE in *WebSphere MQ for z/OS System Setup Guide*.

## CICS

### Distributed queueing:

The CICS Mover feature, and therefore distributed queueing using CICS ISC, has been removed. You are recommended to remove any associated CICS resource definitions that you have (such as those previously supplied in the sample CSQ4D100 in thlqual.SCSQPROC) and any associated MQ object definitions (such as those previously supplied in sample CSQ4DISQ in thlqual.SCSQPROC).

### Bridge:

Multiple bridge monitors can now service a single request queue, see *WebSphere MQ for z/OS System Setup Guide*. If the bridge is started by terminal input, the terminal is unlocked immediately, rather than staying locked until the transaction ends.

## JMS resources

JNDI .jar files are no longer supplied. Instead, you use the .jar files supplied with the JMS client. Existing JMS resources created with the IBM JNDI facilities will continue to run. However, you cannot edit them with the JMS client JNDI facilities. You will have to recreate them using the new facilities. Similarly, you cannot delete them using the JMS client JNDI facilities. Instead you must first recreate them using the new facilities and then delete the recreated resources.

## Changing to full function WebSphere MQ

If you are migrating from the reduced function form of WebSphere MQ V5.3.1 supplied with WebSphere Application Server, you should read all of *WebSphere MQ for z/OS Concepts and Planning Guide* and this book to find out about the extra functions that are available. You can continue to use your existing queue managers, including their page sets, log data sets, and object definitions. You can continue to use your existing queues, including system queues such as SYSTEM.CHANNEL.SYNCQ. You should not cold start your queue managers. If you do, you will lose all your messages and other information. However, there are some tasks that you might need to perform, depending on which of the now available features you want to use. Review all the customization tasks in Chapter 1, "Customizing your queue managers" in the *WebSphere MQ for z/OS System Setup Guide*, but especially:

- Task 13: Customize the initialization input data sets
- Task 17: Tailor your system parameter module
- Task 18: Tailor the channel initiator parameters

Also review:

- Chapter 4, "Customizing for CICS"
- Chapter 5, "Customizing for IMS"
- Chapter 7, "Setting up security"



If you are going to use queue-sharing groups, see . If you used client channels in the reduced function WebSphere MQ, you must install the Client Attachment feature to allow them to continue to be used.

## Reverting to previous versions

This section tells you what to do if you want to revert to using a previous version of WebSphere MQ for z/OS for some exceptional reason.

If you choose to revert to a previous version of WebSphere MQ, note that, as a general rule, data such as attributes (or in some cases, objects) relating to the new function in Version 7.0 will be lost. The more you have used the new functions of Version 7.0, the less practical it will be to go back to an earlier version.

If you want to revert to an earlier version of WebSphere MQ, contact your IBM support center. The support center will provide you with the 'Backward Migration PTF', which you must apply to your system to revert to an earlier version. The information in the PTF includes a description of the data relating to new function that will be lost.

When you have reverted to a previous version, if you have defined a CF structure with CFLEVEL(4), that structure, queues defined to use it, and any messages on those queues are not accessible while running with a version of code earlier than Version 6.0. The queues and messages are retained, but if the Coupling Facility fails and all members of the queue-sharing group have been reverted to a previous level, messages might be lost.

## Migrating from an earlier unsupported release of WebSphere MQ or MQSeries

### Migrating from earlier unsupported releases of MQSeries

For production systems, the recommended migration path from an unsupported release of MQSeries is first to migrate to WebSphere MQ Version 6.0, following the instructions given in the documentation for that release, and then to migrate to Version 7.0 following the instructions in this chapter. Ensure that your system is stable at Version 6.0 before migrating to Version 7.0, so that you have a system to revert to if you need to. Details about reversion are given in this chapter:

- "Reverting to previous versions"

For a test system, it might be appropriate to migrate straight to WebSphere MQ Version 7.0. The additional steps that you must consider when migrating from unsupported releases of MQSeries are as follows:

- "Additional steps when migrating from Version 5.3.1" on page 64
- "Additional steps when migrating from Version 5.2 to Version 5.3.1" on page 73
- "Additional steps when migrating from Version 2.1 to Version 5.3.1" on page 80
- "Additional steps when migrating from Version 1.2 to Version 5.3.1" on page 82

After you have migrated to WebSphere MQ Version 7.0 using this method, you cannot revert to the original version. You should take complete backups of your system to ensure you can restart from backups if you need to use the old release again.

When WebSphere MQ page sets and log data sets have been used by a queue manager running at Version 7.0, the effect of trying to use those data sets with a queue manager that has been reverted to use Version 5.3.1 or earlier is undefined and the queue manager will fail to start if it can detect this has happened.

You can restart a queue manager running at Version 5.3.1 or earlier using the full set of backups taken before migration. Any changes made to the system after the backups were taken or while running at Version 7.0 will be lost.

This collection of topics about migrating from unsupported releases has been taken verbatim from earlier versions of the *WebSphere MQ for z/OS System Setup Guide*.

## **Migrating to WebSphere MQ for z/OS Version 6**

You might need to perform some additional tasks when migrating from a previous version, depending on which of the new features you want to use, and which level of WebSphere MQ you are migrating from. Generally, the earlier the version of WebSphere MQ you are migrating from, the more tasks you need to perform.

The functions for this release are described in the *WebSphere MQ for z/OS Concepts and Planning Guide* for WebSphere MQ Version 6.0. If you are migrating to Version 6.0 before migrating to Version 7.0, consider which of these functions you want to use before customizing WebSphere MQ because you might not need to perform all the migration tasks.

When you migrate from a previous version of WebSphere MQ for z/OS or MQSeries for OS/390, you can continue to use your existing queue managers with the new version, including their page sets, log data sets, object definitions, and initialization input data sets. You can continue to use your existing queues, including system queues such as the SYSTEM.CHANNEL.SYNQCQ. **You should not cold start your queue managers when migrating from a previous version because you do not need to. If you do, you will lose all your messages and other information such as channel state.**

If you use the Client Attachment feature, you must install and use the V6 feature. Previous versions of this feature will not work with WebSphere MQ Version 6.0.

If you are using CICS, restart the CICS region after the migration to ensure that you use the new versions of any changed files. Likewise, if you are using IMS (other than the WebSphere MQ IMS bridge), restart the IMS control region after the migration.

### **Additional steps when migrating from Version 5.3.1**

If you are migrating from Version 5.3.1 to Version 7.0, you must consider the following when you customize your new version.

#### **Software levels:**

If you are migrating from WebSphere MQ for z/OS Version 5.3.1 to Version 7.0, check that you have the correct levels of prerequisite and corequisite software.

The minimum levels for some of the items of software required to use WebSphere MQ have changed. Check that you have the correct levels from the list in the *WebSphere MQ for z/OS Concepts and Planning Guide*.

#### **Libraries:**

This might affect Task 19 in Chapter 1, "Customizing your queue managers" in the *WebSphere MQ for z/OS System Setup Guide*.

The library thlqual.SCSQMVR2 is no longer used.

The Version 5.3 library thlqual.SCSQSKL is no longer used.

### **Migrating queue-sharing groups to Version 7.0:**

This might affect Tasks 9, 10 and 16 in Chapter 1, "Customizing your queue managers" of the *WebSphere MQ for z/OS System Setup Guide*.

The steps outlined below are designed to let you migrate one or more existing queue-sharing groups containing Version 5.3.1 queue managers to Version 7.0. The sequence has been designed so that at no stage is an outage of the entire queue-sharing group required. Version 5.3.1 and Version 7.0 queue managers can coexist within a queue-sharing group, however, some functions are not supported on the Version 5.3.1 queue managers and some operations are not available until all queue managers in the queue-sharing group have been migrated to Version 7.0. See "Multiple queue manager versions in a queue-sharing group" on page 16.

After all queue managers in the queue-sharing group have been migrated to Version 7.0 you can take full advantage of new Version 7.0 function.

*Applying MQSeries for OS/390 Version 5.3.1 migration & coexistence PTFs:*

**Note:** This step can be performed at any suitable time in preparation for a migration to WebSphere MQ for z/OS Version 7.0 or as part of normal maintenance. It is not dependant on Version 7.0 being available.

You cannot add a Version 7.0 queue manager to a queue-sharing group, or start an existing queue manager in a queue-sharing group at Version 7.0 level, until all the queue managers in a queue-sharing group within the DB2 data-sharing group have had a migration & coexistence PTF applied. This is because Version 7.0 requires new DB2 tables and additional changes to existing DB2 tables.

Similarly, once a Version 7.0 queue manager has been started in a queue-sharing group you cannot start a Version 5.3.1 queue manager as a member of the group unless it has a migration & coexistence PTF applied.

You need to take the following steps:

1. Apply the PTF.
2. The PTF changes some of the DB2 operations performed by the Version 5.3.1 queue manager so that it is compatible with WebSphere MQ for z/OS Version 7.0. This means that the PTF contains some replacement DBRMs. You should bind these DBRMs into new plans with a 531 version number (as detailed in the job supplied in the HOLDDATA of the PTF). The new plan names are as listed in the sample program CSQ45BPL, which is supplied with the PTF.

This means that you have two sets of plans, for queue managers without the PTF and for queue managers with the PTF applied. Module CSQ5PLAN (and its aliases) also changes the DB2 plans to be used by WebSphere MQ in the PTF names.

3. Bind the DBRMs supplied in the PTF into new version plans using the job supplied with the PTF.

4. Grant execute authority on new DB2 plans to the same userids as for existing Version 5.3.1 plans, using the job supplied with the PTF.
5. By turn, stop each queue manager and restart it so that it picks up the new code level.
6. Perform testing of the new code level.

*Migrating DB2 tables:*

**Note:** This step cannot be performed until all queue managers defined in a queue-sharing group within the DB2 data-sharing group have been started with the migration & coexistence PTF applied.

If you have queue managers defined in the queue-sharing group that cannot be started with the PTF, they can be removed from the queue-sharing group using the CSQ5PQSG utility.

If these jobs fail because of a DB2 locking problem, it is probably because of contention for a DB2 resource, especially if the system is being heavily used. Resubmit the job later, preferably when the system is lightly used or quiesced.

*To migrate the tables and all queue-sharing groups:*

Upgrading from WebSphere MQ V6 requires additional steps to migrate DB2 tables.

1. Customize and run the CSQ45ATB sample JCL in thlqual.SCSQPROC supplied with WebSphere MQ for z/OS Version 7.0. This performs the following steps:
  - a. Bind the new DB2 plan for the CSQ5PQSG utility.
  - b. Grant execute authority to the DB2 plan.
  - c. Check that the data-sharing group is in a state suitable for migration, using the MIGRATE DSG function.
  - d. Modify the existing tables, and create the new tables required for WebSphere MQ for z/OS Version 7.0.
2. Bind the Version 7.0 DBRMs into plans and grant execute authority to them using the supplied jobs CSQ45BPL and CSQ45GEX, as described in Task 9 in Chapter 1, "Customizing your queue managers" in the *WebSphere MQ for z/OS System Setup Guide*.

*To migrate one queue-sharing group at a time:*

Upgrading from WebSphere MQ Version 6 requires additional steps to migrate DB2 tables.

1. Customize the CSQ45ATB sample JCL in thlqual.SCSQPROC supplied with WebSphere MQ for z/OS Version 7.0, editing the step that executes the MIGRATE QSG function and specifying the name of the first queue-sharing group that is to be migrated. Run the job. This performs the following steps:
  - a. Bind the new DB2 plan for the CSQ5PQSG utility.
  - b. Grant execute authority to the DB2 plan.
  - c. Check that the queue-sharing group is in a state suitable for migration, using the MIGRATE QSG function.
  - d. Modify the existing tables, and create the new tables required for WebSphere MQ for z/OS Version 7.0.

2. Bind the Version 7.0 DBRMs into plans and grant execute authority to them using the supplied jobs CSQ45BPL and CSQ45GEX, as described in Task 9 in Chapter 1, "Customizing your queue managers" in the *WebSphere MQ for z/OS System Setup Guide*.
3. To migrate subsequent queue-sharing groups, use the MIGRATE QSG function of CSQ5PQSG for each queue-sharing group to be migrated. It is not necessary to run CSQ45ATB for any queue-sharing group other than the first to be migrated.

#### **Setting up a new queue-sharing group:**

This might affect Tasks 6, 9, 10, 13, 16, and 17 in Chapter 1, "Customizing your queue managers" in the *WebSphere MQ for z/OS System Setup Guide*.

If you want to set up a new queue-sharing group or data-sharing group in WebSphere MQ Version 7.0, follow these steps:

1. Review your DB2 data-sharing requirements.  
A single DB2 data-sharing group can be used to support multiple WebSphere MQ queue-sharing groups. If you intend to add this new queue-sharing group to a DB2 data-sharing group that already supports WebSphere MQ queue-sharing groups, you need to ensure that the migration & coexistence PTF has been applied, so that the DB2 tables used by WebSphere MQ can support Version 7.0 queue managers (see "Applying MQSeries for OS/390 Version 5.3.1 migration & coexistence PTFs" on page 65).
2. Customize the CSQ45ATB sample JCL in thlqual.SCSQPROC supplied with WebSphere MQ for z/OS Version 7.0, deleting or bypassing the step that executes the MIGRATE QSG function. Run the job.
3. Set up the Coupling Facility. See Task 10 in Chapter 1, "Customizing your queue managers" in the *WebSphere MQ for z/OS System Setup Guide*.
4. Add the WebSphere MQ entries to the DB2 data-sharing group using the CSQ5PQSG program. See Task 16 in Chapter 1, "Customizing your queue managers" in the *WebSphere MQ for z/OS System Setup Guide*.
5. Tailor the system parameter module to add DB2 data-sharing group and WebSphere MQ queue-sharing group information. See Task 17 in Chapter 1, "Customizing your queue managers" in the *WebSphere MQ for z/OS System Setup Guide*.
6. Customize and include the initialization input sample thlqual.SCSQPROC (CSQ4INSS) in the CSQINP2 data set.

See the WebSphere MQ for z/OS System Administration Guide for more information about managing queue-sharing groups.

#### **CF structures:**

This might affect Task 10 in Chapter 1, "Customizing your queue managers" in the *WebSphere MQ for z/OS System Setup Guide*.

WebSphere MQ local queue objects defined with QSGDISP(SHARED) have their messages stored on a Coupling Facility (CF) list structure so that they can be accessed by any other queue managers within the queue-sharing group.

As with the CFSTRUCT attribute of queues, the name is specified without the initial four-character queue-sharing group name that forms the name used by z/OS.

WebSphere MQ Version 6.0 or later supports large messages on shared queues (up to 100 MB) and uses DB2 to store data from those messages. CF structures supporting large shared queue messages must have the new CFLEVEL(4).

*CFLEVEL(4) function:*

Shared queues defined on a CFLEVEL(4) CF structure can hold messages of up to 100 MB. Shared queues of a lower CFLEVEL can only hold messages of up to 63 KB.

A CFSTRUCT with CFLEVEL(4) can only be backed up or recovered by a Version 6.0 or later queue manager. CF structure objects defined with CFLEVEL(4) are only usable by Version 6.0 or later queue managers.

*CFLEVEL(3) functions:*

Queues defined on a CFLEVEL(3) CF structure can have the INDXTYPE(GROUPID) attribute.

Persistent messages can be stored on a queue defined on a CF structure with CFLEVEL(3) and the RECOVER(YES) attribute. The MQSC commands BACKUP CFSTRUCT and RECOVER CFSTRUCT are provided to support recovery. Ensure that regular backups of such structures are taken.

To use the RECOVER CFSTRUCT command, all the active and archive log data sets, and bootstrap data sets of each queue manager in a queue-sharing group must be accessible to each other. You must ensure that your security setup allows this, and use Access Method Services to change the SHAREOPTIONS of these data sets to be (2 3), for example:

```
ALTER '++h1q++.logcopy.ds01.data' SHAREOPTIONS (2 3)
```

For information about other CFLEVELs, see the DEFINE CFSTRUCT command in WebSphere MQ Script (MQSC) Command Reference.

*Recommendations:*

Once all queue managers in the queue-sharing group have been migrated to Version 7.0, alter all CF structure objects to have at least CFLEVEL(3). This gives greater resilience in the unlikely event of a Coupling Facility structure failure, as Version 7.0 queue managers can tolerate the failure of a CFLEVEL(3) CF structure.

### **Channel initiators:**

This might affect Task 7 in Chapter 1, "Customizing your queue managers" in the *WebSphere MQ for z/OS System Setup Guide*.

*Channel initiator parameters:*

The channel initiator parameters, which used to be specified in a channel initiator parameter module using CSQ6CHIP, are now queue manager attributes. To migrate your parameters, use the XPARM function of the WebSphere MQ utility program, CSQUTIL, which converts your parameter module into an ALTER QMGR MQSC command. The PARM parameter on START CHINIT is now obsolete. A warning message will be issued if this parameter is used, but the command will continue.

The minimum value of the TRAXTBL attribute has changed from 0 to 2 (2 MB). If your value of TRAXTBL is less than 2, CSQUTIL will change it to 2 during migration.

The initial value of the ADOPTMCA attribute has changed from NO to ALL. To retain the current behavior, use the ALTER QMGR command to change it to NO before starting the channel initiator.

For Version 5.3 users, there are new attributes RCVTIME, RCVTTYPE and RCVTMIN.

#### *TCP/IP Interface:*

Only one type of TCP/IP interface is now supported, which is that formerly specified by TCPTYPE=OESOCKET in the channel initiator parameters.

#### *Migration from IUCV to OESOCKET:*

##### MQ Steps

1. Use the ALTER QMGR command to set the TCPNAME attribute to be the name of the OpenEdition stack for TCP/IP if this is not the same name as the TCP/IP address space.

##### Interlink Steps

1. Ensure you are using Interlink V5.2 or greater.
2. Create an OE segment for your channel initiator's user ID.
3. Check your BPXPARMs to ensure that MAXSOCKET is set high enough (the Interlink documentation suggests MAXSOCKETS of 4096).
4. It is no longer necessary to run an IUCV job.
5. For specific details of configuration steps for UNIX System Services, check the Interlink documentation.

#### *Migration from SNSTCPACCESS to OESOCKET:*

##### MQ Steps

1. Change the channel initiator started task procedure (xxxxCHIN) to use hlq.SCSQMVR1 instead of hlq.SCSQMVR2.
2. Use the ALTER QMGR command to set the TCPNAME attribute to be the name of the OpenEdition stack for TCP/IP.

##### Interlink Steps

1. Ensure you are using Interlink V5.2 or greater.
2. Create an OE segment for your channel initiator's user ID.
3. Check your BPXPARMs to make ensure that MAXSOCKET is set high enough (the Interlink book suggests MAXSOCKETS of 4096).
4. For specific details of configuration steps for UNIX System Services, check the Interlink documentation.

#### *Enable or disable channel event messages:*

The queue manager attributes CHLEV, BRIDGEEV and SSLEV are introduced. These attributes enable you to control whether to generate channel events, IMS Bridge events and SSL events. By default, all these attributes are set to

"ENABLED" to retain the behavior seen in previous releases. For more information about setting these attributes, see WebSphere MQ Script (MQSC) Command Reference.

*Channel attributes:*

Heartbeat interval is now supported for server-connection channels. By default, existing such channels will have heartbeats set on with an interval of 300 seconds.

Although message retry is now supported for receiving channels, existing such channels will have message retry set to "no retry". However, if these channels are subsequently deleted and redefined they will have message retry set on.

**Commands:**

*PCF:*

PCF commands, identified by a message format of MQFMT\_ADMIN, are now supported.

MQSC commands put to the system command input queue with a message format of MQFMT\_ADMIN were accepted in previous versions. In Version 6, such messages will be treated as PCF commands, and so will be rejected.

Applications currently using PCF will need changing to use PCF on z/OS, to use the necessary extended command formats and handle the extended response formats. See WebSphere MQ Programmable Command Formats and Administration Interface for details.

*Security profiles:*

The MQSC command DISPLAY DQM has been renamed DISPLAY CHINIT, although the old command still works as a synonym. If you have a security profile for the old command, you should rename it.

There are several new MQSC commands, and several commands now allowed outside the queue manager initialization data sets, for which you should add security profiles:

- DISPLAY CONN
- ALTER/DEFINE/DELETE PSID
- ALTER/DEFINE/DELETE BUFFPOOL
- DEFINE LOG

Most MQSC commands now have PCF equivalents. These generally use the same security profiles but there are some which need new profiles, for example Change Queue. For further information see Profiles for command security.

If you are using the WebSphere MQ Version 6.0 Explorer or later, you need appropriate security profiles for the queues SYSTEM.MQEXPLORER.REPLY.MODEL and AMQ.EXPLORER.\*. For further information see Profiles used to control access to WebSphere MQ resources.

*Channel Initiator:*



Channel initiator commands (other than START CHINIT) are no longer allowed in the queue manager initialization input data set CSQINP2. They should be moved to the channel initiator initialization input data set CSQINPX. Output from these commands is written to the output data set CSQOUTX. For more details, see Task 13: Customize the initialization input data sets in Chapter 1, "Customizing your queue managers" in the *WebSphere MQ for z/OS System Setup Guide*.

The START TRACE, DISPLAY TRACE, ALTER TRACE, and STOP TRACE commands have been changed to handle channel initiator traces better. For information about WebSphere MQ commands, see the WebSphere MQ Script (MQSC) Command Reference manual.

### **Initialization input data sets CSQINP1 and CSQINP2:**

This affects Tasks 6 and 13 in Chapter 1, "Customizing your queue managers" in the *WebSphere MQ for z/OS System Setup Guide*.

The supplied samples have been extended to provide extra object definitions. There are more samples and the contents of some have been moved to other samples. Review them and update the customized versions you are currently using as required.

In WebSphere MQ Version 7.0 the ordering of definitions becomes important. Member CSQ4MSTR of SCSQPROC provides a recommended ordering of supplied definitions in the CSQINP2 concatenation. If you are migrating from a previous release, the following ordering is important for correct publish/subscribe operation:

1. Storage class SYSLNGLV must be defined mapping to a defined pageset.
2. SYSTEM.DURABLE.SUBSCRIBER.QUEUE (which uses SYSLNGLV) is defined
3. SYSTEM.DEFAULT.SUB (a durable subscription) is defined

### **DB2 plan names:**

Plan names for accessing the DB2 tables containing the WebSphere MQ object descriptions have changed. You need to bind the new plans in order to access the tables. Member CSQ45BPL of hlq.SCSQPROC contains the plan names required for WebSphere MQ Version 7.0. Tailor this member to your environment, using your DB2 subsystem names and data set names, and submit it.

If this job fails because of a DB2 locking problem, it is probably due to contention for a DB2 resource, especially if the system is being heavily used. Resubmit the job later, preferably when the system is lightly used or quiesced.

### **Clustering:**

The cluster cache can now be dynamic. See the description of CLCACHE in *WebSphere MQ for z/OS System Setup Guide*.

### **CICS:**

#### *Compilers:*

You can write WebSphere MQ applications using any compiler capable of generating standard OS linkage to the WebSphere MQ stub routines. WebSphere MQ Version 7.0 introduces many new WebSphere MQ APIs; some of the data types used by these APIs are not supported on some older compilers. While efforts have

been made to ensure that preexisting applications can be recompiled, if you modify applications to use new APIs, you might have to use a more recent compiler. Some known limitations are:

- Assembler copy books contain blank lines, which are not tolerated by assemblers earlier than HLASM
- Some older releases of PL/I do not support fixed bin(63) type. A macro has been provided to define such fields as char(8) when an earlier compiler is detected.
- Some older releases of COBOL do not support function-pointers, which are used by the MQCB API.

#### *Bridge:*

Multiple bridge monitors can now service a single request queue, see *WebSphere MQ for z/OS System Setup Guide*. If the bridge is started by terminal input, the terminal is unlocked immediately, rather than staying locked until the transaction ends.

#### **JMS resources:**

JNDI .jar files are no longer supplied. Instead, you use the .jar files supplied with the JMS client. Existing JMS resources created with the IBM JNDI facilities will continue to run. However, you cannot edit them with the JMS client JNDI facilities. You will have to recreate them using the new facilities. Similarly, you cannot delete them using the JMS client JNDI facilities. Instead you must first recreate them using the new facilities and then delete the recreated resources.

#### **Changing to full function WebSphere MQ:**

If you are migrating from the reduced function form of WebSphere MQ V5.3.1 supplied with WebSphere Application Server, you should read all of WebSphere MQ for z/OS Concepts and Planning Guide and this book to find out about the extra functions that are available. You can continue to use your existing queue managers, including their page sets, log data sets, and object definitions. You can continue to use your existing queues, including system queues such as SYSTEM.CHANNEL.SYNCQ. You should not cold start your queue managers. If you do, you will lose all your messages and other information. However, there are some tasks that you might need to perform, depending on which of the now available features you want to use. Review all the customization tasks in Chapter 1, "Customizing your queue managers" in the *WebSphere MQ for z/OS System Setup Guide*, but especially:

- Task 13: Customize the initialization input data sets
- Task 17: Tailor your system parameter module
- Task 18: Tailor the channel initiator parameters

Also review:

- Chapter 4, "Customizing for CICS"
- Chapter 5, "Customizing for IMS"
- Chapter 7, "Setting up security"

If you are going to use queue-sharing groups, see . If you used client channels in the reduced function WebSphere MQ, you must install the Client Attachment feature to allow them to continue to be used.

## Additional steps when migrating from Version 5.2 to Version 5.3.1

If you intend to migrate from Version 5.2 to Version 5.3.1, you must consider the following when you customize your new version (you do not need to install and customize Version 5.3):

### System parameters:

This might affect Task 16 in Chapter 1, "Customizing your queue managers" of the *WebSphere MQ for z/OS System Setup Guide*.

The system parameter QINDXBLD was introduced in Version 5.3. Consider whether you need to use this parameter, and change your system parameter module accordingly. If you do not need to use this parameter, you do not need to relinkedit your system parameter module.

Some system parameters can be changed while a queue manager is running. You might want to take advantage of this by setting them in the initialization input data sets.

### Queue-sharing groups:

If you have not previously used queue-sharing groups, do not introduce them now. Wait until you have migrated to Version 7.0. If you are using queue-sharing groups, you must migrate them to Version 5.3.1.

This might affect Tasks 8, 9, and 15 in Chapter 1, "Customizing your queue managers" of the *WebSphere MQ for z/OS System Setup Guide*.

The steps outlined below are designed to let you migrate an existing queue-sharing group containing Version 5.2 queue managers to Version 5.3.1. The sequence has been designed so that at no stage is an outage of the entire queue-sharing group required. Version 5.2 and 5.3.1 queue managers can coexist within a queue-sharing group, however, some functions are not supported on the Version 5.2 queue managers and some operations are not available until all queue managers in the queue-sharing group have been migrated to Version 5.3.1. See the *WebSphere MQ for z/OS System Setup Guide* for more information about the coexistence of different versions.

*Applying MQSeries for OS/390 Version 5.2 migration & coexistence PTFs:*

**Note:** This step can be performed at any suitable time in preparation for a migration to WebSphere MQ for z/OS Version 5.3.1 or as part of normal maintenance. It is not dependant on Version 5.3.1 being available.

You cannot add a Version 5.3.1 queue manager to a queue-sharing group, or start an existing queue manager in a queue-sharing group at Version 5.3.1 level, until all the queue managers in the DB2 data-sharing group have had a migration & coexistence PTF applied. This is because Version 5.3.1 requires new DB2 tables and additional changes to existing DB2 tables.

Similarly, once a Version 5.3.1 queue manager has been started in a queue-sharing group you cannot start a Version 5.2 queue manager as a member of the group unless it has the migration & coexistence PTF applied.

You need to take the following steps:

1. Apply the PTF.
2. The PTF changes some of the DB2 operations performed by the Version 5.2 queue manager so that it is compatible with WebSphere MQ for z/OS Version 5.3.1. This means that the PTF contains some replacement DBRMs and some new DBRMs. You should bind these DBRMs into new plans with a 221 version number (as detailed in the job supplied in the HOLDDATA of the PTF). For example:

```
BIND PLAN(CSQR221) -
MEMBER(CSQR220) -
...
```

binds replacement DBRM CSQR220 into a new plan CSQR221.

