7.5

Migrating and Upgrading IBM WebSphere MQ

Note

Before using this information and the product it supports, read the information in “Notices” on page 203.

This edition applies to version 7 release 5 of IBM® WebSphere® MQ and to all subsequent releases and modifications until otherwise indicated in new editions.

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

Contents

Migrating and upgrading WebSphere MQ

To migrate a queue manager to run on a new level of code, you must first upgrade WebSphere MQ to install the new code level. When you have verified the upgrade is successful, migrate the queue manager and all the applications and resources associated with it. Before starting this process, create a migration plan, based on the information in this migration guide. If you are applying maintenance, no migration is necessary, but you must test applications with the new level of WebSphere MQ code.

The IBM WebSphere MQ Migration Guide provides information to help you plan the process of migrating from an older version to a new version of the product.

- For an introduction to the guide and its contents, see the IBM Developer blog article WebSphere / IBM MQ Migration Guide.
- To view the guide in your web browser, click the following link: IBM WebSphere MQ Migration Guide - HTML version.
- To download the guide as a PDF file, click the following link: WebSphere / IBM MQ Migration Guide - PDF file.

If you want to see what changes have occurred between two versions of IBM WebSphere MQ, see “Changes that affect migration” on page 137. The changes that are listed affect the behavior of existing applications or the automation of management tasks. The lists do not include changes to IBM WebSphere MQ Explorer, or new features that do not change the way existing applications behave. Study the lists of changes to plan what migration tasks you must perform.

If you are not familiar with IBM WebSphere MQ migration, read the topics in “Introduction to IBM WebSphere MQ migration” on page 5.

If you are familiar with IBM WebSphere MQ migration, start with a planning topic listed in Migration planning topics by version and platform. If you already have a migration plan, see “Migrating IBM WebSphere MQ to a later version on UNIX, Linux, and Windows ” on page 83 for step by step guides about how to migrate queue managers, IBM WebSphere MQ clients, and applications.

If you are running IBM WebSphere MQ Version 7.0.1 on Linux®, UNIX, or Windows , and you are unfamiliar with multiple installations of IBM WebSphere MQ on the same server, examine these three migration scenarios:

1. “UNIX, Linux, and Windows: Single-stage migration to a later version” on page 60
2. “UNIX, Linux, and Windows: Side-by-side migration to a later version ” on page 65
3. “UNIX, Linux, and Windows: Multi-stage migration to a later version” on page 72

Also, consider how multiple installations on the same server can assist you deploying fix packs; see “UNIX, Linux, and Windows: Staging maintenance fixes” on page 183. If you are installing a fix pack that includes optional new function, see “Migrating queue managers to new-function fix packs” on page 170.

“IBM WebSphere MQ maintenance tasks” on page 141 describes how to apply and remove maintenance by version and platform, for platforms other than z/OS.

“Migration commands, utilities, and reference information” on page 186 describe commands and functions that assist you with migration.

Introduction to IBM WebSphere MQ migration

The introductory topics explain the concepts you must understand before planning migration tasks, where to find migration topics, and which migration paths are supported.

If you are running IBM WebSphere MQ Version 7.0.1 on Linux, UNIX, or Windows , and you are unfamiliar with multiple installations of IBM WebSphere MQ on the same server, examine these three migration scenarios:

1. “UNIX, Linux, and Windows: Single-stage migration to a later version” on page 60

2. “UNIX, Linux, and Windows: Side-by-side migration to a later version ” on page 65
3. “UNIX, Linux, and Windows: Multi-stage migration to a later version” on page 72

Also, consider how multiple installations on the same server can assist you deploying fix packs; see “UNIX, Linux, and Windows: Staging maintenance fixes” on page 183. If you are installing a fix pack that includes optional new function, see “Migrating queue managers to new-function fix packs” on page 170.