This means that you have two sets of plans, those with a 220 version number for queue managers without the PTF, and those with a 221 version number for queue managers with the PTF applied. Module CSQ5PLAN (and its aliases) also changes the DB2 plans to be used by WebSphere MQ in the PTF names, so after applying the migration & coexistence PTF, MQSeries for OS/390 Version 5.2 expects plans with a 221 version number to exist.

3. Bind the DBRMs supplied in the PTF into 221 version plans using the job supplied with the PTF, CSQ45B21.
4. Grant execute authority on new DB2 plans to the same userids as for existing 220 version plans, using the job supplied with the PTF, CSQ45G21.
5. By turn, stop each queue manager and restart it so that it picks up the new code level.
6. Perform testing of the new code level.

*Migrating DB2 tables:*

**Note:** This step cannot be performed until all queue managers defined in the DB2 data-sharing group have been started with the migration & coexistence PTF applied.

If you have queue managers defined in the data-sharing group that cannot be started with the PTF, they can be removed from the data-sharing group using the CSQ5PQSG utility.

To migrate the tables:

1. Customize and run the CSQ45ATB sample JCL in thlqual.SCSQPROC supplied with WebSphere MQ for z/OS Version 5.3.1. This performs the following steps:
  - a. Prepare the DB2 tables for the bind in the next step
  - b. Bind the new DB2 plan for the CSQ5PQSG utility.
  - c. Grant execute authority to the DB2 plan.
  - d. Check that the data-sharing group is in a state suitable for migration.
  - e. Create a temporary DB2 table and save the channel definitions.
  - f. Delete the existing channel table. Create a new, larger channel definition table and reload the saved definitions.
  - g. Modify the existing tables, and create the new tables required for WebSphere MQ for z/OS Version 5.3.1.
  - h. Rebind the DB2 plans for MQSeries Version 5.2.

2. Bind the Version 5.3.1 DBRMs into plans and grant execute authority to them using the supplied jobs CSQ45BPL and CSQ45GEX, as described in Task 9 in Chapter 1, "Customizing your queue managers" of the *WebSphere MQ for z/OS System Setup Guide*.
3. Rebind the DBRMs supplied with the PTF, by rerunning the job CSQ45B21 supplied with that PTF.

If these jobs fail because of a DB2 locking problem, it is probably due to contention for a DB2 resource, especially if the system is being heavily used. Resubmit the job later, preferably when the system is lightly used or quiesced.

#### **CF structures:**

This might affect Task 9 in Chapter 1, "Customizing your queue managers" in the *WebSphere MQ for z/OS System Setup Guide*.

WebSphere MQ local queue objects defined with QSGDISP(SHARED) have their messages stored on a Coupling Facility (CF) list structure so that they can be accessed by any other queue managers within the queue-sharing group. The Coupling Facility list structure is defined to z/OS by the CFRM policy. Version 5.3 introduced a WebSphere MQ object called a CF structure that describes capabilities through the CFLEVEL and RECOVER attributes. The MQSC commands DEFINE CFSTRUCT, ALTER CFSTRUCT, DELETE CFSTRUCT, and DISPLAY CFSTRUCT have been added for these objects.

As with the CFSTRUCT attribute of queues, the name is specified without the initial four-character queue-sharing group name that forms the name used by z/OS; all WebSphere MQ messages now use the short form of the name.

In MQSeries for OS/390 Version 5.2, CF structure objects were implicitly created and deleted. When the first queue naming a CF structure was defined, a CF structure object with that name was implicitly created. Similarly, when the last queue naming a CF structure was deleted, the CF structure object was deleted. These CF structure operations happened invisibly to the user. Such CF structure objects have CFLEVEL(1).

CF structure objects defined with CFLEVEL(2) (on a Version 5.3.1 queue manager) are for compatibility between Version 5.2 and Version 5.3.1 queue managers. They can be used by Version 5.2 queue managers, and they can be used and manipulated by Version 5.3.1 queue managers.

CF structure objects defined with CFLEVEL(3) are only usable by Version 5.3.1 queue managers. New function is supported for queues defined on a CFLEVEL(3) CF structure.

Once all queue managers in the queue-sharing group are at Version 5.3.1 level, you can migrate a CF structure from CFLEVEL(2) to CFLEVEL(3) using the DEFINE REPLACE or ALTER CFSTRUCT commands.

The RECOVER attribute of a CF structure specifies whether messages in the structure can be recovered if the structure fails.

There are very strict conditions that apply before a CF structure can be altered. For these conditions and all the command details see the WebSphere MQ Script (MQSC) Command Reference.

*CFLEVEL(3) functions:*

Queues defined on a CFLEVEL(3) CF structure can have the new INDXTYPE(GROUPID) attribute.

Persistent messages can be stored on a queue defined on a CF structure with CFLEVEL(3) and the RECOVER(YES) attribute. The MQSC commands BACKUP CFSTRUCT and RECOVER CFSTRUCT are provided to support recovery. Ensure that regular backups of such structures are taken.

To use the RECOVER CFSTRUCT command, all the active and archive log data sets, and bootstrap data sets of each queue manager in a queue-sharing group must be accessible to each other. You must ensure that your security setup allows this, and use Access Method Services to change the SHAREOPTIONS of these data sets to be (2 3), for example:

```
ALTER '++hlq++.logcopy.ds01.data' SHAREOPTIONS (2 3)
```

For CF structure objects with CFLEVEL(3) only, in the unlikely event of the failure of a coupling facility structure, a Version 5.3.1 queue manager remains active. Any in-flight units of work are backed out, but units of work that have progressed beyond commit are disconnected from the applications (which can terminate normally) and are completed when the CF structure becomes available. However, MQSeries for OS/390 Version 5.2 queue managers will terminate abnormally.

*Recommendations:*

It is recommended that as soon as a Version 5.3.1 queue manager is started in a mixed version queue-sharing group, you alter your CF structure objects to be CFLEVEL(2). This stops them being implicitly deleted by a Version 5.2 queue manager in the queue-sharing group.

After all queue managers in the queue-sharing group have been migrated to Version 5.3.1, alter all CF structure objects to have CFLEVEL(3). This gives greater resilience in the unlikely event of a Coupling Facility structure failure, as Version 5.3.1 queue managers can tolerate the failure of a CFLEVEL(3) CF structure.

### **Context profiles:**

This might affect Task 10 in Chapter 1, "Customizing your queue managers" of the *WebSphere MQ for z/OS System Setup Guide*.

Context profiles can now be defined for each individual queue. If you use context security, and you want to use fully qualified context profiles, you need to define a fully qualified context profile for each queue, using the form

```
hlq.CONTEXT.queueName
```

and a generic context profile using the form

```
hlq.CONTEXT.**
```

to cover any queues that do not have fully qualified context profiles, in addition to your existing hlq.CONTEXT profile. This new generic profile should have the same characteristics and accesses granted as your existing hlq.CONTEXT profile. The old hlq.CONTEXT profile should be kept until you have completed migration.

If you use context security and you do not want to use fully qualified context profiles, you need to define a generic profile using the form  
hlq.CONTEXT.\*\*

for all queues belonging to the specified queue manager or queue-sharing group. This new generic profile should have the same characteristics and accesses granted as your existing hlq.CONTEXT profile. The old hlq.CONTEXT profile should be kept until you have completed migration.

**If you leave your existing profiles unchanged, you will get security failure messages.**

**Channel initiator:**

This might affect Task 12 in Chapter 1, "Customizing your queue managers" of the *WebSphere MQ for z/OS System Setup Guide*.

The queue SYSTEM.CHANNEL.REPLY.INFO is no longer used.

A process definition is no longer required if you want channels to start automatically when messages arrive on the transmission queue.

The value returned by the CONNAME attribute of the DISPLAY CHSTATUS command has changed. It is now always the connection name; a new attribute RQMNAME reports the remote queue manager name.

The size of channel objects was increased for Version 5.3 to allow for the added Secure Sockets Layer (SSL) channel attributes. WebSphere MQ automatically updates each of these SSL channel objects the first time that it is changed. This happens whether you are using SSL or not. However, due to the nature of space reclamation in WebSphere MQ, the space used on page set zero might increase dramatically until all these channel objects have been updated.

To avoid this, run a job on your Version 5.3.1 queue manager similar to that in Figure 2 on page 78 that changes all your channel objects, enabling WebSphere MQ to update them all at the same time. This job will display channels and then use the output from this display to define replacement channels. The Channel initiator must be stopped when running this job.

Multiple channel exits are now supported. A channel auto-definition exit program could change the message, send and receive exits associated with a channel, and the user data associated with the exits, by altering the MsgExit, MsgUserData, SendExit, SendUserData, ReceiveExit and ReceiveUserData fields of the MQCD. If you have such an exit, you must change it to alter the fields addressed by MsgExitPtr, MsgUserDataPtr, SendExitPtr, SendUserDataPtr, ReceiveExitPtr and ReceiveUserDataPtr in the MQCD instead. The fields in the MQCD are described in WebSphere MQ Intercommunication.

```

| //STEP1 EXEC PGM=CSQUTIL,PARM='CSQ1'
| //STEPLIB DD DISP=SHR,DSN=th1qua1.SCSQANLE
| // DD DISP=SHR,DSN=th1qua1.SCSQAUTH
| //OUTPUT1 DD DISP=OLD,DSN=MY.COMMANDS(DEFS)
| //SYSPRINT DD SYSOUT=*
| //SYSIN DD *
| COMMAND DDNAME(CMDINP) MAKEDEF(OUTPUT1)
| /*
| //CMDINP DD *
| DISPLAY CHANNEL(*) ALL
| /*
| /** STEP2
| /*******
| /** PERFORM A GLOBAL CHANGE ON THE OUTPUT DATA SET FROM STEP 1, THAT *
| /** IS: MY.COMMANDS(DEFS). CHANGE 'NOREPLACE' TO 'REPLACE' *
| /** THE CHANGED MY.COMMANDS(DEFS) WILL BE THE INPUT FOR STEP3. *
| /*******
| /**
| //STEP3 EXEC PGM=CSQUTIL,PARM='CSQ1'
| //STEPLIB DD DISP=SHR,DSN=th1qua1.SCSQANLE
| // DD DISP=SHR,DSN=th1qua1.SCSQAUTH
| //SYSPRINT DD SYSOUT=*
| //SYSIN DD *
| COMMAND DDNAME(DEFINES)
| /*
| /*******
| /** THE DEFINE COMMAND FOR THE SYSTEM.COMMAND.INPUT QUEUE MIGHT FAIL, *
| /** BUT THIS DOES NOT MATTER. ALTERNATIVELY, REMOVE THE DEFINE FOR THAT *
| /** QUEUE FROM THE OBJECT DATA SET FROM STEP 1. *
| /*******
| /**
| //DEFINES DD DISP=SHR,DSN=MY.COMMANDS(DEFS)

```

Figure 2. Example job for migrating channel objects

Existing channels have the following values set for the new attributes added in Version 5.3:

|                  |          |
|------------------|----------|
| KAINIT           | AUTO     |
| LOCLADDR         | blank    |
| BATCHHB          | zero     |
| SSLPEER          | blank    |
| SSLCIPH          | blank    |
| SSLCAUTH         | REQUIRED |
| Extra exit names | blank    |
| Extra user data  | blank    |

In previous versions of the product, a channel auto-definition exit program could change the message, send and receive exits associated with a channel, and the user data associated with the exits, by altering the MsgExit, MsgUserData, SendExit, SendUserData, ReceiveExit and ReceiveUserData fields of the MQCD. However, now that multiple channel exits are supported, a channel auto-definition exit program must alter the fields addressed by MsgExitPtr, MsgUserDataPtr, SendExitPtr, SendUserDataPtr, ReceiveExitPtr and ReceiveUserDataPtr in the MQCD. The fields in the MQCD are described in WebSphere MQ Intercommunication

### Objects:



This might affect Task 12 in Chapter 1, "Customizing your queue managers" of the *WebSphere MQ for z/OS System Setup Guide*.

Existing objects other than channels have the following values set for new attributes added in Version 5.3:

#### **Namelist**

|        |      |
|--------|------|
| NLTYPE | NONE |
|--------|------|

#### **Queue manager**

|          |          |
|----------|----------|
| SSLKEYR  | NONE     |
| SSLCRLNL | blank    |
| SSLTASKS | blank    |
| CONFIGEV | DISABLED |
| MAXUMSGS | 10 000   |
| EXPRYINT | OFF      |

Changes to the INDXTYPE attribute for queues take effect immediately, if possible, and no longer wait until queue manager restart.

The circumstances under which you can make a change to the INDXTYPE attribute of a queue are more restrictive in Version 5.3.1 than in previous versions. For detailed information about these circumstances, see the WebSphere MQ Script (MQSC) Command Reference.

#### **Libraries:**

In previous versions and releases of this product, we recommended that you include the early code load modules in a library in the link list. This is no longer required, and we now recommend that you do not include early code libraries in the link list. This might affect Task 3 in Chapter 1, "Customizing your queue managers" of the *WebSphere MQ for z/OS System Setup Guide*.

Change any WebSphere MQ library names in all STEPLIBs if they have new names. This might affect Tasks 1, 2, 3, 6, 7, 11, 18 and 19 in Chapter 1, "Customizing your queue managers" of the *WebSphere MQ for z/OS System Setup Guide*.

In consequence of the Secure Sockets Layer (SSL) support, the load library thlqual.SCSQMVR1 must be in PDS-E format, even if you do not use SSL. This might affect Task 7 in Chapter 1, "Customizing your queue managers" of the *WebSphere MQ for z/OS System Setup Guide*.

#### **CICS message table names:**

In SCSQPROC member CSQ4B100, CSQCMTXT has been changed to CSQFCTAB, and in CSQ4D100, CSQKMSG has been changed to CSQFKTAB. However, CICS resource definitions using the old names will continue to work until you are notified otherwise.

#### **DEFINE MAXSMGS command:**

This might affect Task 12 in Chapter 1, "Customizing your queue managers" of the *WebSphere MQ for z/OS System Setup Guide*.

This command is superseded by the MAXUMSGS queue manager attribute, which defines the maximum number of uncommitted messages which can be put and retrieved within a syncpoint. MAXUMSGS can be set by the WebSphere MQ ALTER QMGR command, and its value is retained across restart. The DEFINE MAXSMSGS command is retained for compatibility; it has the same effect as using ALTER QMGR, and so can no longer be issued from the CSQINP1 initialization input data set. DISPLAY MAXSMSGS is also retained for compatibility; it can no longer be issued from the CSQINP1 initialization data set.

For further information about the ALTER QMGR command, see *WebSphere MQ Script (MQSC) Command Reference*.

### **Additional steps when migrating from Version 2.1 to Version 5.3.1**

If you intend to migrate from Version 2.1 to Version 5.3.1, you must consider the following when you customize your new version, in addition to the tasks in the previous sections (you do not need to install and customize the intervening versions):

#### **System parameter module**

This might affect Task 16 in Chapter 1, "Customizing your queue managers" of the *WebSphere MQ for z/OS System Setup Guide*.

There were several system parameters introduced in Version 5.2 (QSGDATA, RESAUDIT, DEALLCT, and UNIT2) and the MAXRTU parameter superseded the MAXALLC parameter, which is no longer used. There were also several changed parameters. Consider whether you need to use these parameters and change your system parameter module accordingly.

If you do not need to use these parameters, you do not need to relinkedit your system parameter module.

#### **Channel initiator parameter module**

This might affect Task 17 in Chapter 1, "Customizing your queue managers" of the *WebSphere MQ for z/OS System Setup Guide*.

There were several channel initiator parameters introduced in Version 5.2 (ADOPTCHK, ADOPTMCA, DNSGROUP, DNSWLM, LUGROUP, OPORTMAX, and OPORTMIN). Consider whether you need to use these parameters and change your channel initiator parameter module accordingly.

If you do not need to use these parameters, you do not need to relinkedit your channel initiator parameter module.

#### **Initialization data sets**

This might affect Task 12 in Chapter 1, "Customizing your queue managers" of the *WebSphere MQ for z/OS System Setup Guide*.

A sample input initialization data set for queue sharing groups called thlqual.SCSQPROC(CSQ4INSS) is supplied with WebSphere MQ. If you are planning to use queue-sharing groups, customize and include this data set. (Queue sharing groups are described in the *WebSphere MQ for z/OS Concepts and Planning Guide*.)

Review sample data set thlqual.SCSQPROC(CSQ4INYG) to see if you want to use the default buffer pool, storage class, and page set definitions.

**Logs** Review sample data set thlqual.SCSQPROC(CSQ4BSDS) to see if you want to use the default settings for log placement and size.

If you are using very large messages, the amount of storage required for your log and archive data sets might increase. This is described in the *WebSphere MQ for z/OS Concepts and Planning Guide*.

#### **Installation verification program**

The name of the samples for the IVP in thlqual.SCSQPROC have been changed to CSQ4IVPQ and CSQ4IVPR. New samples called CSQ4IVPG and CSQ4IVPS have been added for the queue-sharing group IVP. These are described in Chapter 3, "Testing your queue manager" of the *WebSphere MQ for z/OS System Setup Guide*.

#### **Migrating queues and queue definitions to shared queues**

*WebSphere MQ for z/OS System Administration Guide* describes how to migrate your existing queues and queue definitions to be used as shared queues. You do not have to do this, but you should consider it if you are going to use shared queues.

#### **Change log inventory utility (CSQJU003)**

The STARTRBA and ENDRBA keyword value of NEWLOG must end in 000 and FFF respectively.

#### **Return codes**

MQRC\_PAGESET\_FULL and new return code MQRC\_STORAGE\_MEDIUM\_FULL have the same value.

#### **Data conversion exits**

Data conversion exits written for MQSeries for OS/390 Version 2.1 will continue to function correctly with Version 5.3.1. However, they cannot convert messages containing text using the Unicode UCS-2 coded character sets (1200, 13488, 17584) and need to be updated to do so, if you require such conversions.

Exits generated using the CSQUCVX utility need to be reassembled and link-edited, ensuring that you use the thlqual.SCSQMACS library supplied with Version 5.3.1. See the *WebSphere MQ Application Programming Guide* and the CSQ4BAX9 and CSQ4CAX9 samples for information about using CSQUCVX.

Other exits place calls to **MQXCNVC** to perform data conversion. If these exits need to deal with UCS-2, the *Options* parameter of this call must be updated to specify the byte order of the UCS-2 text. See the *WebSphere MQ Application Programming Reference* for information about **MQXCNVC** and sample exit CSQ4BAX8, which demonstrates how to calculate this parameter.

If these exits are not updated, applications cannot convert to or from Unicode UCS-2 CCSIDs. Typically, this is seen in the response from **MQGET**, which returns the message unconverted with reason code MQRC\_SOURCE\_INTEGER\_ENC\_ERROR or MQRC\_TARGET\_INTEGER\_ENC\_ERROR (this depends on the behavior of the exit).

**Note:** Early versions of the CSQ4BAX8 sample exit incorrectly filled in the *Options* parameter of **MQXCNVC**, and exits that have copied this behavior might convert UCS-2 text incorrectly, without reporting a failure.

## Additional steps when migrating from Version 1.2 to Version 5.3.1

If you intend to migrate from Version 1.2 to Version 5.3.1, you must consider the following when you customize your new version, in addition to the tasks in the previous sections (you do not need to install and customize the intervening versions):

### System parameter module

This might affect Tasks 16 and 17 in Chapter 1, "Customizing your queue managers" in the *WebSphere MQ for z/OS System Setup Guide*.

The system parameters EXITLIM, EXITTCB, and WLMTIME, and channel initiator parameters TCPTYPE, LU62ARM, and LSTRTMR, were introduced in Version 2.1. Consider whether you need to use these parameters, and change your parameter modules accordingly.

### Installation process

This might affect Tasks 1, 2, 3, and 11 in Chapter 1, "Customizing your queue managers" in the *WebSphere MQ for z/OS System Setup Guide*.

There are several changes to the installation process, and some additional libraries. The two distributed queuing features for the non-CICS mover have been incorporated into the base product, and the CICS mover has been made an optional feature. The CICS bridge has also been incorporated into the base product.

These are described in the *WebSphere MQ for z/OS Program Directory*.

### Automatic Restart Manager (ARM)

The z/OS Automatic Restart Manager (ARM) is now supported. This support coexists on the same z/OS image with earlier releases that do not support ARM. The queue managers and channel initiators in the earlier releases do not register with ARM and so can not be restarted automatically.

If you do not want to use ARM with your Version 5.3.1 queue managers and channel initiators, specify RESTART\_ATTEMPTS(0) for the WebSphere MQ element in your ARM policy. Note that if you do not specify WebSphere MQ elements in your ARM policy, default ARM policies are used for WebSphere MQ.

z/OS ARM support is described in the *WebSphere MQ for z/OS System Administration Guide*

### Clusters

This might affect Tasks 6 and 12 in Chapter 1, "Customizing your queue managers" in the *WebSphere for z/OS System Setup Guide*.

WebSphere MQ now supports clustering. Before you use clustering you must review all your applications to determine whether each one can operate in a clustering environment. You might have to modify your applications to remove or manage inter-message affinity. Applications that attempt to open nonexistent queues might experience delays, or might even successfully open a queue somewhere in the cluster.

You also need to create the system objects required for clustering. These are described in the *WebSphere MQ for z/OS Concepts and Planning Guide*.

There is a cluster workload user exit; if you use this you need to add a CSQXLIB DD statement to your queue manager started task procedure, xxxxMSTR, and ensure that you have access to the LE runtime library SCEERUN.

Cluster support is described in the WebSphere MQ Queue Manager Clusters manual.

### Storage classes

This might affect Task 12 in Chapter 1, "Customizing your queue managers" in the *WebSphere MQ for z/OS System Setup Guide*.

The supplied default for storage class SYSTEM (which was used by many of the SYSTEM queues) has been changed to page set 01, so that messages are not put on page set 00.

If you currently use the defaults supplied, this change will probably have no effect, even if you use the DEFINE REPLACE option for your storage class definitions in your initialization input data set. This is because some of the queues using that storage class (like the SYSTEM.CHANNEL.SYNCQ for example) have messages on them permanently. If you want to move the queues to another page set, follow the procedure given in the *WebSphere MQ for z/OS System Administration Guide*.

### Initialization data sets

This might affect Task 12 in Chapter 1, "Customizing your queue managers" in the *WebSphere MQ for z/OS System Setup Guide*.

The sample input initialization data sets supplied with WebSphere MQ have been reorganized and renamed.

### Resource Recovery Services (RRS)

This might affect Task 18 in Chapter 1, "Customizing your queue managers" in the *WebSphere MQ for z/OS System Setup Guide*.

You can migrate your existing batch/TSO WebSphere MQ applications to exploit RRS coordination with little or no application program change. If you link-edit your WebSphere MQ application with the CSQBRRSI adapter, MQCMIT and MQBACK synchronize your unit of work across WebSphere MQ and all other RRS-enabled resource managers. If you link-edit your WebSphere MQ application with the CSQBRSTB adapter, you must change MQCMIT and MQBACK to SRRCMIT and SRRBACK.

Version 5.3.1 continues to support the non-RRS managed batch adapter in addition to supporting the RRS managed adapter. Thus different versions of WebSphere MQ queue managers can coexist on the same z/OS image.

### OpenEdition sockets

This might affect Task 17 in Customizing your queue managers.

OpenEdition sockets are now available for use as an alternative to IUCV. If you are using OS/390 Version 2.5 or later, and are using IBM TCP/IP for distributed queuing, IUCV is not available. You must set the TCPTYPE channel initiator parameter to OESOCKETS. Using OpenEdition sockets, you do not need to restart the channel initiator if TCP/IP has to be restarted.

### Channel initiator security

This might affect Task 10 in Chapter 1, "Customizing your queue managers" in the *WebSphere MQ for z/OS System Setup Guide*.

Channel initiator user ID checking has been changed and some facilities added. See *WebSphere MQ for z/OS System Setup Guide* for details, and review your channel definitions to ensure that you are getting the security control you want.

#### **Channel initiator snap dumps**

This might affect Task 7 in Chapter 1, "Customizing your queue managers" in the *WebSphere MQ for z/OS System Setup Guide*.

The channel initiator can now record error information in a data set instead of creating a dump. Add the CSQSNAP DD statement to your channel initiator started task procedure to support this.

#### **IMS language interface module**

The IMS language interface module CSQ2LI00 is no longer supported. All IMS applications should use the IMS supplied DFSLI000 module.

#### **Euro currency symbol**

Support for the euro currency symbol has been added to WebSphere MQ. If you need to modify your applications to use this symbol, ensure that they use one of the coded character sets that include it. These are described in the *WebSphere MQ Application Programming Reference* manual. If you need to change the coded character set used by your queue manager, use the CCSID parameter of the system parameter module. This is described in *WebSphere MQ for z/OS System Setup Guide*.

#### **Queue object size**

The size of queue objects was increased for Version 5.2 to allow for the added cluster attributes. WebSphere MQ automatically updates each of these queue objects the first time that it is changed. This happens whether you are using clustering or not. However, due to the nature of space reclamation in WebSphere MQ, the space used on page set zero might increase dramatically until all these queue objects have been updated.

To avoid this, run a job similar to that in Figure 3 on page 85 that changes all your queue objects, enabling WebSphere MQ to update them all at the same time. Ensure that no applications have any queues open when you run this job.

If you do not run a job like this, applications attempting to open queues might receive return code MQRC\_OBJECT\_IN\_USE. This includes attempts by the channel initiator to open transmission queues.

```

| //STEP1 EXEC PGM=CSQUTIL,PARM='CSQ1'
| //STEPLIB DD DISP=SHR,DSN=thlqua1.SCSQANLE
| // DD DISP=SHR,DSN=thlqua1.SCSQAUTH
| //OUTPUT1 DD DISP=OLD,DSN=MY.COMMANDS(DEFS)
| //SYSPRINT DD SYSOUT=*
| //SYSIN DD *
| COMMAND DDNAME(CMDINP) MAKEDEF(OUTPUT1)
| /*
| //CMDINP DD *
| DISPLAY QUEUE(*) TYPE(QLOCAL) ALL
| DISPLAY QUEUE(*) TYPE(QMODEL) ALL
| DISPLAY QUEUE(*) TYPE(QALIAS) ALL
| DISPLAY QUEUE(*) TYPE(QREMOTE) ALL
| /*
| /** STEP2
| /******
| /* PERFORM A GLOBAL CHANGE ON THE OUTPUT DATA SET FROM STEP 1, THAT *
| /* IS: MY.COMMANDS(DEFS). CHANGE 'NOREPLACE' TO 'REPLACE' *
| /* THE CHANGED MY.COMMANDS(DEFS) WILL BE THE INPUT FOR STEP3. *
| /******
| /*
| //STEP3 EXEC PGM=CSQUTIL,PARM='CSQ1'
| //STEPLIB DD DISP=SHR,DSN=thlqua1.SCSQANLE
| // DD DISP=SHR,DSN=thlqua1.SCSQAUTH
| //SYSPRINT DD SYSOUT=*
| //SYSIN DD *
| COMMAND DDNAME(DEFINES)
| /*
| /******
| /* THE DEFINE COMMAND FOR THE SYSTEM.COMMAND.INPUT QUEUE MIGHT FAIL, *
| /* BUT THIS DOES NOT MATTER. ALTERNATIVELY, REMOVE THE DEFINE FOR THAT *
| /* QUEUE FROM THE OBJECT DATA SET FROM STEP 1. *
| /******
| /*
| //DEFINES DD DISP=SHR,DSN=MY.COMMANDS(DEFS)

```

Figure 3. Example job for migrating queue objects

## Chapter 6. WebSphere MQ classes for JMS and classes for Java considerations

There are several changes to implementation of WebSphere MQ classes for JMS and classes for Java that you might need to consider when migrating to WebSphere MQ Version 7.0.

### Upgrading from previous versions of WebSphere MQ classes for JMS

Compared to previous versions of WebSphere MQ classes for JMS, most error codes and exception messages have changed in Version 7. The reason for these changes is that WebSphere MQ classes for JMS now has a layered architecture and exceptions are thrown from different layers in the code.

For example, if an application tries to connect to a queue manager that does not exist, a previous version of WebSphere MQ classes for JMS threw a JMSException exception with the following information:

```
MQJMS2005: Failed to create MQQueueManager for 'localhost:QM_test'.
```

This exception contained a linked MQException exception with the following information:

```
MQJE001: Completion Code 2, Reason 2058
```

By comparison in the same circumstances, Version 7 of WebSphere MQ classes for JMS throws a JMSEException exception with the following information:

```
Message : JMSWMQ0018: Failed to connect to queue manager 'QM_test' with
 connection mode 'Client' and host name 'localhost'.
Class : class com.ibm.msg.client.jms.DetailedJMSEException
Error Code : JMSWMQ0018
Explanation : null
User Action : Check the queue manager is started and if running in client mode,
 check there is a listener running. Please see the linked exception
 for more information.
```

This exception contains a linked MQException exception with the following information:

```
Message : JMSCMQ0001: WebSphere MQ call failed with compcode '2' ('MQCC_FAILED')
 reason '2058' ('MQRC_Q_MGR_NAME_ERROR').
Class : class com.ibm.mq.MQException
Completion Code : 2
Reason Code : 2058
```

If your application parses or tests exception messages returned by the `Throwable.getMessage()` method, or error codes returned by the `JMSEException.getErrorCode()` method, your application probably needs to be modified in order to use Version 7 of WebSphere MQ classes for JMS.

## Tracing and error reporting in WebSphere MQ classes for JMS

WebSphere MQ classes for JMS now contains a class that an application can use to control tracing. An application can start and stop tracing, specify the required level of detail in a trace, and customize trace output in various ways. For example, you can now configure trace output by using properties that you specify in a client configuration file. For more information, see *WebSphere MQ Using Java*.

## Channel exits

WebSphere MQ Version 7.0 provides new channel exit interfaces that offer improved functionality and performance.

When called from WebSphere MQ classes for JMS, channel exit programs written in C or C++ now behave in the same way as when they are called from a Websphere MQ client. For information, see *WebSphere MQ Using Java*.

You can now write channel exit classes using a new set of interfaces in the `com.ibm.mq.exits` package instead of using the interfaces in WebSphere MQ classes for Java. You can use the new interfaces to define classes to use as channel exits. The old channel exit interfaces are also still supported, but the new interfaces offer improved functionality and performance. For more information, see *WebSphere MQ Using Java*.

## Relationship between WebSphere MQ classes for JMS and WebSphere MQ classes for Java

In WebSphere MQ Version 7.0, the implementation of WebSphere MQ classes for JMS is no longer dependent on WebSphere MQ classes for Java. WebSphere MQ



classes for Java and WebSphere MQ classes for JMS are now peers that use a common Java interface to the MQI. For more information, see *WebSphere MQ Using Java*

## Receiving RFH2 Headers in WebSphere MQ classes for Java

In WebSphere MQ Version 6.0, using all defaults, a got MQMessage includes the RFH2 header (if present) and the message payload (for example, Hello World). If, for example, the message was sent from JMS, this RFH2 would be the standard RFH2 that is generated for a JMS Message. It is possible, though, for the RFH2 to have been generated by other means.

In WebSphere MQ Version 7.0, using all defaults, a got MQMessage does not include the RFH2 header (if present) for folders of the RFH2 that are recognised as being from JMS. If a JMS message is got it consists solely of the message payload. The RFH2 is represented in the message properties of the WebSphere MQ classes for Java message.

To preserve the WebSphere MQ Version 6.0 behavior of propagating the RFH2 to the caller, the queue attribute PROPCTL must be changed from COMPAT to FORCE.

## Publish/subscribe

For WebSphere MQ classes for JMS information that is specific to publish/subscribe, see Using publish/subscribe with WebSphere MQ classes for JMS.

## MQC is replaced by MQConstants

A new package, com.ibm.mq.constants, is supplied with WebSphere MQ Version 7.0. This package contains the class MQConstants, which implements a number of interfaces. MQConstants contains definitions of all the constants that were in the MQC interface and a number of new constants. The interfaces in this package closely follow the names of the constants header files used in Websphere MQ.

For example, the interface CMQC contains a constant MQOO\_INPUT\_SHARED; this corresponds to the header file cmqc.h and the constant MQOO\_INPUT\_SHARED.

com.ibm.mq.constants can be used with both WebSphere MQ classes for Java and WebSphere MQ classes for JMS.

MQC is still present, and has the constants it previously had; however, for any new applications, you should use the com.ibm.mq.constants package.

## Java and secret key reset

Secret key reset, controlled by SSL property SSLRECOUNT, does not work with the following Java versions and platforms:

- Java 5.0 and Java 6.0
- Java 1.4.2 in FIPS mode
- HP or Sun Java 2 Software Development Kit or Java Runtime Environment

| Java 5.0 is included with WebSphere MQ Version 7.0. If you are using secret key  
| reset under WebSphere MQ Version 6 with Java 1.4.2 and you want it to continue  
| working, you must remain at Java 1.4.2. This applies to WebSphere MQ classes for  
| Java and WebSphere MQ classes for JMS.

---

## Chapter 7. Migration to publish/subscribe on WebSphere MQ V7.0

Publish/subscribe function in WebSphere MQ Version 7.0 is performed by the queue manager, rather than by a separate publish/subscribe broker. When you migrate your systems to WebSphere MQ Version 7.0, publish/subscribe function is not automatically migrated. You must upgrade publish/subscribe information to WebSphere MQ Version 7.0 separately.

In WebSphere MQ V6, applications perform publish and subscribe operations by placing special request messages on certain queues. The WebSphere MQ V6 Publish/Subscribe Broker then reads and acts on these messages (for example by publishing messages to subscribing applications). State information such as who is subscribing to which publications is owned and maintained by the publish/subscribe broker. This broker is started and stopped independently from the queue manager.

In WebSphere MQ V7, newly written publish/subscribe applications do not communicate with the broker in order to publish or subscribe; they use the new API directly. The verb MQPUT is used to publish messages to a topic and MQSUB is used to subscribe. The queue manager itself performs the publish/subscribe function, so no separate publish/subscribe broker is required.

When you upgrade a queue manager from WebSphere MQ V6 to WebSphere MQ V7, the publish/subscribe broker is not upgraded. State information must be migrated from the WebSphere MQ publish/subscribe broker into the queue manager. Data that is migrated includes subscriptions, retained publications, hierarchy relations, and authorities. You migrate a queue manager by using the `strmqbrk` command, which previously started the publish/subscribe broker.

WebSphere MQ V6 publish/subscribe brokers could be connected into hierarchies so that publications and subscriptions could flow between them. After migrating (using `strmqbrk`) these hierarchies continue to function in WebSphere MQ V7. WebSphere MQ V7 also contains a new method of allowing publications and subscriptions flow between queue managers; publish/subscribe clusters. An advantage of using publish/subscribe clusters mean that no queue manager provides a single point of failure to the flow of publications or subscriptions. To migrate to a publish/subscribe cluster, first migrate to a WebSphere MQ V7 hierarchy using `strmqbrk` and then convert it to a cluster by creating cluster topics and altering parent/child relations.

---

### **strmqbrk (Migrate WebSphere MQ Version 6.0 broker to Version 7.0)**

Migrate the persistent state of a Websphere MQ publish/subscribe broker.

#### **Purpose**

Use the `strmqbrk` command to migrate WebSphere MQ Version 6.0 publish/subscribe broker's state to WebSphere MQ Version 7.0 publish/subscribe. If the queue manager has already been migrated, no action is taken.

In WebSphere MQ Version 6.0, **strmqbrk** started a broker. The WebSphere MQ Version 7.0 publish/subscribe engine cannot be started in this manner. To enable publish/subscribe for a queue manager, use the ALTER QMGR command; for details, see ALTER QMGR in the *WebSphere MQ Script (MQSC) Command Reference*.

You can also use the runmqbrk command. This has the same parameters as strmqbrk and exactly the same effect.

## Syntax

AIX, HP-UX, Linux, Solaris, and Windows

```

▶▶ strmqbrk [-p ParentQMgrName] [-m QMgrName] [-f]
 [-l LogFileName]

```

## Optional parameters

AIX, HP-UX, Linux, Solaris, and Windows

**-p** *ParentQMgrName*

**Note:** This option is deprecated.

If you specify the current parent queue manager, a warning message is issued and migration continues. If you specify a different queue manager, a warning is issued and migration is not performed.

**-m** *QMgrName*

The name of the queue manager to be migrated. If you do not specify this parameter, the command is routed to the default queue manager.

**-f** Force migration. This option specifies that objects created during the migration replace existing objects with the same name. If this option is not specified, if migration would create a duplicate object, a warning is issued, the object is not created, and migration continues.

**-l** *LogFileName*

Log migration activity to the file specified in *LogFileName*.

## Syntax

i5/OS

```

▶▶ STRMQBRK [-PARENTMQM(ParentQMgrName)] [-MQMNAME QMgrName]

```

## Optional parameters

AIX, HP-UX, Linux, Solaris, and Windows

**-PARENTMQM**(*ParentQMgrName*)

**Note:** This option is deprecated.

If you specify the current parent queue manager, a warning message is issued and migration continues. If you specify a different queue manger, a warning is issued and migration is not performed.

**-MQMNAME** *QMgrName*

The name of the queue manager to be migrated. If you do not specify this parameter, the command is routed to the default queue manager.

---

## Application migration

The WebSphere MQ Version 6.0 publish/subscribe command message interface is being deprecated. If you have applications that use this interface directly, you should migrate those applications to use the new Version 7.0 publish/subscribe functions.

The following sections explain how to replace existing command messages.

### Identity

In WebSphere MQ Version 6 there were two ways of identifying a subscriber. These were referred to as the traditional identity and the subscription name.

The traditional identity was also used to identify a publisher. The traditional identity was a combination of queue name, queue manager name, and optional correlation identifier.

A publisher no longer has an explicit publisher identity, but can be identified in the same way as any other WebSphere MQ application, by means of its connection to the queue manager. Since there is no explicit registration of a publisher, or his identity over and above what can be obtained by displaying the connections to the queue manager, there is no longer a need for the anonymous option on Register Publisher. Your application must now use the SubName field in the MQSD to identify a subscriber.

The correlation identifier also had a secondary use which was to allow subscribers to MQGET by CorrelId to only get publications for a particular subscription, if there were multiple subscriptions all using the same queue. This is provided by using the SubCorrelId field returned in the MQSD at MQSUB time.

### Stream Name

MQPS\_STREAM\_NAME is deprecated since stream names are part of the full topic name. Stream names can be mapped to administrative topic objects, and then the topic name used along with the stream name can be mapped to a topic string to be concatenated with the topic string from the topic object. For example, if the application was previously using a stream queue name of SYSTEM.BROKER.RESULTS.STREAM and a topic of Sport/Soccer/State/LatestScore/\*, then a topic object can be created whose name is SYSTEM.BROKER.RESULTS.STREAM which is defined to have a TOPICSTR of / and the new application will provide a two part topic name in the MQOD or MQSD using an ObjectName of SYSTEM.BROKER.RESULTS.STREAM and an ObjectString of Sport/Soccer/State/LatestScore/.\*

If an administrative topic object that does not exist is used in place of a stream name, the error (effectively mapping to MQRCCF\_STREAM\_ERROR) which is

given is MQRC\_UNKNOWN\_OBJECT\_NAME. supported.

## Application migration details

When migrating to use the MQ API to do publish/subscribe, the code within any one application program must be consistent.

The application program must not contain a mixture of these deprecated APIs and the new MQ API options. An entire application suite, such as the combination of a subscribing application program and a publishing application program, does not all need to be migrated at the same time. Interaction between a publishing application program using the deprecated APIs and a subscribing application using the new MQ API is supported.

## Delete Publication - Version 7 replacement

The Delete Publication command message contains a number of parameters. This should be replaced by using the PCF ClearTopic command. This section details the equivalent options or fields in the PCF command message to show how an application would migrate from using the Delete Publication command message to using the PCF ClearTopic command message.

### Required parameters

MQPS\_COMMAND with value MQPS\_DELETE\_PUBLICATION is implied when you use the ClearTopic command.

MQPS\_TOPIC is provided in a field in the ClearTopic command message. If your application provided more than one MQPS\_TOPIC in a single Delete Publication command message, it must now issue a separate ClearTopic call for each separate topic string.

### Optional parameters

MQPS\_DELETE\_OPTIONS is replaced with an attribute of the ClearTopic command message.

For MQPS\_STREAM\_NAME see *WebSphere MQ Publish/Subscribe User's Guide*.

### Error codes

If your application checked for any of the following error codes, the equivalent MQRC error codes are shown in the following table:

| Reason codes in NameValueString of the broker response message.                                                                                                          | MQRC equivalent          |
|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------|
| MQRCCF_STREAM_ERROR                                                                                                                                                      | MQRC_UNKNOWN_OBJECT_NAME |
| MQRCCF_TOPIC_ERROR                                                                                                                                                       | MQRC_OBJECT_STRING_ERROR |
| MQRCCF_INCORRECT_STREAM                                                                                                                                                  | See Note 1               |
| <b>Notes:</b>                                                                                                                                                            |                          |
| 1. No equivalent since there is no need to provide the stream name twice, once in the command and once by putting it to the stream queue, so you cannot have a mismatch. |                          |

## Deregister publisher - Version 7 replacement

The Deregister Publisher command message contains a number of parameters. You should replace it with the MQCLOSE verb. This section details the equivalent options or fields in the MQ API to show how to migrate an application from the Deregister Publisher command message to MQCLOSE.

A difference in behaviour will be seen because a Register Publisher command could leave an application registered even when it was not connected, whereas the equivalent MQOPEN will only show a publisher's intent when the application is connected and keeps the handle from MQOPEN available. Even without issuing MQCLOSE, an application will be deregistered when the queue manager detects that the application's connection is lost.

### Required parameters

MQPS\_COMMAND with value MQPS\_REGISTER\_PUBLISHER is implied when closing a handle to a topic previously opened using MQOPEN with the MQOO\_OUTPUT option.

### Optional parameters

If your application provided a queue and queue manager name (either by using MQPS\_Q\_MGR\_NAME and MQPS\_Q\_NAME in the command message, or from the ReplyToQ and ReplyToQMgr fields in MQMD of the command message) these attributes are now implied by the provision of the handle obtained when opening the topic.

MQPS\_REGISTRATION\_OPTIONS is replaced with options on the MQCLOSE call. See MQCLOSE for more details. Note that there are two ways you could have specified each of these options in your application, a string constant, MQPS\_\* or an integer constant, MQREGO\_\*. Both are replaced by the use of a single numeric constant.

| String constant                                                                                                                                                                                                                                              | Integer constant             | MQCLOSE Options field constant           |
|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------|------------------------------------------|
| MQPS_CORREL_ID_AS_IDENTITY                                                                                                                                                                                                                                   | MQREGO_CORREL_ID_AS_IDENTITY | See Version 6 (queued) publish/subscribe |
| MQPS_DEREGISTER_ALL                                                                                                                                                                                                                                          | MQREGO_DEREGISTER_ALL        | See Note 1                               |
| <b>Notes:</b>                                                                                                                                                                                                                                                |                              |                                          |
| 1. Since only one topic can be opened by the MQOPEN call, closing the handle closes that one topic. There is no need for an equivalent option. If many topics are opened, simply issuing MQDISC will close them all, saving the need to MQCLOSE each handle. |                              |                                          |

For MQPS\_STREAM\_NAME see Version 6 (queued) publish/subscribe, although in this case, the stream name is implied by the provision of the handle obtained when opening the topic. MQPS\_TOPIC is implied by the provision of the handle obtained when opening the topic.

### Error codes

If your application checked for any of the following error codes, the equivalent MQRC error codes are shown in the following table:

| Reason codes in NameValueString of the broker response message.                                                                                                                                       | MQRC equivalent              |
|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------|
| MQRCCF_STREAM_ERROR                                                                                                                                                                                   | MQRC_HOBJ_ERROR (See note 1) |
| MQRCCF_TOPIC_ERROR                                                                                                                                                                                    | MQRC_HOBJ_ERROR (See note 1) |
| MQRCCF_NOT_REGISTERED                                                                                                                                                                                 | MQRC_HOBJ_ERROR (See note 1) |
| MQRCCF_Q_MGR_NAME_ERROR                                                                                                                                                                               | MQRC_HOBJ_ERROR (See note 1) |
| MQRCCF_Q_NAME_ERROR                                                                                                                                                                                   | MQRC_HOBJ_ERROR (See note 1) |
| MQRCCF_DUPLICATE_IDENTITY                                                                                                                                                                             | MQRC_HOBJ_ERROR (See note 1) |
| MQRCCF_UNKNOWN_STREAM                                                                                                                                                                                 | MQRC_HOBJ_ERROR (See note 1) |
| MQRCCF_REG_OPTIONS_ERROR                                                                                                                                                                              | MQRC_OPTIONS_ERROR           |
| <b>Notes:</b>                                                                                                                                                                                         |                              |
| 1. This error code implies the same type of problem, but since all of these fields are now implied by the provision of the handle obtained when opening the topic, this is the only equivalent error. |                              |

## Deregister subscriber - Version 7 replacement

The Deregister Subscriber command message contains a number of parameters. This should be replaced by using the MQCLOSE verb. This section details the equivalent options or fields in the MQ API to show how an application would migrate from using the Deregister Subscriber command message to using MQCLOSE. If the Deregister Subscriber command message was used in a different program from that of the Register Subscriber command message, the application must now first use the MQSUB call with the MQSO\_RESUME option to get a handle to the subscription, in order to deregister it.

### Required parameters

MQPS\_COMMAND with value MQPS\_DEREGISTER\_SUBSCRIBER is replaced by the use of the MQCLOSE verb with the option MQCO\_REMOVE\_SUB.

### Optional parameters

If your application provided a queue and queue manager name (either by using MQPS\_Q\_MGR\_NAME and MQPS\_Q\_NAME in the command message, or from the ReplyToQ and ReplyToQMgr fields in MQMD of the command message) these attributes are now implied by the provision of the handle obtained when subscribing to the topic.

MQPS\_REGISTRATION\_OPTIONS is replaced with Options on the MQCLOSE call. See MQCLOSE for more details. Note that there are two ways you could have specified each of these options in your application, a string constant, MQPS\_\* or an integer constant, MQREGO\_\*. Both are replaced by the use of a single numeric constant.

| String constant            | Integer constant             | MQCLOSE Options field constant |
|----------------------------|------------------------------|--------------------------------|
| MQPS_CORREL_ID_AS_IDENTITY | MQREGO_CORREL_ID_AS_IDENTITY | See Note 1                     |
| MQPS_DEREGISTER_ALL        | MQREGO_DEREGISTER_ALL        | See Note 2                     |



| String constant                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    | Integer constant        | MQCLOSE Options field constant |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------|--------------------------------|
| MQPS_FULL_RESPONSE                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 | MQREGO_FULL_RESPONSE    | See Note 3                     |
| MQPS_LEAVE_ONLY                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    | MQREGO_LEAVE_ONLY       | See Note 4                     |
| MQPS_VARIABLE_USER_ID                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              | MQREGO_VARIABLE_USER_ID | See Note 1                     |
| <b>Notes:</b> <ol style="list-style-type: none"> <li>1. This option is implied by the provision of the handle obtained when subscribing to the topic.</li> <li>2. Since only one topic (separate topic string that is – of course wildcards can still be used within one topic string) can be subscribed to by the MQSUB call, closing the handle closes that one topic. There is no need for an equivalent option. If many topics are opened, simply issuing MQDISC will close them all, saving the need to MQCLOSE each handle.</li> <li>3. Use of this option is implied in the use of the MQSUB verb. The fields returned in the response message are now populated in the MQSD structure. See MQSUB for more details. Because an MQSUB call must be made in order to obtain the handle to pass to the MQCLOSE call, this option is deprecated.</li> <li>4. Use of these options are deprecated and moved to spiCONN for the one environment where they are needed.</li> </ol> |                         |                                |

For MQPS\_STREAM\_NAME see Version 6 (queued) publish/subscribe, although in this case, the stream name is implied by the provision of the handle obtained when subscribing to the topic. MQPS\_SUBSCRIPTION\_IDENTITY is replaced by a field in spiCONN for the one environment where it is needed.

MQPS\_SUBSCRIPTION\_NAME is replaced by the field in the MQSD called SubName and is therefore implied by the provision of the handle obtained when subscribing to the topic. MQPS\_TOPIC is provided in a field in the MQSD called ObjectString, and is therefore implied by the provision of the handle obtained when subscribing to the topic.

See MQSD for more details

### Error codes

If your application checked for any of the following error codes, the equivalent MQRC error codes are shown in the following table:

| Reason codes in NameValueString of the broker response message. | MQRC equivalent              |
|-----------------------------------------------------------------|------------------------------|
| MQRCCF_STREAM_ERROR                                             | MQRC_HOBJ_ERROR (See note 1) |
| MQRCCF_TOPIC_ERROR                                              | MQRC_HOBJ_ERROR (See note 1) |
| MQRCCF_NOT_REGISTERED                                           | MQRC_HOBJ_ERROR (See note 1) |
| MQRCCF_Q_MGR_NAME_ERROR                                         | MQRC_HOBJ_ERROR (See note 1) |
| MQRCCF_Q_NAME_ERROR                                             | MQRC_HOBJ_ERROR (See note 1) |
| MQRCCF_DUPLICATE_IDENTITY                                       | MQRC_HOBJ_ERROR (See note 1) |
| MQRCCF_UNKNOWN_STREAM                                           | MQRC_HOBJ_ERROR (See note 1) |
| MQRCCF_REG_OPTIONS_ERROR                                        | MQRC_OPTIONS_ERROR           |

|                                                                                                                                                                                                         |                        |
|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------|
| <b>Reason codes in NameValueString of the broker response message.</b>                                                                                                                                  | <b>MQRC equivalent</b> |
| <b>Notes:</b>                                                                                                                                                                                           |                        |
| 1. This error code implies the same type of problem, but because all of these fields are now implied by the provision of the handle obtained when opening the topic, this is the only equivalent error. |                        |

## Publish - Version 7 replacement

The Publish command message contains a number of parameters. This should be replaced by using the MQPUT/MQPUT1 verbs. This section details the equivalent options or fields in the MQ API to show how an application would migrate from using the Publish command message to using MQPUT/MQPUT1.

### Required parameters

MQPS\_COMMAND with value MQPS\_PUBLISH is implied when putting a message to an object handle opening a topic for MQOO\_OUTPUT. If your application did not use Register Publisher, see the details in Publish for remaining unregistered.

MQPS\_TOPIC is provided in a field in the MQOD called ObjectString. See MQOD - Object Descriptor for more details. If your application provided more than one MQPS\_TOPIC in a single Register Publisher command message, it must now issue a separate MQOPEN call for each separate topic string.

### Optional parameters

MQPS\_INTEGER\_DATA can be replaced with a message property.

MQPS\_PUBLICATION\_OPTIONS is replaced with the Options field in the MQPMO structure. See MQPMO for more details. Note that there are two ways you could have specified each of these options in your application, a string constant, MQPS\_\* or an integer constant, MQREGO\_\*. Both are replaced by the use of a single numeric constant.

| String constant              | Integer constant               | MQCLOSE Options field constant                                |
|------------------------------|--------------------------------|---------------------------------------------------------------|
| MQPS_CORREL_ID_AS_IDENTITY   | MQREGO_CORREL_ID_AS_IDENTITY   | See Version 6 (queued) publish/subscribe for more information |
| MQPS_IS_RETAINED_PUBLICATION | MQREGO_IS_RETAINED_PUBLICATION | See Note 1                                                    |
| MQPS_NO_REGISTRATION         | MQREGO_NO_REGISTRATION         | See Note 2                                                    |
| MQPS_OTHER_SUBSCRIBERS_ONLY  | MQREGO_OTHER_SUBSCRIBERS_ONLY  | See Note 3                                                    |
| MQPS_RETAIN_PUBLICATION      | MQREGO_RETAIN_PUBLICATION      | MQPMO_RETAIN                                                  |

| String constant                                                                                                                                                                                                                                                                                                                                                                    | Integer constant | MQCLOSE Options field constant |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------|--------------------------------|
| <b>Notes:</b> <ol style="list-style-type: none"> <li>1. A message property will contain this information</li> <li>2. This option is deprecated because publishers are no longer registered</li> <li>3. This option is deprecated. If an application does not want to receive its own publications it should subscribe using the option MQSO_NO_LOCAL on the MQSUB call.</li> </ol> |                  |                                |

MQPS\_Q\_MGR\_NAME is replaced by the ReplyToQMgr in the MQMD of the publication. If the publisher specifies MQPMO\_NO\_DIRECT\_REQUEST the ReplyToQMgr will not contain the publishers queue manager name, otherwise it will.

MQPS\_Q\_NAME is replaced by the ReplyToQ in the MQMD of the publication. If the publisher does not set this, it is not available.

MQPS\_REGISTRATION\_OPTIONS is replaced with Options in the MQPMO. See MQPMO for more details. These are exactly the same as those in the section on Register Publisher below.

MQPS\_SEQUENCE\_NUMBER is replaced with a message property.

For MQPS\_STREAM\_NAME see Version 6 (queued) publish/subscribe.  
MQPS\_STRING\_DATA is replaced with a message property.

## Register publisher - Version 7 replacement

The Register Publisher command message contains a number of parameters. This should be replaced by using the MQOPEN verb. This section details the equivalent options or fields in the MQ API to show how an application would migrate from using the Register Publisher command message to using MQOPEN. A difference in behaviour will be seen because a Register Publisher command could leave an application registered even when it was not connected, whereas MQOPEN will only show a publishers intent when the application is connected and keeps the handle from MQOPEN available.

### Required parameters

MQPS\_COMMAND with value MQPS\_REGISTER\_PUBLISHER is implied when opening a topic for MQOO\_OUTPUT. If your application did not use Register Publisher, see the details in Publish for remaining unregistered.

MQPS\_TOPIC is provided in a field in the MQOD called ObjectString. See MQOD - Object Descriptor for more details. If your application provided more than one MQPS\_TOPIC in a single Register Publisher command message, it must now issue a separate MQOPEN call for each separate topic string.

### Optional parameters

If your application provided a queue and queue manager name (either by using MQPS\_Q\_MGR\_NAME and MQPS\_Q\_NAME in the command message, or from the ReplyToQ and ReplyToQMgr fields in MQMD of the command message) in

order for subscribing applications to be able to directly contact the publisher, then your application must now provide these details on each published message.

MQPS\_REGISTRATION\_OPTIONS is replaced with Options in the MQPMO. See MQPMO for more details. Note that there are two ways you could have specified each of these options in your application, a string constant, MQPS\_\* or an integer constant, MQREGO\_\*. Both are replaced by the use of a single numeric constant.

| String constant                                                                                                                                                                                                                                                                                                                                                                                                                                              | Integer constant             | MQCLOSE Options field constant           |
|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------|------------------------------------------|
| MQPS_ANONYMOUS                                                                                                                                                                                                                                                                                                                                                                                                                                               | MQREGO_ANONYMOUS             | See Version 6 (queued) publish/subscribe |
| MQPS_CORREL_ID_AS_IDENTITY                                                                                                                                                                                                                                                                                                                                                                                                                                   | MQREGO_CORREL_ID_AS_IDENTITY | See Version 6 (queued) publish/subscribe |
| MQPS_DIRECT_REQUEST                                                                                                                                                                                                                                                                                                                                                                                                                                          | MQREGO_DIRECT_REQUEST        | See Note 1                               |
| MQPS_LOCAL                                                                                                                                                                                                                                                                                                                                                                                                                                                   | MQREGO_LOCAL                 | MQPMO_SCOPE_QMGR                         |
| <b>Notes:</b>                                                                                                                                                                                                                                                                                                                                                                                                                                                |                              |                                          |
| <ol style="list-style-type: none"> <li>Use of this option is implied if the ReplyToQ and ReplyToQMgr fields are provided in the MQMD of the message put. If these fields are not provided, the queue manager will still fill in the ReplyToQMgr as the queue manager local to the publisher. To remain completely anonymous and not even provide this information to subscribers, your application should use the MQPMO_NO_DIRECT_REQUEST option.</li> </ol> |                              |                                          |

For MQPS\_STREAM\_NAME see Version 6 (queued) publish/subscribe.