Migration paths

Where can you find topics describing how to migrate WebSphere MQ from one release to another? Can you revert a migration to an earlier release? Can you migrate from an unsupported WebSphere MQ release to any later release directly?

Use the links in Table 1 on page 7 and in “Where to find a topic about a specific migration path” on page 7, to find product documentation topics explaining how to migrate your system between different levels of WebSphere MQ. Each pair of release levels in the tables is called a migration path. If you migrate a system directly from one release to another, it is a direct migration path. If you migrate a system from one release to another, migrating first to an intermediate release, it is an indirect migration path.

Direct and indirect migration paths

The direct migration paths that are documented in the tables have been tested. The paths that are documented in this version of the product documentation are from one supported release to another. The paths are between releases that are currently within their support life, or only recently beyond their end-of-service dates. The paths between older releases are described in previous editions of the documentation. The links for those paths take you to previous editions of the documentation, where other direct migration paths were previously documented.

If you need to migrate a system from an old version WebSphere MQ, and no direct migration path is documented, you have two choices:

1. Migrate through one or more intermediate releases, using the published migration information.
2. Migrate directly from the old release to the new one.

If you choose the first option, you must start all the queue managers to migrate queue manager data to the intermediate release level. Depending on the release level, you might also need to start a channel on each queue manager to cause channel migration to take place.

If you choose the second option, you are choosing to follow an untested migration path.

Important:

1. If a queue manager is a member of a cluster, and is running at a release earlier than Version 6.0, you must migrate the queue manager to Version 6.0, Version 7.0, or Version 7.0.1, before migrating it to Version 7.5. You must start the queue manager after the first migration step, before proceeding to Version 7.5.
2. If you plan to migrate a queue manager on Linux for System x from 32 bit to 64 bit, you must do so by migrating from Version 6.0 to Version 7.0.1; see Linux: Migrating from a 32 bit to a 64 bit queue manager on System x (mi30200_.htm in the v7.0 documentation).

In both cases, you must read the topics from a combination of direct migration paths to discover what other changes you might need to make.

Reverting to a previous version

Reverting to a previous release (backward migration) is generally more difficult than migrating forward. On platforms other than z/OS, you cannot revert a queue manager to a previous release that differs in the version or release. Instead you are usually advised to back up your existing queue managers before upgrading them. The forward migration tasks start with the step of backing up your existing queue managers. Should you revert to the previous release, you can restore the queue managers to their

previous state from the backup copy. You cannot merge any messages or changes transacted on the new release to the system restored at the previous release level.

Related concepts

“The version naming scheme for IBM WebSphere MQ on UNIX, Linux, and Windows” on page 8
On UNIX, Linux, and Windows, IBM WebSphere MQ releases have a four-digit Version, Release, Modification, and Fix (VRMF) level code. The version and release parts of the code are significant; they identify the service life of a release. To run a queue manager at a different VR level, you must migrate the queue manager, its applications, and the environment in which it runs. Depending on the migration path, the migration might require more or less effort.

“Upgrade, migration, and maintenance of IBM WebSphere MQ on UNIX, Linux, and Windows” on page 25
You can install new releases of IBM WebSphere MQ to upgrade IBM WebSphere MQ to a new maintenance, release, or version level. Multiple installations at the same or different levels can coexist on the same UNIX, Linux, and Windows server. You can apply maintenance level upgrades to upgrade the maintenance or fix level. Applying maintenance level upgrades cannot change the version or release level of IBM WebSphere MQ. Maintenance level upgrades can be reversed, installations cannot be reversed.

Where to find a topic about a specific migration path

You can find topics about direct migration to the current release of IBM WebSphere MQ in this release of the IBM WebSphere MQ product documentation. Only supported migration paths are documented. Paths between other releases are in previous versions of the product documentation. This topic has links to topics about all the direct migration paths.

Note: IBM WebSphere MQ Version 7.5 is not available on IBM i and z/OS. For migration information for these two platforms, see the product documentation for versions of the product that do support for these two platforms. For links to other versions of the product documentation, see the IBM MQ family page.