## Register subscriber - Version 7 replacement

The Register Subscriber command message contains a number of parameters. This should be replaced by using the MQSUB verb. This section details the equivalent options or fields in the MQ API to show how an application would migrate from using the Register Subscriber command message to using MQSUB.

### Required parameters

MQPS\_COMMAND with value MQPS\_REGISTER\_SUBSCRIBER is replaced by the use of the MQSUB verb. If your application did not use Register Subscriber then the use of the MQSUB verb is not required for equivalent behaviour.

MQPS\_TOPIC is provided in a field in the MQSD called ObjectString. See MQSD for more details. If your application provided more than one MQPS\_TOPIC in a single Register Subscriber command message, it must now issue a separate MQSUB call for each separate topic string.

### Optional parameters

If your application provided a non-local queue name and/or a queue manager name other than the one connected to (either by using MQPS\_Q\_MGR\_NAME and MQPS\_Q\_NAME in the command message, or from the ReplyToQ and ReplyToQMgr fields in MQMD of the command message) then your application must now provide an object handle, which has been returned by a MQOPEN call for that queue, in the Hobj parameter of the MQSUB verb.

If your application provided the name of a queue local to the queue manager it connected to, it now has the option to request that the queue manager manage where the publications are sent. This can be done by using the MQSO\_MANAGED option in the field in the MQSD called Options.

MQPS\_REGISTRATION\_OPTIONS is replaced with a field in the MQSD called Options. See MQSD for more details. Note that there are two ways you could have specified each of these options in your application, a string constant, MQPS\_\* or an integer constant, MQREGO\_\*. Both are replaced by the use of a single numeric constant.

| String constant              | Integer constant               | MQCLOSE Options field constant     |
|------------------------------|--------------------------------|------------------------------------|
| MQPS_ADD_NAME                | MQREGO_ADD_NAME                | See Note 1                         |
| MQPS_ANONYMOUS               | MQREGO_ANONYMOUS               | See Identity                       |
| MQPS_CORREL_ID_AS_IDENTITY   | MQREGO_CORREL_ID_AS_IDENTITY   | See Identity (also see Note 7)     |
| MQPS_DUPLICATES_OK           | MQREGO_DUPLICATES_OK           | See Note 2                         |
| MQPS_FULL_RESPONSE           | MQREGO_FULL_RESPONSE           | See Note 3                         |
| MQPS_INCLUDE_STREAM_NAME     | MQREGO_INCLUDE_STREAM_NAME     | See Note 4                         |
| MQPS_INFORM_IF_RETAINED      | MQREGO_INFORM_IF_RETAINED      | See Note 5                         |
| MQPS_JOIN_EXCLUSIVE          | MQREGO_JOIN_EXCLUSIVE          | See Note 6                         |
| MQPS_JOIN_SHARED             | MQREGO_JOIN_SHARED             | See Note 6                         |
| MQPS_LOCAL                   | MQREGO_LOCAL                   | MQSO_SCOPE_QMGR                    |
| MQPS_LOCKED                  | MQREGO_LOCKED                  | See Note 6                         |
| MQPS_NEW_PUBLICATIONS_ONLY   | MQREGO_NEW_PUBLICATIONS_ONLY   | MQSO_NEW_PUBLICATIONS_ONLY         |
| MQPS_NO_ALTERATION           | MQREGO_NO_ALTERATION           | MQSO_RESUME                        |
| MQPS_NON_PERSISTENT          | MQREGO_NON_PERSISTENT          | MQSO_NON_PERSISTENT                |
| MQPS_PERSISTENT              | MQREGO_PERSISTENT              | MQSO_PERSISTENT                    |
| MQPS_PERSISTENT_AS_PUBLISH   | MQREGO_PERSISTENT_AS_PUBLISH   | MQSO_PERSISTENT_AS_PUBLISH         |
| MQPS_PERSISTENT_AS_Q         | MQREGO_PERSISTENT_AS_Q         | MQSO_PERSISTENT_AS_QUEUE_DEF       |
| MQPS_PUBLISH_ON_REQUEST_ONLY | MQREGO_PUBLISH_ON_REQUEST_ONLY | MQSO_PUBLICATIONS_ON_REQUEST       |
| MQPS_VARIABLE_USER_ID        | MQREGO_VARIABLE_USER_ID        | MQSO_ANY_USERID, (also see Note 7) |

| String constant                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             | Integer constant | MQCLOSE Options field constant |
|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------|--------------------------------|
| <b>Notes:</b> <ol style="list-style-type: none"> <li>Use of this option is deprecated since the only identity of a subscription is the SubName. See Deprecation.</li> <li>Use of this option is deprecated since the queued interface has been removed.</li> <li>Use of this option is implied in the use of the MQSUB verb. The fields returned in the response message are now populated in the MQSD structure. See MQSD for more details.</li> <li>Use of this option is deprecated since stream names are part of the full topic name.</li> <li>Use of this option is deprecated since the information about whether a publication is a retained publication or not is a message property that is always present.</li> <li>Use of these options are deprecated and moved to spiCONN for the one environment where they are needed.</li> <li>A tick in this column indicates this option is also relevant for Request Update.</li> </ol> |                  |                                |

For MQPS\_STREAM\_NAME see Version 6 (queued) publish/subscribe, although in this case, the stream name is implied by the provision of the handle obtained when subscribing to the topic.

MQPS\_SUBSCRIPTION\_IDENTITY is replaced by a field in spiCONN for the one environment where it is needed.

MQPS\_SUBSCRIPTION\_NAME is replaced by the field in the MQSD called SubName. See MQSD for more details.

MQPS\_SUBSCRIPTION\_USER\_DATA is replaced by the field in the MQSD called SubUserData. See MQSD for more details.

### Error codes

If your application checked for any of the following error codes, the equivalent MQRC error codes are shown in the following table:

| Reason codes in NameValueString of the broker response message. | MQRC equivalent          |
|-----------------------------------------------------------------|--------------------------|
| MQRCCF_STREAM_ERROR                                             |                          |
| MQRCCF_TOPIC_ERROR                                              |                          |
| MQRCCF_Q_MGR_NAME_ERROR                                         |                          |
| MQRCCF_Q_NAME_ERROR                                             |                          |
| MQRCCF_DUPLICATE_IDENTITY                                       | MQRC_IDENTITY_MISMATCH   |
| MQRCCF_CORREL_ID_ERROR                                          |                          |
| MQRCCF_NOT_AUTHORIZED                                           |                          |
| MQRCCF_UNKNOWN_STREAM                                           |                          |
| MQRCCF_REG_OPTIONS_ERROR                                        |                          |
| MQRCCF_DUPLICATE_SUBSCRIPTION                                   | MQRC_SUB_ALREADY_EXISTS  |
| MQRCCF_SUB_NAME_ERROR                                           |                          |
| MQRCCF_SUB_IDENTITY_ERROR                                       | See note 1               |
| MQRCCF_SUBSCRIPTION_IN_USE                                      | MQRC_SUBSCRIPTION_IN_USE |

| Reason codes in NameValueString of the broker response message.                                                                        | MQRC equivalent |
|----------------------------------------------------------------------------------------------------------------------------------------|-----------------|
| MQRCCF_SUBSCRIPTION_LOCKED                                                                                                             | See note 1      |
| MQRCCF_ALREADY_JOINED                                                                                                                  | See note 1      |
|                                                                                                                                        |                 |
| <b>Notes:</b>                                                                                                                          |                 |
| 1. No equivalent since the use of SubIdentity is deprecated since the only identity of a subscription is the SubName. See Deprecation. |                 |

## Request Update - Version 7 replacement

The Request Update command message contains a number of parameters. This should be replaced by using the MQSUBRQ verb. This section details the equivalent options or fields in the MQ API to show how an application would migrate from using the Request Update command message to using MQSUBRQ.

### Required parameters

MQPS\_COMMAND with value MQPS\_REQUEST\_UPDATE is replaced by the use of the MQSUBRQ verb.

MQPS\_TOPIC is implied by the use of the Hsub handle returned from the MQSUB call which is used as a parameter on the MQSUBRQ call.

### Optional parameters

QMgrName, QName and StreamName are used in exactly the same way in Request Update command messages as they are in Register Subscriber command messages.

See “Register subscriber - Version 7 replacement” on page 98 for details of how to migrate the use of these fields.

See “Register subscriber - Version 7 replacement” on page 98 for details of how to migrate your application’s use of MQPS\_REGISTRATION\_OPTIONS in this command message.

For MQPS\_STREAM\_NAME see Version 6 (queued) publish/subscribe.

MQPS\_SUBSCRIPTION\_NAME is implied by the use of the Hsub handle returned from the MQSUB call which is used as a parameter on the MQSUBRQ call.

---

## New queue manager attributes for publish/subscribe

Five attributes, formerly held in the queue manager configuration file, qm.ini, are now replaced by attributes of the queue manager.

In WebSphere MQ Version 6.0, the attributes listed in the following table were held in the Brokers stanza of the qm.ini file (or the registry in Windows). In WebSphere MQ Version 7.0, they are replaced by the queue manager attributes listed, which can be set by the MQSC command ALTER QMGR or the PCF command Change Queue Manager.

Table 8.

| Attribute in qm.ini          | Queue manager attribute<br>(PCF parameter name) | MQSC parameter name |
|------------------------------|-------------------------------------------------|---------------------|
| MaxMsgRetryCount             | PubSubMaxMsgRetryCount                          | PSRTYCNT            |
| DiscardNonPersistentInputMsg | PubSubNPInputMsg                                | PSNPMSG             |
| DLQNonPersistentResponse     | PubSubNPResponse                                | PSNPRES             |
| DiscardNonPersistentResponse | PubSubNPResponse                                | PSNPRES             |
| SyncPointIfPersistent        | PubSubsyncPoint                                 | PSSYNCP             |

## WebSphere MQ publish/subscribe topology migration

This section contains topics that describe various scenarios for migration to WebSphere MQ Version 7.0 publish/subscribe.

### Migrating a WebSphere MQ Version 6.0 publish/subscribe hierarchy to a Version 7.0 publish/subscribe cluster - all queue managers simultaneously

How to migrate an entire existing Websphere MQ Version 6.0 hierarchy, where the parent and child queue managers are on separate computers, to a Websphere Version 7.0 publish/subscribe cluster, migrating all queue managers at the same time.

#### Before you begin

To migrate the hierarchy, perform the following steps:

#### About this task

1. Install WebSphere MQ Version 7.0 on all of the computers that contain queue managers in the hierarchy, to upgrade all queue managers in the hierarchy to WebSphere MQ Version 7.0.
2. Use the **strmqbrk** control command on each queue manager to migrate all publish/subscribe configuration data into WebSphere MQ Version 7.0.
3. Create a new cluster or nominate an existing cluster, which need not be an existing publish/subscribe cluster. You can do this using WebSphere MQ Script commands (MQSC), or any other type of administration command or utility that is available on your platform, such as the WebSphere MQ Explorer. These methods are described in *WebSphere MQ Queue Manager Clusters*.
4. Ensure that each queue manager is in the cluster by using the MQSC command **DISPLAY CLUSQMGR(\*)**, described in *WebSphere MQ Script (MQSC) Command Reference*. If a queue manager that should be in the cluster is not, then add it. For more information, refer to *WebSphere MQ Queue Manager Clusters*.
5. To remove the hierarchical relationship on each child queue manager within the hierarchy, execute the following MQSC command: **ALTER QMGR PARENT(' ')**
6. Before proceeding to the next step, to confirm that all the hierarchical relationships have been cancelled, use the MQSC command **DISPLAY PUBSUB TYPE(ALL)** on each queue manager.
7. On one of the queue managers within the cluster, define one cluster topic by executing the following MQSC command: **ALTER TOPIC(<topic name>)**



PUBSCOPE(ALL) SUBSCOPE(ALL) CLUSTER(<cluster>) Use a high-level topic, but not the root. For information about cluster topic naming, see Cluster topics.

## Example

## What to do next

### Alternative procedure for i5/OS

The following steps show an alternative procedure for WebSphere MQ for i5/OS, using CL commands and panels in place of MQSC commands.

1. Install WebSphere MQ Version 7.0 on all of the computers that contain queue managers in the hierarchy, to upgrade all queue managers in the hierarchy to WebSphere MQ Version 7.0.
2. Use the **strmqbrk** command on each queue manager to migrate all publish/subscribe configuration data into WebSphere MQ Version 7.0.
3. Create a new cluster or nominate an existing cluster, which need not be an existing publish/subscribe cluster. You can do this using WebSphere MQ Script commands (MQSC), or any other type of administration command or utility that is available on your platform, such as the WebSphere MQ Explorer. These methods are described in *WebSphere MQ Queue Manager Clusters*
4. Ensure that each queue manager is in the cluster. If a queue manager that should be in the cluster is not, then add it.
5. Execute **WRKMQMPS PUBSUBNAME(<parent\_queue\_manager>)** to display the hierarchy.
6. On each child queue manager within the hierarchy, use **option 4=Remove** to detach from the parent, followed by **option 34=Work with Pub/Sub** to move down the sub-hierarchy. Repeat options 4 and 34 until no child queue managers are displayed.
7. Repeat step 6 for each child queue manager belonging to **PUBSUBNAME(<parent\_queue\_manager>)** until no child queue managers are displayed.
8. On one of the queue managers within the cluster, define at least one cluster topic by executing the following command: **CHGMQMTOP TOPNAME(<topic name>) PUBSCOPE(\*ALL) SUBSCOPE(\*ALL) CLUSTER(<cluster>) MQMNAME(<queue manager name>)** Use a high-level topic, but not the root. For information about cluster topic naming, see .

## Migrating a WebSphere MQ Version 6.0 publish/subscribe hierarchy to a Version 7.0 publish/subscribe cluster - queue manager by queue manager

How to migrate an existing WebSphere MQ Version 6.0 hierarchy, where the parent and child queue managers are on separate computers, to a WebSphere Version 7.0 publish/subscribe cluster, one queue manager at a time.

### Before you begin

To migrate the hierarchy, perform the following steps:

### About this task

1. Create a new cluster or nominate an existing cluster, which need not be an existing publish/subscribe cluster. You can do this using WebSphere MQ Script commands (MQSC), or any other type of administration command or utility

that is available on your platform, such as the WebSphere MQ Explorer. These methods are described in *WebSphere MQ Queue Manager Clusters*

2. Select the first queue manager to migrate into the publish/subscribe cluster. To cause the least disruption, select a queue manager that is a leaf node.
3. Install WebSphere MQ Version 7.0 on the computer that contains the selected queue manager in the hierarchy, to migrate the queue manager to WebSphere MQ Version 7.0.
4. Use the **strmqbrk** command on this queue manager to migrate all publish/subscribe configuration data into WebSphere MQ Version 7.0.
5. Join this queue manager into the cluster.
6. If this queue manager has a hierarchical relationship to an existing member of the cluster, use the MQSC command ALTER QMGR PARENT(' ') at the child queue manager to cancel the relationship. Before proceeding to the next step, to confirm that the hierarchical relationship has been cancelled, use the MQSC command DISPLAY PUBSUB TYPE(PARENT) at the child queue manager.
7. If this is the first queue manager to be migrated into the publish/subscribe cluster, define at least one cluster topic by executing the MQSC command ALTER TOPIC(<topic name="">) PUBSCOPE(ALL) SUBSCOPE(ALL) CLUSTER(<cluster>)

**Note:** Use a high-level topic, but not the root. For information about cluster topic naming, see Cluster topics in *WebSphere MQ Publish/Subscribe User's Guide*.

8. To migrate the remainder of the hierarchy without introducing loops, repeat recursively from step 3 for each child queue manager that is not already in the cluster, and repeat recursively from step 3 for each parent queue manager that is not already in the cluster.

## Example

### What to do next

## Migrating a WebSphere MQ Version 6.0 two queue manager publish/subscribe hierarchy to a Version 7.0 hierarchy - parent first

How to migrate an existing WebSphere MQ Version 6.0 hierarchy, where the parent and child queue managers are on separate computers, into a WebSphere Version 7.0 hierarchy, migrating the parent queue manager first.

### Before you begin

To migrate the hierarchy, perform the following steps:

### About this task

1. Install WebSphere MQ Version 7.0 on the computer that contains the parent queue manager in the hierarchy, to migrate the parent queue manager to WebSphere MQ Version 7.0.
2. Use the **strmqbrk** command on the parent queue manager to migrate all publish/subscribe configuration data into WebSphere MQ Version 7.0. At this point you can either run as a mixed WebSphere MQ Version 6.0 and Version 7.0 hierarchy or continue with the migration by upgrading the child in the next step.

3. Install WebSphere MQ Version 7.0 on the computer that contains the child queue manager in the hierarchy, to migrate the child queue manager to WebSphere MQ Version 7.0.
4. Use the **strmqbrk** command on the child queue manager to migrate all publish/subscribe configuration data into WebSphere MQ Version 7.0, and check the migration log to verify that the migration was successful. The migration log is in the queue manager directory: for example, `/var/mqm/<QMgrName>` on Linux or `C:\Program Files\IBM\WebSphereMQ\qmgrs\<QMgrName>` on Windows, unless you have specified otherwise on the command line.

## Example

### What to do next

## Migrating a WebSphere MQ Version 6.0 two publish/subscribe queue manager hierarchy to a Version 7.0 hierarchy - child first

How to migrate an existing WebSphere MQ Version 6.0 hierarchy, where the parent and child queue managers are on separate computers, into a WebSphere Version 7.0 hierarchy, migrating the child queue manager first.

### About this task

To migrate the hierarchy, perform the following steps:

1. Install WebSphere MQ Version 7.0 on the computer that contains the child queue manager in the hierarchy, to migrate the child queue manager to WebSphere MQ Version 7.0.
2. Use the **strmqbrk** command on the child queue manager to migrate all publish/subscribe configuration data into WebSphere MQ Version 7.0. At this point you can either run as a mixed WebSphere MQ Version 6.0 and Version 7.0 hierarchy or continue with the migration by upgrading the parent in the next step.
3. Install WebSphere MQ Version 7.0 on the computer that contains the parent queue manager in the hierarchy, to migrate the parent queue manager to WebSphere MQ Version 7.0.
4. Use the **strmqbrk** command on the parent queue manager to migrate all publish/subscribe configuration data into WebSphere MQ Version 7.0, and check the migration log to verify that the migration was successful. The migration log is in the queue manager directory: for example, `/var/mqm/<QMgrName>` on Linux or `C:\Program Files\IBM\WebSphereMQ\qmgrs\<QMgrName>` on Windows, unless you have specified otherwise on the command line.

---

## Using publish/subscribe with WebSphere MQ classes for JMS

Existing WebSphere MQ classes for JMS applications run unchanged after you upgrade your queue manager to WebSphere MQ V7.0. In some circumstances, you must specify whether WebSphere MQ classes for JMS uses WebSphere MQ Version 6.0 or Version 7.0 publish/subscribe function.

The circumstances in which you must specify whether WebSphere MQ classes for JMS uses WebSphere MQ Version 6.0 or Version 7.0 publish/subscribe function are

described later in this topic. The advantages of using WebSphere MQ Version 7.0 publish/subscribe function, compared with WebSphere MQ Version 6.0 Publish/Subscribe, WebSphere Event Broker, or WebSphere Message Broker, are introduced in *WebSphere MQ Using Java*.

## WebSphere MQ messaging provider

The WebSphere MQ messaging provider has two modes of operation:

- *WebSphere MQ messaging provider normal mode*
- *WebSphere MQ messaging provider migration mode*

The WebSphere MQ messaging provider normal mode uses all the features of the WebSphere MQ Version 7.0 queue managers to implement JMS. This mode is used only to connect to a WebSphere MQ queue manager and can connect to WebSphere MQ Version 7.0 queue managers in either client or bindings mode. The WebSphere MQ messaging provider normal mode is optimized to use the new WebSphere MQ Version 7.0 function.

The WebSphere MQ messaging provider migration mode is based on WebSphere MQ Version 6.0 function and uses only features that were available in the WebSphere MQ Version 6.0 queue manager to implement JMS. You can connect to a WebSphere MQ Version 7.0 queue manager using WebSphere MQ messaging provider migration mode but you cannot use any of the Version 7.0 optimizations. This mode allows connections to either of the following queue manager versions:

- WebSphere MQ Version 7.0 queue manager in bindings or client mode, but this mode uses only those features that were available to a WebSphere MQ Version 6.0 queue manager
- WebSphere MQ Version 6.0 or earlier queue manager in client mode

If you want to connect to WebSphere Event Broker or WebSphere Message Broker using either WebSphere MQ Enterprise Transport or WebSphere MQ Real-Time Transport, use the WebSphere MQ messaging provider migration mode. If you use WebSphere MQ Real-Time Transport, the WebSphere MQ messaging provider migration mode is automatically selected, because you have explicitly selected properties in the connection factory object. Connection to WebSphere Event Broker or WebSphere Message Broker using the WebSphere MQ Enterprise Transport follows the general rules for mode selection described in “Rules for selecting the WebSphere MQ messaging provider mode.”

## Rules for selecting the WebSphere MQ messaging provider mode

If you are not using WebSphere MQ Real-Time Transport, the mode of operation used is determined primarily by the PROVIDERVERSION property of the connection factory. If you cannot change the connection factory you are using, you can use a client configuration property called `com.ibm.msg.client.wmq.overrideProviderVersion`, which overrides any setting on the connection factory. This override applies to all connection factories in the JVM but the actual connection factory objects are not modified. You can set PROVIDERVERSION to three possible values: 7, 6, or unspecified:

### **PROVIDERVERSION=7**

Uses the WebSphere MQ messaging provider normal mode.

If you set PROVIDERVERSION to 7 only the WebSphere MQ messaging provider normal mode of operation is available. If the queue manager that

is connected to as a result of the other settings in the connection factory is not a Version 7.0 queue manager, the `createConnection()` method fails with an exception.

The WebSphere MQ messaging provider normal mode uses the sharing conversations feature, and the number of conversations that can be shared is controlled by the `SHARECNV()` property on the server connection channel. If this property is set to 0, you cannot use WebSphere MQ messaging provider normal mode and the `createConnection()` method fails with an exception.

#### **PROVIDERVERSION=6**

Uses the WebSphere MQ messaging provider migration mode.

The WebSphere MQ classes for JMS use the features and algorithms supplied with WebSphere MQ Version 6.0. If you want to connect to WebSphere Event Broker or WebSphere Message Broker using WebSphere MQ Enterprise Transport, you must use this mode. You can connect to a WebSphere MQ Version 7.0 queue manager using this mode, but none of the new features of a Version 7.0 queue manager are used, for example, read ahead or streaming.

#### **PROVIDERVERSION=unspecified**

This is the default value and the actual text is "unspecified".

A connection factory that was created with a previous version of WebSphere MQ classes for JMS in JNDI takes this value when the connection factory is used with V7.0 of WebSphere MQ classes for JMS. The following algorithm is used to determine which mode of operation is used. This algorithm is used when the `createConnection()` method is called and uses other aspects of the connection factory to determine if WebSphere MQ messaging provider normal mode or WebSphere MQ messaging provider migration mode is required.

- Firstly, an attempt to use WebSphere MQ messaging provider normal mode is made.
- If the queue manager connected is not WebSphere MQ Version 7.0, the connection is closed and WebSphere MQ messaging provider migration mode is used instead.
- If the `SHARECNV()` property on the server connection channel is set to 0, the connection is closed and WebSphere MQ messaging provider migration mode is used instead.
- If `BROKERVER` is set to V1 or unspecified, WebSphere MQ messaging provider normal mode continues to be used, and therefore any publish/subscribe operations use the new WebSphere MQ V7.0 features. If WebSphere Event Broker or WebSphere Message Broker are used in compatibility mode (and you want to use Version 6.0 publish/subscribe function rather than the WebSphere MQ Version 7 publish/subscribe function), set `PROVIDERVERSION` to 6 to ensure WebSphere MQ messaging provider migration mode is used.
- If `BROKERVER` is set to V2 and `BROKERQMGR` is nonblank, this means `BROKERQMGR` has been explicitly changed from the default, so the assumption is the connection factory really is intended for use with WebSphere Event Broker or WebSphere Message Broker and WebSphere MQ Enterprise Transport. Therefore WebSphere MQ messaging provider migration mode is used.
- If `BROKERVER` is set to V2, `BROKERQMGR` is blank, the specified `BROKERCONQ` command queue exists and can be opened for output

(that is, MQOPEN for output succeeds), and PSMODE on the queue manager is set to COMPAT or DISABLED, WebSphere MQ messaging provider migration mode is used.

You can find further guidance about using PROVIDERVERSION in “When to use PROVIDERVERSION .”

## When to use PROVIDERVERSION

There are two scenarios where you cannot use the algorithm described in “Rules for selecting the WebSphere MQ messaging provider mode” on page 106; consider using PROVIDERVERSION in these scenarios.

1. If WebSphere Event Broker or WebSphere Message Broker is in compatibility mode, you must specify PROVIDERVERSION for them to work correctly.
2. If you are using WebSphere Application Server Version 6.0.1, Version 6.0.2, or Version 6.1, you define connection factories using the WebSphere Application Server administrative console.

In WebSphere Application Server the default value of the BROKERVER property on a connection factory is V2. The default BROKERVER property for connection factories created by using JMSAdmin or WebSphere MQ Explorer is V1. This property is now “unspecified” in WebSphere MQ Version 7.0.

If BROKERVER is set to V2 (either because it was created by WebSphere Application Server or the connection factory has been used for publish/subscribe before) and the existing queue manager has a BROKERCONQ defined (because it has been used for publish/subscribe messaging before), the WebSphere MQ messaging provider migration mode is used.

However, if you want the application to use peer-to-peer communication and the application is using an existing queue manager that has previously been used for publish/subscribe, and has a connection factory with BROKERVER set to 2 (which is the default if the connection factory was created in WebSphere Application Server), the WebSphere MQ messaging provider migration mode is used. Using WebSphere MQ messaging provider migration mode in this case is unnecessary; use WebSphere MQ messaging provider normal mode instead. You can use one of the following methods to work around this:

- Set BROKERVER to V1 or unspecified. This is dependent on your application.
- Set PROVIDERVERSION to 7; this is a custom property in WebSphere Application Server Version 6.1. The option to set custom properties in WebSphere Application Server Version 6.1 and later is not currently documented in the WebSphere Application Server Information Center.

Alternatively, use the client configuration property (see “Rules for selecting the WebSphere MQ messaging provider mode” on page 106 for details about how you can specify this system property for all environments), or modify the queue manager connected so it does not have the BROKERCONQ, or make the queue unusable.

## Subscription name migration on the JMS client

On the JMS client, if the ConnectionFactory property brokerPubQ is not the default, WebSphere MQ adds the stream name to the subscription name.

In WebSphere MQ Version 6.0, a subscription name needed to be unique only within the stream and not across the queue manager. In WebSphere MQ Version 7.0, a subscription name must be unique across the queue manager. Therefore to

migrate WebSphere MQ Version 6.0 durable subscriptions to WebSphere MQ Version 7.0, the subscription names must be unique. WebSphere MQ does this when it migrates the queue manager, by appending the stream name to the existing subscription name. For any existing durable subscription that uses a stream other than the default of "SYSTEM.BROKER.DEFAULT.STREAM" the migration process appends the stream name to the subscription name.

On the JMS client, if the ConnectionFactory property brokerPubQ is not the default, it is assumed that a WebSphere MQ Version 6.0 durable subscription is being resumed, and WebSphere MQ Version 7.0 appends the stream name to match the action of the migration process. Subscription names that use the default stream are migrated across with the subscription name unchanged.

---

## Migration implications of mapping an alias queue to a topic object

WebSphere MQ Version 7.0 introduces an extension to the alias queue object that allows an alias queue to be mapped to a topic object.

The new TARGTYPE attribute allows you to specify that a queue alias resolves to a queue or a topic. The TARGQ attribute, defined in WebSphere MQ Version 6.0 as the name of the queue to which the alias queue resolves, is renamed to TARGET in WebSphere MQ Version 7.0 and generalized to allow you to specify the name of either a queue or a topic. The attribute name TARGQ is retained for compatibility with your existing programs.

This feature is useful for migrating your existing applications to a publish/subscribe message model.

A useful example of this feature is the queue to which statistics messages are written. Prior to WebSphere MQ Version 7.0 there could be only a single consumer of a statistic message because a single statistics message only was written to a queue and got from a queue.

By defining a queue alias that points to a topic object, it is possible for each person interested in processing statistics messages to subscribe to the topic, rather than getting from the queue, allowing multiple consumers of the statistics information.

Within a queue sharing group it is possible to define a queue alias as a group object - this means that each queue manager in the queue sharing group will create a queue alias definition with the same name and the same properties as the QSGDISP(GROUP) object.

The new TARGTYPE attribute may be set or altered in a QSGDISP(GROUP) object by a new Version 7.0 queue manager, so that the queue alias refers to a topic object. However, any Version 6 queue managers in the queue sharing group do not understand and will ignore the new TARGTYPE attribute. A V6 queue manager will interpret the queue alias as referring to a queue object, regardless of the setting of TARGTYPE.

Defining a queue alias is described in *WebSphere MQ System Administration Guide*.

---

## Migrated topologies

If you have a WebSphere MQ publish/subscribe broker network, you can continue to use this network unchanged. The introduction of WebSphere Message Broker or WebSphere Event Broker to your environment, and the creation of brokers in that broker domain, does not affect your WebSphere MQ publish/subscribe broker domain until you take specific action to connect the two networks.

If you want to have two separate, independent networks, you do not have to do anything. You can retain your existing WebSphere MQ publish/subscribe network, and install and configure a WebSphere Message Broker or WebSphere Event Broker network, without any interaction.

## Heterogeneous networks

A *heterogeneous network* is a network of brokers, some of which form a WebSphere MQ publish/subscribe network and some of which belong to WebSphere Message Broker or WebSphere Event Broker.

With WebSphere Message Broker and WebSphere Event Broker, there are two ways in which a broker can be joined to the WebSphere MQ publish/subscribe network; it can be joined as a leaf node or as a parent node.

### Leaf node

When a broker is joined as a leaf node, it is joined as a child broker of another broker in the WebSphere MQ publish/subscribe network.

Adding the broker as a leaf node rather than as a parent node causes the new broker to receive only some of the WebSphere MQ publish/subscribe message traffic that is directed to the brokers for which this new broker is a child broker.

### Parent node

When a broker is joined as a parent node, it is joined as a parent broker of one or more brokers in the WebSphere MQ publish/subscribe network.

Adding the broker as a parent node rather than as a leaf node causes the new broker to receive all the WebSphere MQ publish/subscribe message traffic that is directed to the child brokers for which this new broker is the parent broker.



---

## Chapter 8. Upgrading and migrating WebSphere MQ clients from WebSphere MQ Version 6.0 to WebSphere MQ Version 7.0

This section contains topics that describe how to upgrade and migrate clients from WebSphere MQ Version 6.0 to Version 7.0.

---

### Upgrading a WebSphere MQ client from Version 6.0 to Version 7.0

You can upgrade a WebSphere MQ Version 6.0 client to a WebSphere MQ Version 7.0 client without migrating.

However, if you are using SSL on Windows you must **migrate** a WebSphere MQ Version 5.3 client to WebSphere MQ Version 7.0 because of changes made to SSL support for WebSphere MQ Version 6.0. See Migrating Windows Secure Sockets Layer (SSL) connections for further guidance.

The Websphere MQ client configuration file is introduced in WebSphere MQ Version 7.0. For more information see *WebSphere MQ Clients*. This has a number of implications:

- It is no longer possible to use a client qm.ini on i5/OS and UNIX systems, or the Windows registry, to enable TCP KeepAlive for client applications. Use the KeepAlive attribute in the client configuration file instead.
- At WebSphere MQ V7.0, the ClientExitPath is configured using the client configuration file. Existing values for ClientExitPath, which are held in mqs.ini on UNIX and i5/OS, and in the registry on Windows, are moved to the client configuration file when you migrate. Any values subsequently added to mqs.ini or the Windows registry are ignored.
- Use of the Java System property com.ibm.mq.exitClasspath is deprecated. Use the JavaExitsClasspath attribute in the WebSphere MQ client configuration file instead.

#### AIX

During client installation on AIX , you can choose to upgrade your client using the SMIT tool. Change the **Automatically install requisite software** to **No** and **Overwrite same or newer versions** to **Yes**.

#### Windows

During client installation on Windows, you can choose to upgrade your client in the following way:

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

1. If you select Modify, see *Modifying the client installation* in *WebSphere MQ Clients*.
2. If you select Remove, see *Uninstalling WebSphere MQ client using the installation process* in *WebSphere MQ Clients*.

---

## Migration and client channel definition tables

In general, the internal format of the client channel definition table might change from one release level of WebSphere MQ to the next. Because of this, a WebSphere MQ client can use a client channel definition table only when it has been prepared by a server queue manager that is at the same release level as the client, or at an earlier release level.

A Version 7.0 WebSphere MQ client can use a client channel definition table that has been prepared by a Version 6.0 queue manager but a Version 6.0 client cannot use a client channel definition table that has been prepared by a Version 7.0 queue manager.

---

## Chapter 9. Migrating from WebSphere MQ Version 5.3 to WebSphere MQ Version 7.0

All of the topics in this section are specific to a WebSphere MQ Version 5.3 to WebSphere MQ Version 7.0 migration only. This section highlights some of the main areas that are believed to impact users and gives some guidance on the type of action you will need to take to continue 'business as usual'.

**Note:** Before you install or migrate any software, you are recommended to make a full backup of your system to ensure that you can recreate the system to its original working environment should any problems occur.

Specific information about migrating from WebSphere MQ Version 5.3 to WebSphere MQ Version 6.0 was provided in the WebSphere MQ Version 6.0 documentation. Refer to *Migration Information* and the *Quick Beginnings* for your platform. For z/OS, refer to *WebSphere MQ for z/OS System Setup Guide* and the *WebSphere MQ for z/OS Program Directory*. You can download this information from the WebSphere MQ library Web site: <http://www.ibm.com/software/integration/wmq/library/library6x.html>

### Some general considerations

The following lists some of the general issues to consider when making a migration plan.

- Develop a backup plan in which you back up the relevant information on the queue manager and server.
- Read the latest README file for the product you are working with. You can find these files on the IBM home pages for the relevant products.
- Do you need to have the queue manager active all the time? If this is the case, and you are unable to shut down the system running this queue manager, you might need to consider a different migration approach. This will mean performing the migration using the following general steps:
  - Copy all resources from the server concerned to another server.
  - Perform a migration on the duplicate server.
  - Switch over to the new server and queue manager at a convenient time.
- Be prepared to record a number of details about the existing system topology, including such things as the names of the queue managers and their queues, clients, channels and so on.
- This is also a good time to check through all existing queue managers to see whether there are any queues that are no longer needed, and whether there are any queue managers that are no longer required. You might also decide that you want to keep certain queue managers at an earlier level and administer them from a migrated system.
- After you migrate a queue manager to WebSphere MQ Version 7.0, you must start that queue manager to migrate your file system structure before you start any WebSphere MQ listener associated with that queue manager. Otherwise, you will not be able to start WebSphere MQ listeners after migration.

## Migrating from a beta version

If you have previously installed a WebSphere MQ Version 7.0 beta driver, you MUST uninstall this driver BEFORE you install the GA (general availability) version of WebSphere MQ Version 7.0.

## Supported environments

The supported environments vary according to the changes to the product. Check each of the sections for details of the environment that is impacted.

Any queue managers from previous versions of WebSphere MQ (for example WebSphere MQ Version 5.3 and MQSeries Version 5.2) are supported, provided that they are put onto a supported platform of a WebSphere MQ Version 7.0 system.

The operating system on which you want to run a migrated WebSphere MQ Version 7.0 queue manager must be a supported WebSphere MQ Version 7.0 platform as documented on the Supported Environment Web page at: <http://www.ibm.com/software/integration/wmq/requirements/>.

All advice in the WebSphere MQ Version 7.0 documentation about migration also applies to any previous versions of WebSphere MQ.

**Note:** Where WebSphere MQ Version 5.3 is stated, the information also applies to WebSphere MQ Version 5.3.1 on z/OS. For all distributed platforms, where WebSphere MQ Version 5.3 is stated, the information also applies to any previous versions of WebSphere MQ.

---

## 64-bit queue manager migration information

This section provides you with information regarding the changes that might be necessary for applications to work with the 64-bit queue manager in WebSphere MQ Version 7.0 .

### Introduction

The 64-bit queue manager, which was introduced in WebSphere MQ Version 6.0 for some distributed platforms, removes many of the 32-bit addressing limitations. It gives a large increase to the amount of memory available for applications and WebSphere MQ, and the number and size of shared storage segments is also increased.

This move to 64-bit queue managers is transparent to most applications. However, a small number of 32-bit applications might not work as expected and will require some form of modification. This migration documentation provides information to help you identify those applications and those files that are needed as 64-bit.

Where other changes have been made, and where these could have an impact on your current installation without your applications requiring a change (for example, performance), these have also been identified. As a general rule, 32-bit applications do not need to be modified, but 32-bit switches and exits must be rebuilt.

## Queue manager environments

The queue manager is 64-bit on the following WebSphere MQ Version 7.0 platforms. Both 32-bit and 64-bit applications are supported in all of these environments:

- AIX platform
- HP-UX Itanium® platform
- HP-UX PA-RISC platform
- Linux POWER™ platform
- Linux x86-64 platform
- Linux zSeries (s390x) platform
- Sun Solaris SPARC platform
- Sun Solaris x86-64 platform

The queue manager is 32-bit on the following WebSphere MQ Version 7.0 platforms:

- Microsoft Windows platforms (from WebSphere MQ Version 7.0, 64-bit applications are also supported)
- Linux x86 platform

On i5/OS, the queue manager supports teraspace addressing.

## API implications

Applications built using an earlier version of WebSphere MQ do not need to be recompiled when a new version of WebSphere MQ is installed, even when the installation of the new version of WebSphere MQ changes the queue manager from being 32-bit to 64-bit. This is because all versions of WebSphere MQ continue to support 32-bit applications. When the queue manager is 64-bit, all versions of WebSphere MQ also support 64-bit applications. When WebSphere MQ for Windows is installed on a 64-bit version of the operating system, from WebSphere MQ Version 7.0, you can build WebSphere MQ applications as 64-bit.

Although the API has not changed, you need to be aware that in WebSphere MQ Version 5.3 and earlier versions, an `MQLONG`, an `int`, a `long`, and a `size_t` were all 32-bit so effectively interchangeable in their use. In WebSphere MQ Version 7.0 this is no longer true because a `long`, and a `size_t` are now 64-bit on at least some platforms.

You also need to be aware that pointers for 64-bit applications and 64-bit MQ exits are 64-bit. If your user-exits were storing pointers in a WebSphere MQ exit `ExitUserArea`, which is an `MQBYTE16`, you are now able to store only two pointers. This does not cause you a problem if you have written your user-exit to store a pointer to a block of pointers in the `ExitUserArea`. If your exits have not been written in this way, you must rewrite them to use this approach if there is now no longer enough room in `ExitUserArea` to store the pointers that your application needs.

The following table gives you the sizes of the basic 'C' types to aid you to see where similar problems might occur.

Table 9. Data type platform sizes for 32-bit and 64-bit applications (Bracketed figures are for 64-bit size)

| Data type                                                                   | Sun Solaris                      | HP-UX               | AIX               | Linux (POWER platform) | Linux (x86-64 platform) | Linux (zSeries 64-bit platform) | Microsoft Windows XP Professional x64 |
|-----------------------------------------------------------------------------|----------------------------------|---------------------|-------------------|------------------------|-------------------------|---------------------------------|---------------------------------------|
| char                                                                        | 1 byte (1 byte)                  | 1 byte (1 byte)     | 1 byte (1 byte)   | 1 byte (1 byte)        | 1 byte (1 byte)         | 1 byte (1 byte)                 | 1 byte (1 byte)                       |
| short                                                                       | 2 bytes (2 bytes)                | 2 bytes (2 bytes)   | 2 bytes (2 bytes) | 2 bytes (2 bytes)      | 2 bytes (2 bytes)       | 2 bytes (2 bytes)               | 2 bytes (2 bytes)                     |
| int                                                                         | 4 bytes (4 bytes)                | 4 bytes (4 bytes)   | 4 bytes (4 bytes) | 4 bytes (4 bytes)      | 4 bytes (4 bytes)       | 4 bytes (4 bytes)               | 4 bytes (4 bytes)                     |
| long                                                                        | 4 bytes (8 bytes)                | 4 bytes (8 bytes)   | 4 bytes (8 bytes) | 4 bytes (8 bytes)      | 4 bytes (8 bytes)       | 4 bytes (8 bytes)               | 4 bytes (4 bytes)                     |
| float                                                                       | 4 bytes (4 bytes)                | 4 bytes (4 bytes)   | 4 bytes (4 bytes) | 4 bytes (4 bytes)      | 4 bytes (4 bytes)       | 4 bytes (4 bytes)               | 4 bytes (4 bytes)                     |
| double                                                                      | 8 bytes (8 bytes)                | 8 bytes (8 bytes)   | 8 bytes (8 bytes) | 8 bytes (8 bytes)      | 8 bytes (8 bytes)       | 8 bytes (8 bytes)               | 8 bytes (8 bytes)                     |
| long double                                                                 | 16 bytes (16 bytes) <sup>1</sup> | 16 bytes (16 bytes) | 8 bytes (8 bytes) | 8 bytes (8 bytes)      | 12 bytes (16 bytes)     | 8 bytes (8 bytes)               | 8 bytes (8 bytes)                     |
| pointer                                                                     | 4 bytes (8 bytes)                | 4 bytes (8 bytes)   | 4 bytes (8 bytes) | 4 bytes (8 bytes)      | 4 bytes (8 bytes)       | 4 bytes (8 bytes)               | 4 bytes (8 bytes)                     |
| ptrdiff_t                                                                   | 4 bytes (8 bytes)                | 4 bytes (8 bytes)   | 4 bytes (8 bytes) | 4 bytes (8 bytes)      | 4 bytes (8 bytes)       | 4 bytes (8 bytes)               | 4 bytes (8 bytes)                     |
| size_t                                                                      | 4 bytes (8 bytes)                | 4 bytes (8 bytes)   | 4 bytes (8 bytes) | 4 bytes (8 bytes)      | 4 bytes (8 bytes)       | 4 bytes (8 bytes)               | 4 bytes (8 bytes)                     |
| time_t                                                                      | 4 bytes (8 bytes)                | 4 bytes (8 bytes)   | 4 bytes (8 bytes) | 4 bytes (8 bytes)      | 4 bytes (8 bytes)       | 4 bytes (8 bytes)               | 4 bytes (8 bytes)                     |
| clock_t                                                                     | 4 bytes (8 bytes)                | 4 bytes (4 bytes)   | 4 bytes (4 bytes) | 4 bytes (8 bytes)      | 4 bytes (8 bytes)       | 4 bytes (8 bytes)               | 4 bytes (4 bytes)                     |
| wchar_t                                                                     | 4 bytes (4 bytes)                | 4 bytes (4 bytes)   | 2 bytes (4 bytes) | 4 bytes (4 bytes)      | 4 bytes (4 bytes)       | 4 bytes (4 bytes)               | 2 bytes (2 bytes)                     |
| <b>Note:</b> 1. A 32-bit long double on Sun Solaris for x86-64 is 12 bytes. |                                  |                     |                   |                        |                         |                                 |                                       |

## WebSphere MQ Version 7.0 bindings: UNIX systems and Windows

On UNIX systems, three sets of bindings are provided for 64-bit applications: isolated, standard and fastpath bindings. For 32-bit applications only isolated and standard bindings are available. If a 32-bit application tries to connect to the queue manager using a fastpath binding it is silently downgraded to standard bindings. This will cause a noticeable loss of performance (it has virtually identical performance as when connecting directly with a standard binding) to anyone currently using fastpath and who is unwilling or unable to port the application to 64-bit. If you require to connect your 32-bit applications using fastpath bindings because performance is an issue, and you intend to keep your applications as 32-bit applications, then you must remain with WebSphere MQ Version 5.3.

In contrast, on Windows, two sets of bindings are provided for 64-bit applications: isolated and standard bindings. For 32-bit applications isolated and standard and fastpath bindings are available. If a 64-bit application tries to connect to the queue manager using a fastpath binding it is silently downgraded to standard bindings.

## EXTSHM on WebSphere MQ for AIX

On AIX, 32-bit applications can attach only 10 shared storage segments concurrently. When an application connects to WebSphere MQ, typically two storage segments are attached. However, the number of shared storage segments needed might increase depending on the workload of the queue manager. If, when connecting to WebSphere MQ, your 32-bit application fails to connect because it cannot attach all of the shared storage segments, consider setting the environment variable EXTSHM=ON in the environment of the queue manager. Set EXTSHM=ON before running the strmqm command and in the environment of your 32-bit applications. Alternatively you can connect to the queue manager using isolated bindings, which use only a single shared storage segment when connecting to the queue manager. Do this by replacing MQCONN calls with MQCONNX calls and specifying MQCNO\_ISOLATED\_BINDING in the MQCNO parameter of the MQCONNX calls.

There is virtually no limit to the number of shared storage segments attached by 64-bit applications.

## 32-bit and 64-bit files

This section provides you with information that helps you to identify where you need to replace your 32-bit files with 64-bit versions.

### Executable files and libraries

The majority of WebSphere MQ Version 7.0 files are built as 64-bit files on the following platforms:

- WebSphere MQ for AIX
- WebSphere MQ for HP-UX
- WebSphere MQ for Sun Solaris
- WebSphere MQ for Linux (POWER platform)
- WebSphere MQ for Linux (x86-64 platform)
- WebSphere MQ for Linux (zSeries s390x platform)

All the executable files in /xxx/mqm/bin and /xxx/mqm/samp/bin are 64-bit.

Most of the WebSphere MQ Version 7.0 libraries are 64-bit and put into /xxx/mqm/lib64. There are no symbolic links to the /xxx/mqm/lib64 directory. Any existing 64-bit client applications will continue to work.

The few libraries that are required by 32-bit applications are installed as usual in /xxx/mqm/lib and any current customer 32-bit applications will continue to work (subject to changes needed to exits and switches, see below) after upgrading to WebSphere MQ Version 7.0.

If you are using 64-bit applications, you might encounter problems if your LIBPATH or LD\_LIBRARY\_PATH variable includes the /usr/lib directory. The /usr/lib directory contains symbolic links to the 32-bit WebSphere MQ libraries, which do not work with 64-bit applications. Either remove /usr/lib from the library path that is used by your 64-bit application or remove the symbolic links from /usr/lib. For more information, see the *Quick Beginnings* guide for your platform.

## Exits and switch load files

On UNIX platforms, when using 32-bit applications with a 64-bit queue manager, some types of exit and XA switch load files also need to have a 64-bit version available for use by the queue manager. If the 64-bit version of the exit or XA switch load file is required and is not available, then the relevant API call or command fails.

On UNIX platforms, two attributes are supported in the `qm.ini` file for `ExitPath`. These are `ExitsDefaultPath=/var/mqm/exits` and `ExitsDefaultPath64=/var/mqm/exits64`. Using these ensures that the appropriate library can be found. If an exit is used in a WebSphere MQ cluster, this also ensures that the appropriate library on a remote system can be found.

The WebSphere MQ for Windows, Version 7.0 queue manager processes are 32-bit, therefore when using 64-bit applications, some types of exit and XA switch load files also need to have a 32-bit version available for use by the queue manager. If the 32-bit version of the exit or XA switch load file is required and is not available, then the relevant API call or command fails.

On WebSphere MQ for Windows, Version 7.0, two attributes are supported in the `qm.ini` file for `ExitPath`. These are `ExitsDefaultPath=install_location\exits` and `ExitsDefaultPath64=install_location\exits64`. Using these ensures that the appropriate library can be found. If an exit is used in a WebSphere MQ cluster, this also ensures that the appropriate library on a remote system can be found.

Where you have used your own directory and use a fully qualified path, you are advised to create a symbolic link from the `ExitsDefaultPath` and `ExitsDefaultPath64` to the appropriate library. If this is not provided, WebSphere MQ Version 7.0 searches for the library and uses the first matching library in the search path. If the library is still not found an error message is generated.

The following table lists the different types of Exit and Switch load files and notes whether 32-bit or 64-bit versions, or both, are required, according to whether 32-bit or 64-bit applications are being used:

| File types                                | 32-bit applications | 64-bit applications |
|-------------------------------------------|---------------------|---------------------|
| API-crossing exit                         | 32-bit              | 32-bit and 64-bit   |
| Data conversion exit                      | 32-bit              | 64-bit              |
| Server Channel exits (all types)          | 32-bit              | 32-bit              |
| Client Channel exits (all types)          | 32-bit              | 64-bit              |
| Installable service exit                  | 32-bit              | 32-bit              |
| Service trace module                      | 32-bit              | 32-bit and 64-bit   |
| Cluster WLM exit                          | 32-bit              | 32-bit              |
| Pub/Sub routing exit                      | 32-bit              | 32-bit              |
| Database switch load files                | 32-bit              | 32-bit and 64-bit   |
| External Transaction Manager AX libraries | 32-bit              | 64-bit              |



## Communications protocols

The only supported communications protocols for 64-bit applications are TCP and LU 6.2. Applications require the appropriate communications libraries to be available to them.

- A 32-bit client application requires the 32-bit communication library.
- A 64-bit client application requires the 64-bit communication library.
- On a 64-bit queue manager, the MQ channel processes are 64-bit and require the 64-bit communication library.

**Note:** With TCP, SSL is available for both 32-bit and 64-bit applications.

## Internal Transactional Manager XA support

WebSphere MQ Version 7.0 no longer supports versions of databases that do not include 64-bit support. You must rebuild your switch load files for these databases to support 64-bit processing as well as upgrading the databases to 64-bit versions. If you fail to do this, you will get an error when you start the queue manager and your application will fail to do any work with the database.

## COBOL copybooks

This section provides you with information in relation to the changes to the COBOL copybooks supplied with WebSphere MQ Version 7.0.

Due to the size of structures with pointers growing on 64-bit platforms, the lengths defined in some of the COBOL copybooks supplied with earlier versions of WebSphere MQ are incorrect for 64-bit applications. Updated COBOL copybooks are being supplied with WebSphere MQ Version 7.0 as follows:

### WebSphere MQ for z/OS

64-bit COBOL copybooks are not shipped in this release of the product.

### WebSphere MQ for AIX

All 32-bit COBOL copy books will be installed in the directory  
`/usr/mqm/inc/cobcpy32`.

Symbolic links to all the 32-bit COBOL copy books will be created in directory  
`/usr/mqm/inc`.

All 64-bit COBOL copy books will be installed in the directory  
`/usr/mqm/inc/cobcpy64`.

### All other UNIX platforms

All 32-bit COBOL copy books will be installed in the directory  
`/opt/mqm/inc/cobcpy32`.

Symbolic links to all the 32-bit COBOL copy books will be created in directory  
`/opt/mqm/inc`.

All 64-bit COBOL copy books will be installed in the directory  
`/opt/mqm/inc/cobcpy64`.

## WebSphere MQ for Windows

All 32-bit COBOL copy books will be installed in the directory C:\Program Files\IBM\WebSphere MQ\Tools\cobo1\copyBook.

**Note:** C:\Program Files\IBM\WebSphere MQ is the default directory for the installation of WebSphere MQ for Windows. If you install it in a different directory, you will adjust the specified path to take this into account.

---

## Internet Protocol Version 6 (IPv6) migration

This section deals with using IPv4 and IPv6 when you are planning to install WebSphere MQ Version 7.0

### General introduction

The Internet Protocol Version 6 (IPv6) is designed by the Internet Engineering Task Force (IETF) to replace the current version Internet Protocol, Version 4 (IPv4). IPv4 has been around for over 20 years and is one of the primary methods for machines to communicate to each other over the internet. IPv4 is limited to 32-bit addressing for internet addresses. These addresses are needed by all new machines added to the Internet and they are beginning to run out. The IETF is the controlling standards body for the Internet and to meet the growing demand for internet addresses has increased the number of digits used for Internet addresses from 32 to 128 bits. IPv6 offers a far larger number ( $2^{128}$ ) of internet addresses and should solve the address shortage for the foreseeable future. IPv6 is expected to gradually replace IPv4, with the two protocols coexisting for a number of years while this transition period exists. IPv6 also simplifies header formats and improves support for extensions and options, flow labeling capability, and consolidated authentication and privacy capabilities

WebSphere MQ Version 6.0 introduced the ability for queue managers to communicate using the IPv6 protocol in addition to the existing, IPv4, protocol.

You can use Internet Protocol Version 6 (IPv6) addresses in your WebSphere MQ classes for JMS and WebSphere MQ classes for Java applications.

Further information on IPv6 can be found at <http://www.ipv6.org/> and <http://www.ipv6forum.com/>.

## Key points in migrating to IPv6 and using WebSphere MQ Version 7.0

This section lists some key points to be aware of when you are planning to install WebSphere MQ Version 7.0 and using IPv6.

- WebSphere MQ Version 7.0 now recognizes IPv6 hexadecimal addresses (for example fe80:43e4:0204:acff:fe97:2c34:fde0:3485) as well as IPv4 dotted decimal addresses (for example 9.20.9.30).
- For a system running both IPv4 and IPv6 system, the connection name (CONNNAME) you specify for a given channel determines the IP protocol for the channel making a connection.

## Considerations when implementing IPv6 in a network

This section lists some things that you must consider when you are planning to install WebSphere MQ Version 7.0 on an IPv6 network.

- To ensure consistency across the network, you must plan the introduction of IPv6 for the whole network, especially where clusters are involved. For example, although a queue manager is now IPv6 capable, this does not imply that the queue managers it can communicate with are also IPv6 capable.
- When setting the domain name server (DNS) or equivalent, consider whether the system on which the target queue manager is running can resolve to an IPv4 address, an IPv6 address or a dual IPv4 and IPv6 address.
- If the system that you are installing WebSphere MQ Version 7.0 on does not support IPv6, WebSphere MQ Version 7.0 will only be able to connect using IPv4.
- For a queue manager running on an IPv6 enabled system to be able to communicate with a queue manager running on an IPv4 enabled system, the IPv4 enabled system must have a host name that resolves to an IPv4 address only.
- If there are multiple domain name servers in a WebSphere MQ network, each host name that is used in a channel definition must resolve to the same address (or addresses), regardless of which DNS is used.
- If the host name that is used in a channel definition resolves to a system that hosts a queue manager from WebSphere MQ Version 5.3 or earlier, the hostname must resolve to an IPv4 address only.

## Migrating a queue manager to IPv6

This section deals with migrating a queue manager when you are planning to install WebSphere MQ Version 7.0 on an IPv6 network.

The IPv6 protocol can be utilized only by WebSphere MQ Version 6.0 or later. To make use of the IPv6 protocol, install WebSphere MQ on a system that is IPv6 capable.

The preferred IP version that two systems use for communicating (if both IPv4 and IPv6 are available) is determined by a new queue manager attribute IPADDRV. This parameter has an effect only if the host name resolves ambiguously to both an IPv4 address and an IPv6 address.

To migrate a queue manager to use the IPv6 protocol:

1. Configure dual IPv4 and IPv6 protocols on the system where the queue manager to be migrated resides.
2. Install WebSphere MQ Version 7.0.
3. Add an entry to the DNS to resolve the host name of the system that is to be migrated, to both an IPv4 address and an IPv6 address.
4. Set the IPADDRV parameter to IPv6 (or set the LOCLADDR parameter to resolve to an IPv6 address).

**CAUTION:**

Not all IPv6 software can interpret an IPv4 mapped IPv6 address. If the combination of CONNAME and LOCLADDR results in an IPv4 mapped IPv6 address, ensure that the system hosting the target queue manager is capable of handling this.

Using mapped addresses can require protocol translators in the IP network.

## Migration scenarios (non-cluster topology)

It is possible to come up with a number of different interconnection possibilities, and the following sections aim to help you understand how WebSphere MQ will work in each case.

### Non-cluster migration scenario 1

Three systems exist that are IPv4 only capable. Each system hosts a queue manager (QM1, QM2, and QM3) and each queue manager connects to the other two. All CONNAMEs in the cluster channel definitions are made using DNS names rather than IP addresses.

Enable QM1 to be able to use channels running over IPv6 as follows:

1. Upgrade the host system to have dual IPv4 and IPv6 stacks.

**Important:** A listener is required for each IP stack.

2. Install WebSphere MQ Version 7.0.
3. Update the DNS table so that it has two entries for the system running QM1; one entry for its IPv4 address and one for its IPv6 address. This enables a DNS name request to return both IPv4 and IPv6 addresses for this host.
4. Set the queue manager IPADDRV attribute to IPv6.

**Note:** Even with these changes made to support IPv6 addressing, QM1 will still be able to communicate with queue managers (both existing and new ones) that are only IPv4 capable.

Enable QM2 to be able to use channels running over IPv6 as for QM1 above.

- Communications between QM1 and QM2 will now be over IPv6.
- Communications between QM1 and QM3 will still be over IPv4.
- Communications between QM2 and QM3 will still be over IPv4.

With the queue manager IPADDRV attribute set to IPv6, the preference has been set for the queue manager to connect using the IPv6 protocol. If a channel from QM1 to QM3 has LOCLADDR set to a host name which resolves to an IPv6 address, or both IPv4 and IPv6 addresses (with the IPADDRV attribute set to IPv6, the IPv6 address will be returned because that is the preference), this channel will attempt to use the IPv6 protocol. If the IPv6 protocol installed on the QM1 host system is capable of using a mapped address then QM1 will communicate with QM3 over IPv6. Otherwise, the channel will fail to resolve CONNAME.

While QM3 remains a WebSphere MQ Version 5.3 or earlier queue manager, you will need to check that all CONNAMEs used to start a channel to QM3 do not resolve to an IPv6 address or dual IPv4 and IPv6 addresses where the IPv6 address could be returned. This would cause

QM1 to attempt to start the channel over IPv6 which would fail, because it would be unable to resolve the CONNAME.

It is possible to upgrade a system to have dual IPv4 and IPv6 capability and still run a WebSphere MQ Version 5.3 or earlier queue manager on the system. Although it is not recommended to run this type of configuration, provided that the addresses that are returned to this level of queue manager are either IPv4 or an IPv4 mapped version of an IPv6 address, this should work.

### Non-cluster migration scenario 2

Three systems exist that are IPv4 only capable. Each system hosts a queue manager (QM1, QM2, and QM3) and each queue manager connects to the other two. All CONNAMES in the cluster channel definitions are made using IP addresses.

Because addresses have been specified instead of DNS names, to allow a queue manager to connect to another using the IPv6 protocol you will need to duplicate the definitions that use IPv4 addresses between them and provide them with IPv6 addresses instead. The original definitions that use IPv4 addresses will continue to work, but if you intend to take advantage of the IPv6 protocol, you will need to connect using the new definitions.

Enable QM1 to be able to use channels running over IPv6 as follows:

1. Upgrade the host system to have dual IPv4 and IPv6 stacks.

**Important:** A listener is required for each IP stack.

2. Install WebSphere MQ Version 7.0.
3. Duplicate the channel, transmission queue and, where applicable, any process definitions using IPv6 addresses where required.

**Note:** Even with these changes made to support IPv6 addressing, QM1 will still be able to communicate with existing queue managers that are only IPv4 capable.

Enable QM2 to be able to use channels running over IPv6 as for QM1 above.

1. Upgrade the host system to have dual IPv4 and IPv6 stacks.

**Important:** A listener is required for each IP stack.

2. Install WebSphere MQ Version 7.0.
3. Where necessary amend applications to write to the new remote queue (created above for QM1 with the IPv6 addresses).
4. Verify the channels can be started.

The queue managers can now connect as follows:

- QM1 can now connect with QM2 over either IPv4 or IPv6 depending on the channel the application writes its messages to.
- QM1 still connects with QM3 over IPv4 using the original definitions.

## Migrating a cluster to IPv6

This section deals with migrating clusters when you are planning to install WebSphere MQ Version 7.0 on an IPv6 capable network.

The following gives an overview of approaches that can be taken when migrating a cluster to WebSphere MQ Version 7.0 from WebSphere MQ Version 5.3. Because of the variations that can occur in a cluster, the detail and guidance on the likely course of action you will need to take are deliberately general.

## Migration scenarios (cluster topology)

Where an IPv6 capable system is to be added to a WebSphere MQ cluster, all full repository systems in that cluster must be IPv6 capable.

The following scenarios are seen as the ones most likely to occur in customer installations. They describe the changes that are likely to be required.

### Scenario 1

A WebSphere MQ Version 5.3 or earlier cluster is installed on IPv4 only capable systems and you need to connect an IPv6 only capable system into the cluster. All CONNAMES in cluster channel definitions are made using DNS names rather than IP addresses.

When adding a new IPv6 only system to the cluster, identify those queue managers that your new system will communicate with. These include:

- The queue managers your new system will send messages to.
- The queue managers your new system will receive messages from.
- The full repository queue managers

The systems that you have identified must be upgraded before introducing the new system.

Recommended migration procedure:

- Upgrade each of the systems hosting a full repository queue manager as shown in "Migrating a queue manager to IPv6" non-cluster scenario 1.
- Upgrade the remaining cluster systems which need to be IPv6 capable as shown in "Migrating a queue manager to IPv6" non-cluster scenario 1.

With this configuration:

- The new IPv6 only capable system will communicate with the cluster using IPv6 addressing
- All other IPv4 systems that connect into the cluster will continue to communicate using IPv4 addressing
- The systems in the cluster will be able to connect to each other using either IPv4 or IPv6 addressing. The decision as to which address is used depends on whether you have set IPADDRV to specify IPv4 or IPv6 connections.

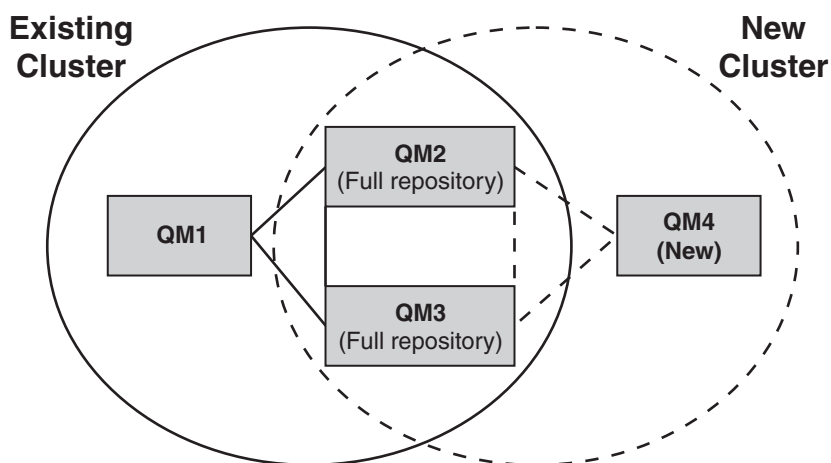
### Scenario 2

A WebSphere MQ Version 5.3 or earlier cluster is installed on IPv4 only capable systems and you need to connect an IPv6 only capable system into the cluster. Your network does not support adding both IPv6 and IPv4 addresses using the same host name or you are using IP addresses rather than DNS names in the cluster channel CONNAMES.

The problem here is likely to be that all of the systems cannot be switched to IPv6 simultaneously and some at least must remain only IPv4 capable. The systems that your new IPv6 only system communicates with must be IPv4 and IPv6 capable. We do not recommend simply adding a new set of IPv6 channels into the cluster for the IPv6 system to use, as the IPv4 system would also try to use them, resulting in communication errors.

The recommended approach is:

- Define a new cluster which contains the IPv6 only capable system or systems with new IPv6 addresses and channel definitions. The existing cluster remains, and contains the IPv4 only system definitions. The image below gives a pictorial representation of this. QM1, QM2, and QM3 represent the original IPv4 cluster. QM2, QM3, and QM4 represent the new cluster created to allow the IPv6 only capable system (QM4) to connect into your configuration.
- If you are using DNS names, you can give each of the systems separate DNS names for IPv4 and IPv6 (for example system1\_IPv4.ibm.com and system1\_IPv6.ibm.com).
- Define a new CLUSRCVR channel and any corresponding CLUSSDR channels using the new IPv6 names or IP addresses on each system in the new cluster. In this way the systems with only IPv4 or IPv6 capability do not see channels which they are not able to use and no communications error will result.



**Note:** There are both IPv4 and IPv6 definitions connecting the full repositories so that definitions for both new and existing cluster definitions are replicated between them. Also be aware that the queue managers QM1 and QM4 cannot communicate directly because they do not share a common network. They could communicate indirectly, for example by using ALIAS queues defined in the queue managers QM2 and QM3. In the configuration shown above you would need to pay attention to the ordering of application messages flowing between QM2 and QM3 because multiple routes exist, if this is relevant you could use BIND\_OPEN to fix the route.

## Abbreviated migration scenarios

This section gives some abbreviated scenarios for when you are planning to install WebSphere MQ Version 7.0.

### Abbreviated scenarios: Effects of CONNAME and LOCLADDR settings

The following table provides an overview of what will occur for the different TCP/IP stacks (IPv4 only, IPv6 only and dual IPv4 and IPv6 stacks) and, given the settings for CONNAME and LOCLADDR, the expected connection result.

**Note:** Using mapped addresses can require protocol translators in the IP network.

*Table 10. Effects of CONNAME and LOCLADDR settings*

| Stack Type               | CONNAME setting                                    | LOCLADDR setting                                   | Connection result                                                                                                                                 |
|--------------------------|----------------------------------------------------|----------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------|
| IPv4 only stack          | IPv4 address                                       |                                                    | Channel binds to IPv4 stack                                                                                                                       |
|                          | IPv6 address                                       |                                                    | Channel fails to resolve CONNAME                                                                                                                  |
|                          | Host name resolves to both IPv4 and IPv6 addresses |                                                    | Channel binds to IPv4 stack                                                                                                                       |
|                          | IPv4 address                                       | IPv4 address                                       | Channel binds to IPv4 stack                                                                                                                       |
|                          | IPv6 address                                       | IPv4 address                                       | Channel fails to resolve CONNAME                                                                                                                  |
|                          | Host name resolves to both IPv4 and IPv6 addresses | IPv4 address                                       | Channel binds to IPv4 stack                                                                                                                       |
|                          | Any address                                        | IPv6 address                                       | Channel fails to resolve LOCLADDR                                                                                                                 |
|                          | IPv4 address                                       | Host name resolves to both IPv4 and IPv6 addresses | Channel binds to IPv4 stack                                                                                                                       |
|                          | IPv6 address                                       | Host name resolves to both IPv4 and IPv6 addresses | Channel fails to resolve CONNAME                                                                                                                  |
|                          | Host name resolves to both IPv4 and IPv6 addresses | Host name resolves to both IPv4 and IPv6 addresses | Channel binds to IPv4 stack                                                                                                                       |
| Dual IPv4 and IPv6 stack | IPv4 address                                       |                                                    | Channel binds to IPv4 stack                                                                                                                       |
|                          | IPv6 address                                       |                                                    | Channel binds to IPv6 stack                                                                                                                       |
|                          | Host name resolves to both IPv4 and IPv6 addresses |                                                    | Channel binds to stack determined by IPADDRV                                                                                                      |
|                          | IPv4 address                                       | IPv4 address                                       | Channel binds to IPv4 stack                                                                                                                       |
|                          | IPv6 address                                       | IPv4 address                                       | Channel fails to resolve CONNAME                                                                                                                  |
|                          | Host name resolves to both IPv4 and IPv6 addresses | IPv4 address                                       | Channel binds to IPv4 stack                                                                                                                       |
|                          | IPv4 address                                       | IPv6 address                                       | Maps an IPv4 CONNAME to an IPv4 mapped IPv6 address. IPv6 implementations that do not support IPv4 mapped IPv6 addressing fail to resolve CONNAME |
|                          | IPv6 address                                       | IPv6 address                                       | Channel binds to IPv6 stack                                                                                                                       |
|                          | Host name resolves to both IPv4 and IPv6 addresses | IPv6 address                                       | Channel binds to IPv6 stack                                                                                                                       |
|                          | IPv4 address                                       | Host name resolves to both IPv4 and IPv6 addresses | Maps an IPv4 CONNAME to an IPv4 mapped IPv6 address. IPv6 implementations that do not support IPv4 mapped IPv6 addressing fail to resolve CONNAME |
| IPv6 address             | Host name resolves to both IPv4 and IPv6 addresses | Channel binds to IPv6 stack                        |                                                                                                                                                   |



Table 10. Effects of CONNAME and LOCLADDR settings (continued)

| Stack Type      | CONNAME setting                                    | LOCLADDR setting                                   | Connection result                                                                                                                                 |
|-----------------|----------------------------------------------------|----------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------|
|                 | Host name resolves to both IPv4 and IPv6 addresses | Host name resolves to both IPv4 and IPv6 addresses | Channel binds to IPv6 stack                                                                                                                       |
| IPv6 only stack | IPv4 address                                       |                                                    | Maps an IPv4 CONNAME to an IPv4 mapped IPv6 address. IPv6 implementations that do not support IPv4 mapped IPv6 addressing fail to resolve CONNAME |
|                 | IPv6 address                                       |                                                    | Channel binds to IPv6 stack                                                                                                                       |
|                 | Host name resolves to both IPv4 and IPv6 addresses |                                                    | Channel binds to IPv6 stack                                                                                                                       |
|                 | Any address                                        | IPv4 address                                       | Channel fails to resolve LOCLADDR                                                                                                                 |
|                 | IPv4 address                                       | IPv6 address                                       | Maps an IPv4 CONNAME to an IPv4 mapped IPv6 address. IPv6 implementations that do not support IPv4 mapped IPv6 addressing fail to resolve CONNAME |
|                 | IPv6 address                                       | IPv6 address                                       | Channel binds to IPv6 stack                                                                                                                       |
|                 | Host name resolves to both IPv4 and IPv6 addresses | IPv6 address                                       | Channel binds to IPv6 stack                                                                                                                       |
|                 | IPv4 address                                       | Host name resolves to both IPv4 and IPv6 addresses | Maps an IPv4 CONNAME to an IPv4 mapped IPv6 address. IPv6 implementations that do not support IPv4 mapped IPv6 addressing fail to resolve CONNAME |
|                 | IPv6 address                                       | Host name resolves to both IPv4 and IPv6 addresses | Channel binds to IPv6 stack                                                                                                                       |
|                 | Host name resolves to both IPv4 and IPv6 addresses | Host name resolves to both IPv4 and IPv6 addresses | Channel binds to IPv6 stack                                                                                                                       |

## Abbreviated scenarios: System configurations

Table 12 on page 128 gives a number of abbreviated scenarios based on the configuration of the installed queue managers and the IP configuration they are running on. The list is not intended to be exhaustive, but to give a number of examples of what to expect based on the configurations shown.

The following table lists the abbreviations used in Table 12 on page 128. The abbreviations in this table are combined in Table 12 on page 128 to give the configuration of the systems involved in trying to establish communication. For example:

- v53 + IPv6: Represents a WebSphere MQ Version 5.3 or earlier queue manager on a system with a TCP/IP version 6 stack
- v7 + Dual: Represents a WebSphere MQ Version 7.0 queue manager on system with a dual TCP/IP version 4 and version 6 stack

Table 11. Abbreviations used in system configurations

| Abbreviation | Meaning                                                                                           |
|--------------|---------------------------------------------------------------------------------------------------|
| v53          | a WebSphere MQ Version 5.3 or earlier queue manager                                               |
| v7           | a WebSphere MQ Version 7.0 queue manager                                                          |
| IPv4         | a system using an IPv4 only stack                                                                 |
| IPv6         | a system using an IPv6 only stack                                                                 |
| Dual         | a system using both an IPv4 and an IPv6 stack                                                     |
| IPv4DNS      | DNS returns an IPv4 address only for host name of system holding the responding queue manager     |
| IPv6DNS      | DNS returns an IPv6 address only for host name of system holding the responding queue manager     |
| DualDNS      | DNS returns an IPv4 and IPv6 address for host name of system holding the responding queue manager |
| LOCLADDR4    | The LOCLADDR parameter is set to IPv4 addressing                                                  |
| LOCLADDR6    | The LOCLADDR parameter is set to IPv6 addressing                                                  |
| IPADDR4      | IPADDRV is set to IPv4 addressing                                                                 |
| IPADDR6      | IPADDRV is set to IPv6 addressing                                                                 |

Table 12. System configurations

| Originating queue manager |                                                  | Responding queue manager |                                      |                    | Result                             |
|---------------------------|--------------------------------------------------|--------------------------|--------------------------------------|--------------------|------------------------------------|
| Queue manager and Stack   | LOCLADDR                                         | IPADDRV                  | Queue Manager and Stack              | DNS Return         |                                    |
| v53 + IPv6                | Any                                              | Not applicable           |                                      |                    | IP Error                           |
| v53 + IPv4 or v53 + Dual  | Both LOCLADDR4 & LOCLADDR6                       | Not applicable           | v53 + IPv4 or v53 + Dual             | IPv4DNS or DualDNS | IPv4 connection can be established |
| v53 + IPv4 or v53 + Dual  | Blank or LOCLADDR4                               | Not applicable           | v53 + IPv4 or v53 + Dual             | IPv4DNS or DualDNS | IPv4 connection can be established |
| v53 + IPv4 or v53 + Dual  | Blank or LOCLADDR4                               | Not applicable           | v53 + Dual                           | IPv6DNS            | Unable to resolve CONNAME          |
| v53 + IPv4 or v53 + Dual  | Blank or LOCLADDR4                               | Not applicable           | v53 + Dual or v7 + Dual or v7 + IPv4 | IPv4DNS or DualDNS | IPv4 connection can be established |
| v53 + IPv4 or v53 + Dual  | LOCLADDR6                                        | Not applicable           |                                      |                    | IP Error                           |
| v53 + IPv4 or v53 + Dual  | Blank or LOCLADDR4 or both LOCLADDR4 & LOCLADDR6 | Not applicable           | v7 + IPv6                            | IPv6DNS            | Unable to resolve CONNAME          |
|                           |                                                  |                          |                                      |                    |                                    |

Table 12. System configurations (continued)

| Originating queue manager |                                         | Responding queue manager |                                       |                    | Result                                                                               |
|---------------------------|-----------------------------------------|--------------------------|---------------------------------------|--------------------|--------------------------------------------------------------------------------------|
| Queue manager and Stack   | LOCLADDR                                | IPADDRV                  | Queue Manager and Stack               | DNS Return         |                                                                                      |
| v7 + IPv4                 | Blank or LOCLADDR4                      | Not specified            | v53 + IPv4 or v53 + Dual or v7 + IPv4 | IPv4DNS or DualDNS | IPv4 connection can be established                                                   |
| v7 + IPv4                 | LOCADDR6                                | Not specified            |                                       |                    | Unable to resolve LOCLADDR                                                           |
| v7 + IPv4                 | Blank or LOCLADDR4                      | Not specified            | v7 + IPv6                             | IPv6DNS            | Unable to resolve CONNAME                                                            |
| v7 + IPv6                 | Blank or LOCLADDR6                      | Not specified            | v53 + Dual                            | DualDNS            | Attempts to start IPv6 channel and fails as there will be no IPv6 listener available |
| v7 + IPv6                 | Blank or LOCLADDR6                      | Not specified            | v53 + IPv4                            | IPv4DNS            | Attempts to start IPv6 channel and fails as there will be no IPv6 listener available |
| v7 + IPv6 or v7 + Dual    | LOCLADDR6                               | Blank or IPADDR6         | v7 + IPv6 or v7 + Dual                | IPv6DNS or DualDNS | IPv6 connection can be established                                                   |
| v7 + Dual                 | LOCLADDR6                               | IPADDR4                  | v7 + Dual                             | IPv4DNS or DualDNS | IPv6 connection can be established where mapped addressing can be used               |
| v7 + Dual                 | Blank or LOCLADDR4                      | IPADDR4                  | v53 + Dual                            | IPv4DNS or DualDNS | IPv4 connection can be established                                                   |
| v7 + Dual                 | Both LOCLADDR4 & LOCLADDR6              | Blank or IPADDR4         | v53 + Dual                            | IPv4DNS or DualDNS | IPv4 connection can be established                                                   |
| v7 + Dual                 | LOCLADDR4                               | IPADDR4                  |                                       |                    | Unable to resolve LOCLADDR                                                           |
| v7 + Dual                 | LOCLADDR6 or both LOCLADDR4 & LOCLADDR6 | Blank or IPADDR6         | v7 + IPv6 or v7 + Dual                | IPv6DNS or DualDNS | IPv6 connection can be established                                                   |

## Migrating Windows Secure Sockets Layer (SSL) connections

This section deals with migrating Windows Secure Sockets Layer (SSL) connections from WebSphere MQ Version 5.3 to WebSphere MQ Version 7.0.

### General Introduction

WebSphere MQ Version 7.0 provides the Global Security Toolkit (GSKit) on Windows platforms for improved SSL (Secure Sockets Layer) support for queue manager and WebSphere MQ client channels. Follow the guidance in this section to determine whether WebSphere MQ Version 5.3 queue managers or clients have been set up to use SSL connections, and to ensure these channels continue to work

with WebSphere MQ Version 7.0. The migration process causes a copy of the certificates stored in the WebSphere MQ Certificate Stores used by WebSphere MQ Version 5.3, to be migrated to a GSKit Key database.

### Points to consider

- If you are intending to uninstall WebSphere MQ Version 5.3 before installing WebSphere MQ Version 7.0, you must do the following:
  - Before uninstalling WebSphere MQ Version 5.3, check that the certificate chains are complete. If they are not, use the AMQMCERT utility supplied with WebSphere MQ Version 5.3 to complete the chains.
  - After installing WebSphere MQ Version 7.0, run the AMQTCERT command from the command line to migrate your certificate stores.

If you install WebSphere MQ Version 7.0 after uninstalling WebSphere MQ Version 5.3, the certificate chain checker cannot then check the chains and AMQMCERT is no longer available to repair them. This also means that the processes used in the installation to migrate the stores cannot find them and you must manually check the chains and manually migrate the stores. See the *WebSphere MQ System Administration Guide* for details of the AMQMCERT and AMQTCERT commands.

- The Pre-installation Launchpad is run at the beginning of the installation process. From this the Check WebSphere MQ Certificate Store Wizard can be run. This checks that the certificate chains in the certificate stores are complete, that is, that each certificate in the chain is signed by the entity identified by the next certificate in the chain, terminating with a root CA certificate signed by the CA itself. If you elect not to run the Check WebSphere MQ Certificate Store Wizard, the Post-installation Prepare Wizard does not present any migration panels and your certificate stores can not be scheduled for automatic migration.
- The chain checker application used to verify that all the required certificates are there, before migrating certificates from the WebSphere MQ for Windows V5.3 store to the GSKit store, is available in WebSphere MQ V5.3 Fix Pack 10 (CSD10) or later.
- On client installation, the Check WebSphere MQ Certificate Store Wizard is run from the Install panels directly because there is no launchpad on the client installation to run it from.
- The Check WebSphere MQ Certificate Store Wizard and the Post-installation Prepare Wizard for scheduling certificate migration are made available only if you are migrating from a WebSphere MQ Version 5.3 installation directly to a WebSphere MQ Version 7.0 installation. If you uninstall WebSphere MQ Version 5.3 and then install WebSphere MQ Version 7.0, the installation process is not aware that the previous version was WebSphere MQ Version 5.3 and will not present the Check WebSphere MQ Certificate Store Wizard or any of the SSL migration panels to you. In this situation, you might consider running the Pre-installation Launchpad and the Check WebSphere MQ Certificate Store Wizard before uninstalling WebSphere MQ Version 5.3. This confirms the completeness of the certificate chains and allows you to import them using AMQTCERT after installing WebSphere MQ Version 7.0.
- If you are installing WebSphere MQ Version 7.0 silently, there are options which can be passed to the Post-installation Prepare Wizard silently to have it schedule certificate migration for you. If you follow this process, the Check WebSphere MQ Certificate Store Wizard is not run to check the certificate chains. If you are intending to run a silent installation, you should either run the Pre-installation

Launchpad and the Check WebSphere MQ Certificate Store Wizard to check the completeness of the certificate chains or check the stores manually using AMQCCERT prior to the installation.

## Certificates that are not migrated

A number of certificates are not migrated during this process. These are:

- Certificates that match GSKit's default supplied set. These are not migrated as GSKit provides its own set which are assumed to be the same or more up to date.
- Orphaned certificates that do not have a full valid Certification Authority certificate chain. A certificate can only be imported into a GSKit key database file if a certificate from its certification authority is already present or if it is a root certificate. Certificates can only be added to the GSKit key database starting with the root Certification Authority certificate, proceeding down the chain of intermediate Certification Authority certificates, if any exist, and ending with the personal certificate issued by the lowest member of the Certification Authority chain, again if any exists.
- Certificates that have expired.

## Types of certificate migration

There are two types of certificate migration.

- Automatic migration. For a queue manager the actual migration occurs when the queue manager is started for the first time. When the migration has completed, it is not attempted again even if the migration process failed. The queue manager attempts to start, irrespective of the success or failure of the migration. For a client, the actual migration occurs when the client first connects to the queue manager using an SSL channel. If the migration completes successfully then it is not attempted again. The starting of the client is dependent on the outcome of the migration; if the migration fails then so does the client. Where a certificate store has been successfully validated in the pre-installation phase, the Post-installation Prepare Wizard uses the automatic migration method for each of the queue managers and client stores specified.
- Manual migration. This occurs at the time the new Transfer Certificates (AMQTCERT) control command is run. Manual migration requires you to use AMQTCERT for each queue manager and client. You must specify the location and name stem of the WebSphere MQ Certificate Store and the GSKit key database to be used.

Automatic migration has the advantage that you do not need to specify the location and names for all the WebSphere MQ Certificates Stores and their corresponding GSKit key databases for all the queue managers and the clients because this is derived from the information gathered during the pre-installation processing.

## Friendly Name attribute

In the WebSphere MQ Certificate Store file there is one certificate assigned to the queue manager or client. During migration, the copy of this certificate is modified before it is imported into the GSKit database. The modification sets the certificate's Friendly Name attribute to the string `ibmwebspheremq` followed in lower case by the queue manager name or the client logon ID. This Friendly Name value becomes the label of the certificate in the GSKit key database. The previous Friendly Name value, if any, is lost.

## Working with migrated certificates

When WebSphere MQ Version 7.0 has been fully installed, and the certificates from the WebSphere MQ Certificate Stores have been migrated to the GSKit database, you can use the IBM Key Management (iKeyman) utility to view and manage your certificates. Full details of the iKeyman utility can be found in the WebSphere MQ Security book. Alternatively, you can use the command line interfaces iKeycmd and GSKCapiCmd to manage keys and digital certificates. These interfaces are described in *WebSphere MQ System Administration Guide*.

## Determining whether SSL connections have been set up

This section deals with determining whether SSL connections have been set up for WebSphere MQ.

- For channel definitions see the section "Checking whether channel definitions have been SSL-enabled".
- For channels set up using client application MQCONN calls, see the section "Checking whether client-connection channels set up using MQCONN calls have been SSL-enabled".

## Checking whether channel definitions have been SSL-enabled

For each queue manager on the computer you are working with, you must check whether any channels have been defined to use SSL. Display the SSLCIPH (CipherSpec) value for each channel defined on the queue manager. To do this you must have the queue manager running and have started the RUNMQSC environment. Enter `DIS CHL(*) CHLTYPE SSLCIPH` to display the channel details. The output should be similar to the following:

```
AMQ8414: Display Channel details.
CHANNEL(SYSTEM.DEF.SENDER) CHLTYPE(SDR) SSLCIPH()
AMQ8414: Display Channel details.
CHANNEL(SYSTEM.DEF.SERVER) CHLTYPE(SVR) SSLCIPH()
AMQ8414: Display Channel details.
CHANNEL(TO.QM4) CHLTYPE(CLUSRCVR) SSLCIPH(RC4_MD5_EXPORT)
AMQ8414: Display Channel details.
CHANNEL(TO.QM5) CHLTYPE(CLUSSDR) SSLCIPH(RC4_MD5_EXPORT)
AMQ8414: Display Channel details.
CHANNEL(TO.QM6) CHLTYPE(SVR) SSLCIPH()
AMQ8414: Display Channel details.
CHANNEL(TO.QM7) CHLTYPE(CLNTCONN) SSLCIPH(NULL_SHA)
AMQ8414: Display Channel details.
CHANNEL(TO.QM7) CHLTYPE(SVRCONN) SSLCIPH(NULL_SHA)
```

Channel definitions that have a value in the brackets after SSLCIPH are SSL channels. If there are any SSL channels the section "SSL migration steps" will apply. In the above example, the 'TO.QM4', 'TO.QM5', and both 'TO.QM7' channel definitions have a value for SSLCIPH.

Any client channel definition tables copied from another computer or accessed as a shared file on another computer will also need to be checked for SSLCIPH values. To check these values, either:

- Use `DIS CHL(*) CHLTYPE SSLCIPH` on the queue manager they were defined on
- If your client is running on a system that has a local queue manager, change the MQCHLLIB and MQCHLTAB environment variables which relate to RUNMQSC to specify the directory path and filename of the relevant client channel definition table, then use the `DIS CHL(*) CHLTYPE SSLCIPH` command on the

local queue manager. (Note: You should return the MQCHLLIB and MQCHLTAB settings to their previous values after completing this check.)

Any client-connection channel definitions that have been imported into Active Directory will also need to be checked for non-null SSLCIPH values. Display these definitions using the command `setmqscp -d`.

### **Checking whether client-connection channels set up using MQCONN calls are SSL-enabled**

For each client application that uses an MQCONN call, search the MQCD channel definition structure for the optional SSLCipherSpec field, which provides equivalent values to SSLCIPH.

If the value of the SSLCipherSpec field is not null, the MQI channel used by the client application is an SSL channel and the section "SSL migration steps" will apply.

## **SSL migration steps**

This section deals with steps for migrating Secure Sockets Layer (SSL) connections to work with WebSphere MQ Version 7.0.

Migrating SSL connections to work with WebSphere MQ Version 7.0 consists of several steps:

1. Ensuring WebSphere MQ certificate stores contain complete certificate chains
2. Migrating SSL certificates to Global Security Toolkit key database files
3. Converting Certificate Revocation Lists
4. Ensuring SSLPEER values have correctly ordered OU entries

### **Step 1: Ensuring WebSphere MQ certificate stores contain complete certificate chains**

This section deals with ensuring that complete certificate chains exist in the WebSphere MQ certificate store. This should be completed before installing WebSphere MQ Version 7.0.

WebSphere MQ Version 7.0 uses the Global Security Toolkit (GSKit) to manage SSL certificates. Before installing WebSphere MQ Version 7.0 you must ensure that all WebSphere MQ certificate stores contain complete certificate chains.

In Step 2: Migrating SSL certificates to Global Security Toolkit key database files, it gives guidance on migrating certificates used by WebSphere MQ Version 5.3 queue managers and WebSphere MQ clients into key database files for use with Global Security Toolkit.

As an alternative to Step 1: Ensuring WebSphere MQ certificate stores contain complete certificate chains and Step 2: Migrating SSL certificates to Global Security Toolkit key database files, you can manually configure a key database for each queue manager and WebSphere MQ client and import SSL certificates directly into it without migrating them. See "Working with the Secure Sockets Layer (SSL) on UNIX and Windows systems" in *WebSphere MQ Security* for details of how to do this. You will still need to complete Step 3: Ensuring Certificate Revocation Lists are in the correct format and Step 4: Ensuring SSLPEER values have correctly ordered OU entries.

The following sections give the background and necessary steps for "Ensuring WebSphere MQ certificate stores contain complete certificate chains":

- SSL certificates and certificate chains
- Why you must ensure complete SSL certificate chains exist before installation
- Where to locate WebSphere MQ certificate stores
- How to check WebSphere MQ certificate stores contain complete certificate chains
- How to add missing certification authority certificates into WebSphere MQ certificate stores
- What to do if WebSphere MQ version 7.0 has already been installed

### SSL certificates and certificate chains:

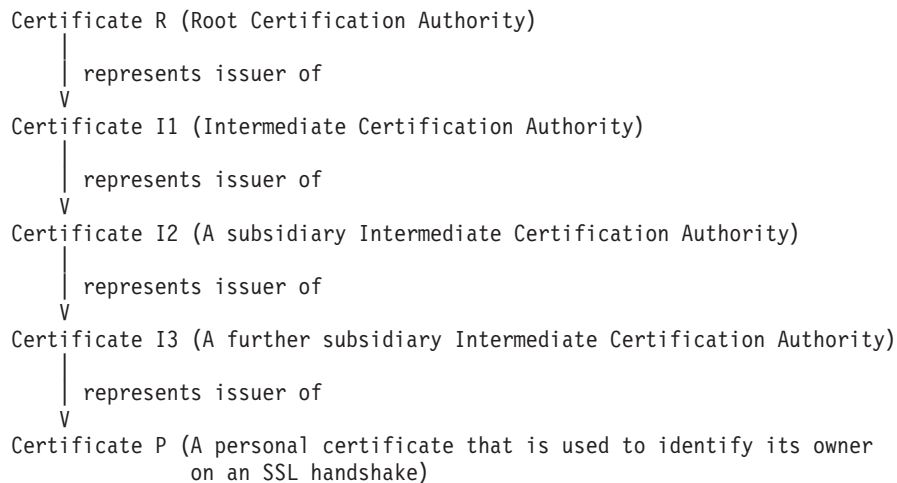
This section explains what SSL certificates are, what a complete certificate chain is and why WebSphere MQ certificate stores need them.

Personal certificates (certificates issued to an individual or a company) can be used by queue managers and WebSphere MQ clients to uniquely identify themselves when they are involved in starting an SSL connection.

Certification authority certificates are used by queue managers and WebSphere MQ clients to verify the authenticity of any personal certificate they receive during an SSL handshake (certification authority certificates are sometimes referred to as Certification Authority (CA) or signer certificates in other WebSphere MQ documentation.)

Each personal certificate has zero or more certificate chains of certification authority certificates that extend back to the root certification authority.

For example:



Certificate chains are used to verify the authenticity of each certificate in that chain, including the personal certificate. Each certificate in the chain is validated using its 'parent' certificate, which in turn is validated using the next certificate up the chain, and so on, from the personal certificate up to the root certification authority certificate.

The Global Security Toolkit (provided by WebSphere MQ Version 7.0) promotes good certificate housekeeping by automatically verifying the authenticity of any



personal certificate it manages. For this reason, it requires a complete set (or chain) of certification authority certificates to be stored with each personal certificate.

WebSphere MQ Version 5.3 on Windows allows personal certificates to be held in certificate stores without a complete certificate chain. If you attempt to migrate such certificates to a Global Security Toolkit key database this migration will fail, and your SSL channel connections will no longer work.

#### **Why you must ensure complete SSL certificate chains exist before installation:**

During the installation of WebSphere MQ Version 7.0, the AMQMCERT command line interface and the Manage SSL Certificates graphical interface, which provide access to WebSphere MQ Version 5.3 certificate stores, are removed, and you will not be able to add any missing certification authority certificates to these stores after installation.

#### **Where WebSphere MQ certificate stores are located:**

This section gives details on where you can find the WebSphere MQ certificate stores installed on your systems.

In WebSphere MQ Version 5.3 for Windows, certificates are held in a WebSphere MQ Certificate Store and are stored in a (store) file that has the suffix `.sto`.

The pathname for each of the WebSphere MQ certificate stores can be found by looking in the following attributes and variables:

- for WebSphere MQ queue managers, this is the value of the SSLKEYR attribute
- for WebSphere MQ clients, this is the value of the MQSSLKEYR environment variable
- for applications using MQCONNX calls, this is the value of the KeyRepository field in the MQSCO structure

The store file usually contains the following:

- The certificate assigned to the queue manager or client
- The chain of certification authority certificates
- Any other certificates that were loaded by default
- Any other personal or certification authority certificates.

#### **How to check WebSphere MQ certificate stores contain complete certificate chains:**

This section details how you can check if the SSL certificate chains that exist in your SSL certificate store are complete.

In WebSphere MQ Version 7.0, a new command is supplied that will check to see if the certificate chains are complete. This command is AMQCCERT (Check Certificate Chains) and can either be run from a command line or batch file, or as part of a wizard. This section will only deal with the use of the wizard. For information on using AMQCCERT from a command line or batch file, see the *WebSphere MQ System Administration Guide*.

The wizard is used to select the queue managers and clients that have certificate stores to migrate. It will run the AMQCCERT command against the certificate stores that have been specified and allows you to check the results of the

command. If the wizard has been run previously, any queue managers and clients that were previously selected, will display again.

The wizard also allows you to specify that a queue manager does not use SSL connections and the certificate store (if it exists) will not be checked or migrated.

Identify the queue managers or clients that are using SSL channels. See "Determining whether SSL connections have been set up" for guidance on how this is done.

1. Insert the WebSphere MQ Version 7.0 installation CD into the machine where the certificate store to be checked resides.
2. Run the WebSphere MQ Pre-install Launchpad.
3. Select the SSL tab in the WebSphere MQ Pre-install Launchpad.
4. From the WebSphere MQ Pre-install Launchpad, run the Check WebSphere MQ Certificate Store Wizard.
5. Use the Check WebSphere MQ Certificate Store Wizard to check all the certificate chains in your certificate stores. There is help information in the wizard to assist in each of the steps.

When the Check WebSphere MQ Certificate Store Wizard shows which certificate stores have passed and which have failed, the wizard allows you to look at the details of why a certificate store might have failed. The following is an example of the type of information shown when you display the details of why a certificate store might have failed:

```
C:\ssl\client
5724-B41 (C) Copyright IBM Corp. 1994, 2005. ALL RIGHTS RESERVED.
The number of certificates in the Microsoft Certificate Store
 'c:\ssl\client' is '13'.
```

```
The signer certificate 'GlobalSign Primary Class 1 CA' is missing for
 the following certificate.
```

```
Microsoft Certificate Store: 'c:\ssl\client'.
Certificate Subject: 'GlobalSign PersonalSign Class 1 CA'.
Certificate Issuer: 'GlobalSign Primary Class 1 CA'.
Certificate Serial Number: '0400 0000 0000 FA3D EEE9 D9'.
Certificate Valid From: '22/01/2004' to '28/01/2009'.
```

```
The signer certificate 'GlobalSign PersonalSign Class 1 CA' is missing
 for the following certificate.
```

```
Microsoft Certificate Store: 'c:\ssl\client'.
Certificate Subject: 'wm.shakespeare@hamlet.com'.
Certificate Issuer: 'GlobalSign PersonalSign Class 1 CA'.
Certificate Serial Number: '0100 0000 0001 0170 978B 1E'.
Certificate Valid From: '14/01/2005' to '14/02/2005'.
```

```
Certificate chain checking has completed with some failures.
The Check Certificate Chains (amqccert) command has completed.
```

As well as being visible through the wizard, this information, along with other progress information, is also written into a log file. This log file is located in the WebSphere MQ data directory and is named amqmccw.txt.

At this point you have the ability to replace out of date certificates or add missing ones and then go back to the wizard and recheck the stores to ensure they now pass. The wizard will only complete when all the selected certificate stores have been checked and have passed.

**Note:** If the state UNTESTED is displayed for the certificate stores, it indicates that the wizard was unable to launch AMQCCERT to test the given stores. The most likely cause of this is that AMQCCERT or one of its dependent libraries is not available. Check that the WebSphere MQ bin directory is available in the path.

**Note:** AMQMCERT and the Services GUI provided with WebSphere MQ Version 5.3 can be used to work with the certificate stores to correct any errors prior to migrating them.

### **Adding missing certification authority certificates into WebSphere MQ certificate stores:**

This section gives the commands necessary to add SSL certificates into WebSphere MQ certificate stores to complete the certificate chains before migrating to WebSphere MQ Version 7.0.

Missing certificates can be obtained from the certification authority that issues them. To import a certification authority certificate into a WebSphere MQ certificate store, issue the AMQMCERT command at a command prompt.

- For a queue manager enter:

```
amqmcert -a -s CertificateFilename -m queueManager
```

To list all certificates in the WebSphere MQ queue manager certificate store, enter:

```
amqmcert -l -m queueManager
```

Where:

- *CertificateFilename* is the fully qualified filename of the file where the certification authority certificate is stored in (certification authority certificates are usually provided in files with extensions .DER, .pb7, or .CER)
- *queueManager* is the name of the queue manager that requires the certificate to be added to its certificate store

- For a client (which uses the WebSphere MQ client certificate store identified by the MQSSLKEYR environment variable) enter:

```
amqmcert -a -s CertificateFilename
```

To list all certificates in the WebSphere MQ client certificate store, enter:

```
amqmcert -l
```

Where:

- *CertificateFilename* is the fully qualified filename of the file where the certification authority certificate is stored in (certification authority certificates are usually provided in files with extensions .DER, .pb7, or .CER)

- For client applications that use MQCONN calls enter:

```
amqmcert -a -s CertificateFilename -k KeyRepository
```

To list all certificates in the WebSphere MQ client certificate store, enter:

```
amqmcert -l -k KeyRepository
```

Where:

- *CertificateFilename* is the fully qualified filename of the file where the certification authority certificate is stored in (certification authority certificates are usually provided in files with extensions .DER, .pb7, or .CER)
- *KeyRepository* is the value (the fully qualified stem name of the repository file) stored in the MQSCO structure

For full details of the AMQMCERT command see the "WebSphere MQ Version 5.3 System Administration Guide".

## If WebSphere MQ Version 7.0 has already been installed:

This section deals with the issue of having upgraded to WebSphere MQ Version 7.0 without having complete certificate chains.

If you have installed WebSphere MQ version 7.0 without completing Step 1: Ensuring WebSphere MQ certificate stores contain complete certificate chains, you will not be able to use AMQMCERT to add any missing certification authority certificates.

You should still use AMQTCERT (Transfer Certificates) command as detailed in Step 2: Migrating SSL certificates to Global Security Toolkit key database files, to attempt certificate migration. Any certificates that have an incomplete certificate chain will fail the migration, and will be copied into a special directory.

If this occurs you will need to:

1. Individually import each certification authority certificate in the failed certificates issuer chain in the correct order
2. Import the failed certificate from the special directory

For more details of these steps, see Reasons and remedies for failed certificate migration.

## Step 2: Migrating SSL certificates into Global Security Toolkit database files

This section deals with migrating SSL certificates into Global Security Toolkit key database files and must be performed after installing WebSphere MQ Version 7.0.

The Global Security Toolkit (GSKit) requires SSL certificates to be stored in key database files that it creates and manages. This can be achieved by migrating the SSL certificates held in certificate stores used by WebSphere MQ Version 5.3 queue managers and WebSphere MQ clients into key database files.

This section details how to do this using the Prepare WebSphere MQ Wizard, and the AMQTCERT (Transfer Certificates) command.

Perform Step 1: Ensuring WebSphere MQ certificate stores contain complete certificate chains, before installing WebSphere MQ Version 7.0.

The following sections describe the process for migrating your SSL certificates to the GSKit database files:

- Using the Prepare WebSphere MQ Wizard to schedule certificate migration
- Using the AMQTCERT (Transfer Certificates) command

**Note:** You cannot schedule certificate migration using the Prepare WebSphere MQ Wizard unless you have run the Check WebSphere MQ Certificate Stores Wizard. AMQTCERT can be run without completing Step 1, but you are advised to complete Step 1 before moving on to Step 2.

### Using the Prepare WebSphere MQ Wizard to schedule certificate migration:

This section describes the use of the Prepare WebSphere MQ Wizard which can schedule the migration of your certificate stores and also prompts you to alter other system features for WebSphere MQ Version 7.0 SSL.

The Prepare WebSphere MQ Wizard has two functions in relation to SSL certificate migration:

- It offers to schedule the migration for all certificate stores checked by the Check WebSphere MQ Certificate Stores Wizard during Step 1: Ensuring WebSphere MQ certificate stores contain complete certificate chains
- It also prompts you to check that Step 3: Ensuring Certificate Revocation Lists are in the correct format and Step 4: Ensuring SSLPEER values have correctly ordered OU entries have been completed correctly.

The Prepare WebSphere MQ Wizard is automatically launched at the end of the installation of WebSphere MQ Version 7.0, and can also be launched from the Start Menu.

After starting the Prepare WebSphere MQ Wizard, follow the instructions on each of the panels to schedule the migration or to check that the scheduling of the migration of the certificates has been successful.

**Note:** The Prepare WebSphere MQ Wizard only offers the certificate stores checked by the Check WebSphere MQ Certificate Stores Wizard. Certificate chains checked manually using AMQCCERT are not presented for selection in the Prepare WebSphere MQ Wizard.

#### **Using the AMQTCERT (Transfer Certificates) command:**

This section describes the use of the AMQTCERT (Transfer Certificates) command, which can be used to create key database files and transfer existing SSL certificates held in certificate stores used by queue managers and WebSphere MQ clients.

The AMQTCERT (Transfer Certificates) command can be used to schedule migration for the next time the queue manager is started, or the next time the WebSphere MQ client connects to a queue manager using an SSL channel. The migration includes:

- Deriving the names of the source certificate store and target key database file from the queue manager SSLKeyRepository attribute (for queue managers)
- Creating the target key database file
- Attempting to migrate all SSL certificates found in the source certificate store

For details of all the options available for this command, including the options to specify key database filenames, list certificate stores, and cancel scheduled transfers, see the WebSphere MQ Version 7.0 System Administration Guide.

#### **Changing migration state**

If you have set a queue manager or client to migrate their stores automatically, it sometimes becomes necessary to migrate the certificate store manually. AMQTCERT can be used to cancel the migration for all or individual queue managers or clients, providing that the migration has not taken place. See the WebSphere MQ Version 7.0 System Administration Guide for details of how to use AMQTCERT to accomplish this.

*Automatically transferring SSL certificates used by all queue managers:*

This section gives an example of using the AMQTCERT command to automatically transfer the SSL certificates from all WebSphere MQ Version 5.3 queue managers certificate stores (on the current system).

You can schedule the transfer of SSL certificates used by all queue managers on the computer by entering at a command prompt:

```
amqtcert -a -m * -p password -e passwordExpiry
```

Where:

- password is the password for all key databases created
- passwordExpiry is the number of days until new passwords are required

For details of all the options available for this command, see the Version 7.0 *WebSphere MQ System Administration Guide*.

Transfer of certificates will occur for each queue manager when it next starts. For each queue manager, the path and stem of where the source WebSphere MQ certificate store and the target GSKit key database file are, is determined by the SSLKeyRepository queue manager attribute. WebSphere MQ client certificate stores will not be transferred by this command.

*Automatically transferring SSL certificates used by a specified queue manager:*

This section gives an example of using the AMQTCERT command to automatically transfer the SSL certificates from a specified WebSphere MQ Version 5.3 queue manager certificate store.

You can schedule the transfer of SSL certificates used by a specified queue manager on the computer by entering at a command prompt:

```
amqtcert -a -m queueManager -p password -e passwordExpiry
```

Where:

- queueManager is the name of the queue manager whose certificate store is to be migrated
- password is the password for all key databases created
- passwordExpiry is the number of days until new passwords are required

For details of all the options available for this command, see the Version 7.0 *WebSphere MQ System Administration Guide*.

Transfer of certificates will occur when the queue manager next starts. For the queue manager, the path and stem of where the source WebSphere MQ certificate store and the target GSKit key database file are, is determined by the SSLKeyRepository queue manager attribute. WebSphere MQ client certificate stores will not be transferred by this command.

*Automatically transferring SSL certificates used by WebSphere MQ clients:*

This section gives an example of using the AMQTCERT command to automatically transfer the SSL certificates from a WebSphere MQ client certificate store.

You can schedule the transfer of SSL certificates used by a WebSphere MQ client on the local system by entering at a command prompt:

```
amqtcert -a -c clientStoreFilename -p password -e passwordExpiry
```

Where:

- clientStoreFilename is the fully qualified filename (excluding the .sto suffix) of the WebSphere MQ client certificate store.

- password is the password for all key databases created
- passwordExpiry is the number of days until new passwords are required

For details of all the options available for this command, see the Version 7.0 *WebSphere MQ System Administration Guide*.

Transfer of certificates will occur the next time a WebSphere MQ client process which uses a key repository value matching clientStoreFilename connects to a queue manager using an SSL channel.

*Manually transferring SSL certificates used by a specified queue manager:*

This section gives an example of using the AMQTCERT command to manually transfer the SSL certificates from a WebSphere MQ queue manager certificate store.

You can manually transfer SSL certificates used by a WebSphere MQ queue manager on the computer by entering at a command prompt:

```
amqtcert -m QM1
 -w "C:\Program Files\IBM\WebSphere MQ\Qmgrs\QM1\SSL\key"
 -g "C:\Program Files\IBM\WebSphere MQ\Qmgrs\QM1\SSL\key"
 -p MyPassword
```

Where:

- -m specifies the name of the queue manager for the certificate store to be migrated
- -w specifies the fully qualified filename (excluding the .sto suffix) of the WebSphere MQ Certificate Store
- -g specifies the fully qualified filename (excluding the .kdb suffix) of GSKit key database
- -p states the password to be set for the GSKit key database

**Note:**

1. The command should appear on a single line, but has been displayed as above for the purposes of this documentation
2. The GSKit key database must not exist prior to using this command as this command will create it

For details of all the options available for this command, see the Version 7.0 *WebSphere MQ System Administration Guide*.

*Reasons and remedies for failed certificate transfer:*

This section gives some reasons and classifications for a failed certificate transfer and how to remedy some of them.

The AMQTCERT command has 2 classifications for why certificates fail to be transferred:

### **Orphan Certificates**

Orphan certificates do not have a complete certificate chain. When the AMQTCERT command detects an orphan certificate, it:

- Creates an OrphanCertificates subdirectory in the directory containing the key database file if one doesn't exist already

- Exports personal certificates into .pfx files
- Exports certification authority certificates into .cer files
- Outputs an error message to the amqerr01.log for each orphaned certificate, identifying the file, the certificate, and its issuer

To remedy this situation you will need to use the Global Security Toolkit to import the certificates missing from the orphaned certificate chain in strict order from root certification authority to the issuer of the orphaned certificate. Then import the orphan certificate from its file. For more details on how to do this see the Version 7.0 *WebSphere MQ Security*.

### **Failed Certificates**

Failed certificates fail to transfer for reasons other than incomplete certificate chains, for example, the certificate having become corrupted. When the AMQTCERT command detects a failed certificate, it:

- Creates an ImportFailedCertificates subdirectory in the directory containing the key database file if one doesn't exist already
- Exports personal certificates into .pfx files
- Exports certification authority certificates into .cer files
- Outputs an error message to the amqerr01.log for each failed certificate, identifying the file, the certificate, and its issuer

To try to remedy this situation you can get a new copy of the certificate from the certification authority. You will then need to use the Global Security Toolkit to import the certificates in strict order from root certification authority to the personal certificate. For more details on how to do this see the Version 7.0 *WebSphere MQ Security*.

### **Step 3: Converting Certificate Revocation Lists and Authority Revocation Lists**

This section describes how to convert the Certificate Revocation Lists and Authority Revocation Lists.

Certificate revocation lists (CRLs) and Authority Revocation Lists (ARLs) are available from Certification Authorities in 2 formats:

- DER-format
- PEM-format

WebSphere MQ Version 5.3 for Windows platforms allows CRLs and ARLs to be in PEM-format. In WebSphere MQ Version 7.0, the Global Security Toolkit requires CRLs and ARLs to be in DER-format and you must ensure that any CRLs and ARLs that you have in PEM-format are changed to be in DER-format.

You can change the PEM-format CRLs and ARLs before or after installing WebSphere MQ Version 7.0.

### **Changing Certificate Revocation Lists and Authority Revocation Lists into DER-format**

You can change PEM-format CRLs and ARLs into DER-format in several ways. Two of these are:

- Obtain replacement CRLs and ARLs in DER-format from your Certification Authority or authorities



- Use a commercial format conversion tool to convert your existing CRLs and ARLs

For each LDAP server that has been configured to hold CRL and ARL information used by WebSphere MQ, you must edit the appropriate LDIF file and update the certificateRevocationList; binary field with the DER-format CRL and ARL data. For further information on configuring and updating LDAP servers with CRL and ARL information, see “Accessing CRLs and ARLs” in *WebSphere MQ Security*.

#### **Step 4: Ensuring SSLPEER values have correctly ordered Organizational Unit entries**

This section describes how to check that the SSLPEER Organizational Unit (OU) values are in the correct order and how to change them if they are not.

**Note:** If you have already installed Fix Pack level 8 or higher for WebSphere MQ Version 5.3, you can ignore this step.

SSL-enabled channels that use the optional SSLPEER field to filter Distinguished Names must be checked to ensure multiple Organizational Unit entries (OUs) have been correctly ordered. This section contains information on why this check is needed, and how to do it.

#### **Why you need to check that SSLPEER values have correctly ordered OU entries:**

This section details why you have to change the order of your SSLPEER values if you are migrating from a WebSphere MQ Version 5.3 Fix Pack 7 or earlier installation.

Every SSL certificate contains a Distinguished Name (DN), used to uniquely identify the person or organization the certificate was issued to. The following attribute types are commonly found in the certificate’s Distinguished Name field:

**CN** Common Name  
**T** Title  
**O** Organization Name  
**OU** Organizational Unit Name  
**L** Locality Name  
**ST (or SP or S)** State or Province Name  
**C** Country

The certificate Distinguished Name can contain multiple OU attributes, listed in descending hierarchical order. For example, a certificate Distinguished Name could be specified as:

```
CN='QM2', O='IBM', C='GB', L='Hursley', OU='Software Group',
OU='Middleware', OU='MQ'
```

If a WebSphere MQ SSL channel has been configured with an optional SSLPEER value, after an SSL handshake, this value is compared to the Distinguished Name in any certificate received. If these values match then the connection is allowed, otherwise the connection is refused. In WebSphere MQ Version 5.3 Fix Pack 7 or earlier, channel definitions containing SSLPEER values with multiple OUs were

entered in ascending hierarchical order on Windows only. All other platforms were in descending hierarchical order. For example on Windows:

```
CN='QM2', O='IBM', C='GB', L='Hursley', OU='MQ', OU='Middleware',
OU='Software Group'
```

These differing approaches to specifying multiple OUs were resolved at Fix Pack 8 - multiple OUs are now always specified in descending hierarchical order in the SSLPEER value on all platforms.

### How to display and change SSLPEER values:

This section details how you can display the SSLPEER values that you have set and gives an example of how to change them.

#### Before you begin

Ensure that the queue manager you are changing the SSLPEER values for is running.

1. Start the WebSphere MQ command processor. You can do this by entering `runmqsc queueManager` at a command prompt where `queueManager` is the name of the queue manager you want to work with.
2. Display the SSLPEER value for each channel defined on the queue manager. You can do this by entering  
`DIS CHL(*) CHLTYPE SSLPEER`

The output should look similar to this:

```
AMQ8414: Display Channel details.
CHANNEL(SYSTEM.DEF.SENDER) CHLTYPE(SDR) SSLPEER()
AMQ8414: Display Channel details.
CHANNEL(SYSTEM.DEF.SERVER) CHLTYPE(SVR) SSLPEER()
AMQ8414: Display Channel details.
CHANNEL(TO.QM7) CHLTYPE(CLNTCONN) SSLPEER(CN='QM7', O='IBM', C='GB',
L='Hursley', OU='MQ', OU='Middleware', OU='Software Group')1
AMQ8414: Display Channel details.
CHANNEL(TO.QM8) CHLTYPE(SVR) SSLPEER(CN='QM2', O='IBM', C='GB',
L='Hursley', OU='MQ', OU='Middleware', OU='Software Group')1
```

3. Where the installation is from WebSphere MQ Version 5.3 Fix Pack 7 or earlier and the SSLPEER values contain multiple organizational unit names (OUs), you will need to reverse the order of these names. You can use the MQSC or the WebSphere MQ Explorer graphical interface to do this. For example, to change the SSLPEER value for the client-connection channel definition 'TO.QM7' in the above example, at the WebSphere MQ command processor prompt, enter  
`ALTER CHL(TO.QM7) CHLTYPE(CLNTCONN) SSLPEER(CN='QM7', O='IBM', C='GB',  
L='Hursley', OU='Software Group', OU='Middleware', OU='MQ')1`

**Note:** 1. These lines have been deliberately split to ensure they fit on the page. In the command line interface, all of these should appear on a single line unless the display forces them to wrap on the screen.

---

## Additional migration information

This section gives additional information if you are migrating directly from WebSphere MQ Version 5.3 to WebSphere MQ Version 7.0.

The information in the following topics is provided as guidance to areas that have changed in WebSphere MQ Version 7.0. In some cases you do not need to make any changes to your applications, but it might be beneficial to do so in the future.

## Channels implemented as queue manager objects on distributed platforms

This section gives information about moving channels to queue manager objects in WebSphere MQ Version 7.0 on distributed platforms.

In versions of WebSphere MQ before Version 6.0, channels on distributed platforms were stored collectively in a single channel definition file and secured by mqm or QMQMADM permissions. From WebSphere MQ Version 6.0, channels are implemented as queue manager objects in a similar way to objects such as queues and namelists. Two of the main advantages are that channel definitions are recoverable from media images stored in the queue manager logs and that authority to channel objects can be granted on a per object basis like any other object. To be able to recover a channel from a media image, you must first record an image of the channel. If you issue the `rcdmqimg` command with the `-t all` parameter, this happens automatically. If you specify particular types of object, you must add commands to record channel and `clntconn` objects. There are two new object authorities relevant to channel objects; control (`ctrl`) and control extended (`ctrlx`). You must have the appropriate authority in order to start, stop, ping, resolve and reset channels. Any user with ALLADM authority has the required authorities implicitly. Details of changes to media recovery and authority commands to support the new objects can be found in the WebSphere MQ System Administration Guide.

### Migration of the channels

Migration to queue manager objects is carried out the first time the queue manager is started after installing WebSphere MQ Version 7.0. A message (AMQ8047) summarizing the total number of channels successfully migrated is displayed and is also written to the queue manager error logs. If channel migration fails, set the `AMQ_MIGFORCE_CHANNEL` environment variable to force remigration the next time the queue manager is started. Once migration has completed, the channel definition file is renamed from `amqrfcda.dat` to `amqrfcda.old` or from `AMQRFCD4` (AMQRFCD4) to `AMQRFOLD` (AMQRFOLD) on i5/OS.

### Setting the MQCD Version field

The effect of initializing an MQCD to `MQCD_DEFAULT` has changed. You might need to change how you set the Version.

In versions of WebSphere MQ before Version 6.0, initializing the MQCD to the macro variables `MQCD_DEFAULT` or `MQCD_CLIENT_CONN_DEFAULT` set the Version field in the MQCD to `MQCD_CURRENT_VERSION`, which equates to 7 for WebSphere MQ Version 5.3, 8 for WebSphere MQ Version 6.0, and 9 for WebSphere MQ Version 7.0.

In WebSphere MQ Version 6.0, `MQCD_DEFAULT` and `MQCD_CLIENT_CONN_DEFAULT` set the Version to 6. In WebSphere MQ Version 7.0, to take advantage of features of the MQCD introduced in WebSphere MQ Version 5.3, 6.0, or 7.0, set Version in the MQCD explicitly to 7, 8, or 9 as appropriate, or to `MQCD_CURRENT_VERSION`. If you intend your applications to

be portable between several environments, use a newer version MQCD only if all of those environments support that version.

## DCE support in WebSphere MQ

This section gives information in relation to the removal of DCE support in WebSphere MQ Version 6.0 and later.

Support for DCE exits and the DCE name service has been withdrawn from WebSphere MQ and the executables and libraries to allow the use of these will no longer be provided. On HP-UX, the executables and libraries that were built using DCE threads will also no longer be provided.

Consider using SSL (Secure Sockets Layer) or other types of channel exit to secure your channels. For further information see *WebSphere MQ Security*.

## External Transaction Manager XA support

This section gives information in relation to changes for external Transaction Manager XA support.

Although there are no migration issues in relation to the support for external Transaction Manager XA, in WebSphere MQ Version 6.0 and later, the WebSphere MQ Server will be brought in line with the enhancements that have been made in earlier releases to the Extended Transactional Client. For example: a single generic library which supports all transaction managers except CICS; xa\_open string support of QMNAME, TPM and AXLIB.

In WebSphere MQ Version 6.0 and later a generic XA library is supplied. This is for the server and client, and is in 32-bit form on all UNIX platforms and in 64-bit form in WebSphere MQ for AIX, WebSphere MQ for HP-UX, WebSphere MQ for Sun Solaris, and WebSphere MQ for Linux (POWER platform).

The CICS XA libraries will be 32-bit until CICS changes to 64-bit processing.

As the UNIX library is now a shared library and not an archive library you are recommended to rebuild your applications. Not to do so means that you will remain at a level of code from when you last built your applications and will not pick up the changes to the library as it is enhanced through changes to functionality and fixes. To rebuild your applications using this library would be a one time exercise as the applications would then use the library files that are shipped with the product in this and any future releases.

## Queue manager attributes SCHINIT and SCMDSERV

The new queue manager attributes SCHINIT and SCMDSERV control whether the channel initiator and command server should start automatically when the queue manager starts. These will be automatically migrated but you can set them using the ALTER QMGR command.

## Starting and stopping services and listeners

You can now start and stop listeners and services using the START and STOP MQSC commands. However, the old commands strmqcsv, endmqcsv, runmqchi, runmqslr, endmqslr, and runmqtrm are still supported.

## UNIX directory permissions

On UNIX systems, queue manager directory permissions are changed to be stricter in WebSphere MQ Version 7.0 than in WebSphere MQ Version 5.3. The change will be made automatically when the queue manager is first started after migration. Any user applications which require access to the queue manager directories (e.g. error logs, FDCs) should run under the authority of a user ID in the **mqm** group.

## High availability clustering

This section gives information on considerations for migrating high availability (HA) clusters.

To update an HA cluster, plan the order of migration so that a system running WebSphere MQ Version 5.3 code can never access the files for a queue manager that is already running with Version 7.0 code. Implementing this rolling migration strategy to minimize total downtime will probably require you to disable failover between some systems in the cluster during the upgrade process.

Because channels are objects in WebSphere MQ Version 7.0, the channels and clntconn directories must be replicated between nodes for channel definitions. (The client channel table file is unchanged.)

On Windows HA systems, the certificate stores must be migrated to GSKit on all nodes which host a WebSphere MQ Version 7.0 queue manager.

---

## Considerations for migrating from WebSphere MQ for Linux Version 5.3

This section gives information on considerations for migrating from WebSphere MQ for Linux Version 5.3

### Migrating from earlier versions of WebSphere MQ for Linux

If you are migrating from a previous version of WebSphere MQ for Linux, you must uninstall your current version before installing WebSphere MQ Version 7.0.

If you are migrating from a previous level of this product, you **cannot** revert to your previous level, so back up your system **before** you install WebSphere MQ Version 7.0. This will enable you to back out the upgrade if necessary. If you back out the upgrade, however, you cannot recover any work performed by WebSphere MQ Version 7.0.

The use of RPM upgrade tools to migrate directly from one version of WebSphere MQ to another is not supported.

If you have already tried to upgrade WebSphere MQ using `rpm -U` or `rpm -F`, you might have deleted your old WebSphere MQ package entries from the RPM database without removing the product from your system. You might also have partially installed WebSphere MQ Version 7.0.

To continue upgrading to WebSphere MQ Version 7.0:

1. Find out which WebSphere MQ packages still have entries in your RPM database using: `rpm -qa | grep MQSeries`
2. Remove all remaining WebSphere MQ packages of any level from your system using: `rpm -e <package name>`

3. Remove the /opt/mqm directory by typing: `rm -rf /opt/mqm`
4. Install WebSphere MQ Version 7.0 using the instructions provided in *WebSphere MQ for Linux Version 7.0 Quick Beginnings*

## WebSphere MQ Explorer for Version 7.0 - error connecting to a migrated queue manager

If you cannot connect to a migrated queue manager and see an error like "SYSTEM.MQEXPLORER.REPLY.MODEL not defined", run:

```
strmqm -c
```

on that queue manager.

This command refreshes existing system objects (for example, setting the MCAUSER attribute of a channel definition to blanks), and creates the queue required by WebSphere MQ Explorer.

---

## Considerations for migrating from WebSphere MQ for Windows Version 5.3

This section gives information on considerations for migrating from WebSphere MQ for Windows Version 5.3

### Customizing your queue managers

#### Migrating from Windows NT

If you have an earlier version of WebSphere MQ installed on a Windows NT<sup>®</sup> machine, you will need to migrate the operating system to Windows XP, Windows 2003, , Windows Vista, or Windows Server 2008. Be aware that your current access control lists (ACLs) for your domain resources might become invalid.

On Windows NT, WebSphere MQ user account details are stored as security identifiers (SIDs) in your ACLs. When you migrate to a new Windows platform these SIDs are lost, invalidating the ACLs. This causes WebSphere MQ commands to fail, and you lose registry permissions and the WebSphere MQ Administration (mqm) security group.

You have the following options:

- To avoid losing WebSphere MQ user account access rights, remap the ACLs on your domain resources.

Consult your operating system documentation on migration from Windows NT for details about how to remap the ACLs on your domain resources. You do not need to reinstall WebSphere MQ if you choose this option.

- If you do not want to remap your ACLs, and would like to start afresh, do the following:
  1. Uninstall WebSphere MQ from Windows NT.
  2. Migrate from Windows NT to your new Windows platform.
  3. Install WebSphere MQ on your new Windows platform.

You must redefine your operating system user accounts and security groups as required, if you choose this option.

## Installable features

WebSphere MQ for z/OS comprises the following features:

**Base** This is required; it comprises all the main functions, including

- Administration and utilities
- Support for CICS, IMS and batch type applications using the WebSphere MQ Application Programming Interface, or C++
- Distributed queuing facility (supporting both TCP/IP and APPC communications)

### National language features

These contain error messages and panels in all the supported national languages. Each language has a language letter associated with it. The languages and letters are:

|          |                           |
|----------|---------------------------|
| <b>C</b> | Simplified Chinese        |
| <b>E</b> | U.S. English (mixed case) |
| <b>K</b> | Japanese                  |
| <b>U</b> | U.S. English (uppercase)  |

You must install the US English (mixed case) option. You can also install one or more other languages. (The installation process for other languages requires US English (mixed case) to be installed, even if you are not going to use US English (mixed case).)

### Client Attachment feature

This is optional; it is only required if you are going to attach clients to your subsystem. When you have installed this feature, there are no configuration parameters to set before you can attach clients to WebSphere MQ for z/OS. Administration for clients is available even if you do not install this feature.

### Java Support feature

This is optional; it is only required if you want to use Java and the Java Message Service. This is described in WebSphere MQ Using Java.

## WebSphere MQ service objects

This section gives information about service objects in WebSphere MQ for Windows.

In WebSphere MQ for Windows, services are now managed by service objects. The listener, command server, trigger monitor, and channel initiator services defined using the MMC plugin in earlier versions of WebSphere MQ will be migrated to queue manager objects as follows:

- The startup property of the command server (automatic or manual) is migrated to the SCMDSERV attribute of the queue manager.
- The startup property of the channel initiator (automatic or manual) is migrated to the SCHINIT attribute of the queue manager.
- For each listener defined in the WebSphere MQ Explorer, a queue manager LISTENER object is created.
- For each trigger monitor defined in the WebSphere MQ Explorer, a queue manager SERVICE object is created.

## WebSphere MQ Explorer - migrating from Version 5.3 to 7.0

Previous versions of the WebSphere MQ Explorer used Microsoft Management Console (MMC) based snap-ins and were part of the Server feature during installation. In WebSphere MQ Version 7.0 the WebSphere MQ Explorer component is provided as a separate feature and is no longer a part of the Server feature. Therefore when you want to migrate the WebSphere MQ Explorer from WebSphere MQ Version 5.3 to 7.0, you must choose a Custom installation. The Update installation can not install Version 7.0 of the WebSphere MQ Explorer because the Update installation tries to update the features installed at the previous level of WebSphere MQ.

## Migrating WebSphere MQ Services

WebSphere MQ services define one or more commands to be run when a queue manager is started or stopped. Examples of a service include:

- Listener
- Channel initiator
- Trigger monitor
- Command server
- Channel
- Queue manager custom service

When migrating from a version of WebSphere MQ earlier than Version 6, the WebSphere MQ services are automatically migrated into WebSphere MQ Objects. (For more information about WebSphere MQ Objects see the WebSphere MQ System Administration Guide.) These objects can then be administered in the WebSphere MQ Explorer.

Examples of the migration of services to WebSphere MQ Objects are:

- A listener is migrated to a WebSphere MQ listener object, equivalent to one defined by the `DEFINE LISTENER ... MQSC` command.
- A trigger monitor is migrated to a WebSphere MQ service object with `STARTCMD` set to `runmqtrm`.
- A channel is migrated to a WebSphere MQ service object with `STARTCMD` set to `runmqchl`.

It is important to note that after migration all services will run **after** the queue manager with which they are associated starts. This is the case even if the service previously ran before the associated queue manager started.

Only services that are associated with queue managers will be migrated. ROOT custom services, which are services that are not associated with a particular queue manager, are not migrated and they will remain as they were.

## Publish/Subscribe users

WebSphere MQ Version 7.0 contains components that provide Publish/Subscribe support that were previously shipped in SupportPac MA0C and included in Fix Pack 8 for WebSphere MQ Version 5.3. If you install WebSphere MQ Version 7.0, you no longer need the SupportPac and you should not install it. The files included with WebSphere MQ Version 7.0 supersede those in the SupportPac.

If you are using SupportPac MA0C with your previous version of WebSphere MQ, uninstall before you install WebSphere MQ Version 7.0. Uninstall by selecting **Start**



→ Settings → Control Panel → Add or Remove Programs → IBM MQSeries Publish/Subscribe, and then selecting Remove.

## Saved views with WebSphere MQ Explorer

WebSphere MQ Version 5.3 uses Microsoft Management Console (MMC) based snap-ins to provide the WebSphere MQ Explorer. MMC allows you to save the current configuration or 'view' as a .msc file.

WebSphere MQ Version 7.0 no longer uses MMC. Therefore, any views that you have previously saved will not work with the WebSphere MQ Explorer.

## MSCS migration issues

If you migrate an MSCS configuration from WebSphere MQ V5.3 to WebSphere MQ V7.0, you must take steps to migrate the SSL keystore and to deal with changes to the registry format.

### SSL Keystore Migration

During the installation phase, WebSphere MQ attempts to migrate any SSL keystores (.sto) defined to WebSphere MQ V7.0 SSL key repositories (.kdb). However, for this to succeed, the installation needs access to the queue manager SSL keystore. A queue manager under MSCS control has its keystore defined on the shared drive, which, at the time of the installation, is not accessible by the Passive node. It is therefore not possible to automatically migrate any WebSphere MQ V5.3 SSL keystores for queue managers under MSCS control as part of the installation process. Instead, export any certificates from the keystore before migration, and after migration is complete import them into the new WebSphere MQ V7.0 key repository.

### MSCS Registry Checkpointing

When you upgrade from WebSphere MQ V5.3, two changes are made to registry settings. However, as a consequence of the registry checkpointing behavior within MSCS, these changes are overwritten when the newly migrated node is brought online. MSCS restores the registry values to those present on the other node and thus the install time modifications are lost.

The changes in registry settings from WebSphere MQ V5.3 to V7.0 are as follows:

- HKEY\_LOCAL\_MACHINE\SOFTWARE\IBM\MQSeries\CurrentVersion\ Configuration\Services\qmname\Channel Initiator  
moves to HKEY\_LOCAL\_MACHINE\SOFTWARE\IBM\MQSeries\CurrentVersion\ Configuration\QueueManager\qmname\QueueManagerStartup
- HKEY\_LOCAL\_MACHINE\SOFTWARE\IBM\MQSeries\CurrentVersion\ Configuration\Services\qmname\Command Server  
moves to HKEY\_LOCAL\_MACHINE\SOFTWARE\IBM\MQSeries\CurrentVersion\ Configuration\QueueManager\qmname\InstanceData

In both these cases, the **Chinit** and **CmdServer** values after migration are set to Yes or No depending on the **Startup** value under the **Services** key before migration. As a result, these changes are lost and so any configured startup settings for the command server and channel initiator are lost after migration. To resolve this, apply APAR IC54088. Note that this APAR must be applied after installation and maintenance has been applied but before starting the queue manager on that node.

Alternatively, you can alter the startup values using runmqsc after migration is complete.

---

## PL/I support in WebSphere MQ for Windows

This section gives information in relation to PL/I support in WebSphere MQ Version 6.0 and later.

Support for PL/I has been withdrawn from WebSphere MQ for Windows. It continues to be supported for WebSphere MQ for z/OS.

---

## Migrating from UDP

The UDP protocol is no longer supported for channels. Use TCP instead. To migrate channels using UDP, stop the channels at both ends and issue the command `ALTER CHANNEL(CHANNEL.NAME) CHLTYPE(type) TRPTYPE(TCP)`. Change your channel definitions before installing WebSphere MQ for AIX, Version 7.0, as otherwise the UDP channels will not be migrated and will have to be defined again.

---

## Java archive (JAR) file `com.ibm.mqbind.jar`

This section gives information about the `com.ibm.mqbind.jar` file.

The Java archive (JAR) file `com.ibm.mqbind.jar` has been deprecated and is no longer supplied in WebSphere MQ Version 6.0 and later versions. If your Java applications have been relying on the contents of this file, you will need to change them to use the file `com.ibm.mq.jar` when you migrate to WebSphere MQ, Version 7.0.

---

## 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-k,u  
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.

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.

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

|           |          |            |
|-----------|----------|------------|
| AIX       | CICS     | DB2        |
| i5/OS     | IBM      | IBMLink™   |
| IMS       | MQSeries | OS/390     |
| POWER     | RACF     | SupportPac |
| WebSphere | z/OS     | zSeries    |

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

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

Intel and Itanium are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.

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

Linux is a trademark of Linus Torvalds in the United States, other countries, or both.

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



---

# Index

## A

- additional migration considerations
  - migrating to Version 7.0 52
- application stubs, coexistence with earlier versions 19
- ARM (automatic restart manager)
  - coexistence 82
  - migration 82
- authorities 37
- automatic restart manager (ARM)
  - coexistence 82
  - migration 82

## B

- base function 149

## C

- CF structure
  - migration considerations 58, 67, 75
- CFLEVEL 58, 68, 76
- change log inventory utility (CSQJU003)
  - migration considerations 81
- changing to full function WebSphere MQ 62, 72
- channel attributes
  - migration considerations 60, 70
- channel attributes, new for Version 5.3 78
- channel initiator
  - migration considerations 59, 68, 84
- channel initiator parameters
  - migration considerations 80
- channel objects
  - migration considerations 77
  - process definition 77
- CICS
  - migration considerations 62, 71
- CICS bridge
  - migration considerations 82
- CICS message table, migration considerations 79
- CICS users
  - migration 55
- client attachment feature 149
- client channel definition table
  - migrating to a later release level of WebSphere MQ 112
- cluster support
  - coexistence 82
  - migration 82
- clustering
  - migration considerations 62, 71
- coexistence PTFs 55, 65
- commands
  - CL
    - ENDMQM 37
    - ENDMQMCSVR 36
    - ENDSBS 36, 37

- commands (*continued*)
  - CL (*continued*)
    - RCDMQMIMG 36, 37, 42
    - STRSBS 41
    - WRKMQM 41
    - WRKMQMCHL 36
    - WRKMQMCHST 36
  - migration considerations 60, 70
- Compilers 71
- context profiles 76
- control commands
  - migrate broker function (strmqbrk) 89
- CSQINP1
  - migration considerations 61, 71
- CSQINP2
  - migration considerations 61, 71
- CSQJU003, migration considerations 81
- customizing
  - when migrating from previous versions 52

## D

- data conversion exits, migration considerations 81
- DB2 plan names
  - migration considerations 61, 71
- DB2 tables 56, 66, 74
- DEFINE MAXSMSGS command,
  - migration considerations 80
- deleting a WebSphere MQ server or client 14
- DiscardNonPersistentInputMsg 101
- DiscardNonPersistentResponse 101
- distributed queuing
  - testing customization 54
- DLQNonPersistentResponse 101
- downward compatibility 10

## E

- early code
  - multiple versions 16
- early code, migration considerations 79
- ENDMQM 37
- ENDMQMCSVR 36
- ENDSBS 36, 37
- euro currency support 84

## F

- features, installable 149
- full function WebSphere MQ
  - changing to 62, 72

## G

- group profiles
  - QMOMADM 41

## I

- initialization input data sets
  - migration considerations 61, 71, 80
- installable features 149
- installation
  - side-by-side 39
  - slip install 39
- installation verification program (IVP)
  - migration considerations 81
- IUCV
  - migration considerations 83
- IVP (installation verification program)
  - migration considerations 81

## J

- Japanese language letter 149
- Java client
  - migrating from MA88 41
- Java Support feature 149
- JMS
  - migration considerations 62, 72
- jobs 36, 37, 39

## L

- language letter 149
- libraries
  - QMOMJAVA 41
- library formats, migration considerations 79
- library names, migration considerations 65, 79
- logs
  - migration considerations 58, 68, 76, 81

## M

- MA0C 150
- MA88 SupportPac 41
- maximum number of uncommitted messages 80
- MaxMsgRetryCount 101
- MAXSMSGS 80
- messages
  - maximum number of uncommitted 80
- migrating
  - publish/subscribe 89
  - WebSphere MQ Services 150
- Migrating
  - MSCS 50

- migration
  - from Version 5.1
    - before you start 36
    - procedure 39
  - from Version 5.2
    - before you start 36
    - overview 35
    - procedure 39
  - from Version 5.3
    - overview 35
  - from Windows NT 148
  - Java client 41
  - post-migration tasks 42
  - Publish/Subscribe users 150
  - side-by-side 39
  - slip install 39
  - verifying 41
- Migration
  - MSCS
    - from V5.3 151
- migration & coexistence PTFs 73
- migration considerations
  - ARM 82
  - CF structure 58, 67, 75
  - CFLEVEL 58, 68, 76
  - change log inventory utility 81
  - changes to installation process 82
  - changing to full function WebSphere MQ 62, 72
  - channel attributes 60, 70
  - channel initiator 59, 68, 84
  - channel initiator parameters 80
  - channel initiator security 84
  - channel objects 77
  - CICS 62, 71
  - CICS bridge 82
  - CICS message table 79
  - clustering 62, 71
  - clusters 82
  - coexistence of different version queue managers 16
  - coexistence PTFs 55, 65
  - coexistence with earlier versions 16
  - commands 60, 70
  - context profiles 76
  - data conversion exits 81
  - DB2 plan names 61, 71
  - DB2 tables 56, 66, 74
  - DEFINE MAXSMGS command 80
  - early code libraries 79
  - euro currency support 84
  - initialization input data sets 61, 71, 80
  - installation verification program 81
  - IUCV 83
  - JMS 62, 72
  - library formats 79
  - library names 65, 79
  - logs 58, 68, 76, 81
  - migrating from Version 1.2 82
  - migrating from Version 2.1 80
  - migrating from Version 5.2 73
  - migrating from Version 5.3.1 54, 64
  - migrating to Version 6 64
  - migrating to Version 7 52
  - migration & coexistence PTFs 73

- migration considerations (*continued*)
  - mixed case RACF support for multiple queue manager versions 18
  - OpenEdition sockets 83
  - plan names 61, 71
  - queue objects 84
  - queue-sharing group 17, 55, 57, 65, 67, 73
  - return codes 81
  - reverting to a previous version 63
  - RRS 83
  - shared queues 81
  - software levels 54, 64
  - storage classes 83
  - system parameters 73, 80
  - TCP/IP 83
  - TCP/IP interface 59, 69
- MSCS
  - migrating 50
  - migration
    - from V5.3 151

**N**  
national language feature 149

**O**  
objects
 

- new for Version 5.3 79

OpenEdition sockets
 

- migration considerations 83

operations and control panels
 

- coexistence with earlier versions 19

**P**  
page set zero, migration 77, 84  
panels
 

- coexistence with earlier versions 19

plan names
 

- migration considerations 61, 71

prefix services 150  
previous versions
 

- coexistence with 16
- reverting to 63

process definitions, queue-sharing group 17  
PSMODE 55  
publish/subscribe
 

- upgrading 89

publish/subscribe daemon 55

**Q**  
QMQM subsystem 36, 38, 41  
QMQM user profile 37, 40  
QMQMADM group profile 41  
QMQMJAVA library 41  
queue manager
 

- coexistence of different versions in a queue-sharing group 16
- running multiple versions 16

queue manager custom services 150

queue objects, migration considerations 84  
queue-sharing groups
 

- coexistence of different version queue managers 16

migration considerations 55, 65, 73, 81  
mixed case RACF support for multiple queue manager versions 18  
process definitions 17  
recovery 76  
restrictions 17  
setting up new 57, 67  
quiescing 37

**R**  
RCDMQMIMG 36, 37, 42  
recovery, queue-sharing group 76  
removing a WebSphere MQ server or client 14  
resizeBuffer
 

- change in behavior 28

Resource Recovery Services (RRS)
 

- migration considerations 83

return codes, migration considerations 81  
reverting to a previous version 63  
ROOT custom services 150  
RRS (Resource Recovery Services)
 

- migration considerations 83

**S**  
security
 

- migration considerations 84

shared memory
 

- cleanup 37

shared queues
 

- migration considerations 81

side-by-side install 39  
Simplified Chinese
 

- language letter 149

slip install 39  
software levels 54, 64  
storage class
 

- migration considerations 83

strmqbrk command 89  
STRSBS 41  
stub programs
 

- coexistence with earlier versions 19

suffix services 150  
Support pack MA0C 150  
SupportPac MA88 41  
SyncPointIfPersistent 101  
system parameters
 

- migration considerations 73, 80

SYSTEM storage class, migration considerations 83

**T**  
TCP/IP
 

- migration considerations 83



## U

- uncommitted messages, maximum number 80
- uninstalling a WebSphere MQ server or client 14
- unit of recovery, maximum number of messages in 80
- unsupported release migration 63
- upgrading
  - publish/subscribe 89
- upward compatibility 10
- US English language letter 149
- user profiles
  - QMQM 37, 40
  - QMQMADM 41

## V

- verifying migration 41

## W

- WebSphere Event Broker users
  - migration 55
- WebSphere Message Broker users
  - migration 55
- WebSphere MQ Explorer users 151
- WebSphere MQ objects
  - new for Version 5.3 79
- WebSphere MQ Services 150
- WebSphere MQ subsystems, running
  - different versions 16
- WRKMQM 41
- WRKMQMCHL 36
- WRKMQMCHST 36



---

## 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.







SC34-6948-01



Spine information:



WebSphere MQ

Migration Guide

Version 7.0