Migration paths: IBM WebSphere MQ on UNIX, Linux, and Windows

Some of the links in Table 1 on page 7 go to topics in previous levels of the product documentation. No topics solely about migration were written for product documentation up to and including V5.3. Notice that backward migration is not possible for IBM WebSphere MQ on UNIX, Linux, and Windows.

If a migration path is marked as "not possible", you cannot restore a queue manager to the previous version from the current version. If a queue manager has not been started, you can uninstall the current version and reinstall a different version of IBM WebSphere MQ. To be precise, read the entries in Table 1 on page 7 as information about whether you can start a queue manager on a different version of IBM WebSphere MQ than the one it was last started at. Do not read the entries as information about which version of IBM WebSphere MQ can be installed. It does not matter what versions of IBM WebSphere MQ are installed between when a queue manager was last started and when it is next started.

If a migration path between two different versions is marked with a dash, it means that direct migration between the versions is not supported by IBM WebSphere MQ. You must follow an indirect migration path, which involves migrating IBM WebSphere MQ more than once.

Table 1. Migration paths:IBM WebSphere MQ on UNIX, Linux, and Windows

From / To	6.0	7.0.0	7.0.1	7.1	7.5
5.3	Supported	Supported	-	-	-
6.0		Supported	Supported	Supported	Supported

Table 1. Migration paths:IBM WebSphere MQ on UNIX, Linux, and Windows (continued)					
From / To	**6.0**	**7.0.0**	**7.0.1**	**7.1**	**7.5**
7.0.0	Not possible		Supported. *See Planning version 7.0 to version 7.0.1 migration by platform* in the Version 7.0.1 product documentation.	Supported. Follow the instructions in Planning IBM WebSphere MQ Version 7.0.1 to IBM WebSphere MQ Version 7.1 migration by platform in the Version 7.1.0 documentation.	Supported
7.0.1	Not possible	Supported. You must undo any application or administrative changes you made that took advantage of new functions in Version 7.0.1.		Supported. See Planning IBM WebSphere MQ Version 7.0.1 to IBM WebSphere MQ Version 7.1 migration by platform in the Version 7.1.0 documentation.	Supported
7.1	Not possible	Not possible	Not possible		Supported. See “Migration planning on UNIX, Linux, and Windows” on page 51

The version naming scheme for IBM WebSphere MQ on UNIX, Linux, and Windows

On UNIX, Linux, and Windows, IBM WebSphere MQ releases have a four-digit Version, Release, Modification, and Fix (VRMF) level code. The version and release parts of the code are significant; they identify the service life of a release. To run a queue manager at a different VR level, you must migrate the queue manager, its applications, and the environment in which it runs. Depending on the migration path, the migration might require more or less effort.

The full version of IBM WebSphere MQ on UNIX, Linux, and Windows is described by a four-digit VRMF code. The VRMF acronym stands for:

```
Version.Release.Modification.Fix
```

`7.5`, `7.5.0.1` are examples of full IBM WebSphere MQ version codes on UNIX, Linux, and Windows.

You can find the full version level of an IBM WebSphere MQ installation by typing the command **DSPMQVER**. It returns the full four-digit VRMF code.

Versions and releases of IBM WebSphere MQ are known by the first two digits of the VRMF code. The two digits are sometimes prefixed by a `V`, such as `V5.3`. A version of IBM WebSphere MQ always has a release level, even if it is the first release in a version.

The first release is normally labeled `V`*x*`.0`, for example, IBM WebSphere MQ Version 7.0. Occasionally, the first release of a version on a specific platform is not labeled `V`*x*`.0`. It is numbered to correspond the command level that has been implemented on the platform.

In documentation, the release level is sometimes dropped from the VRMF code, for example, V5. Dropping the release level can lead to ambiguity, if the context is not clear. For example, V5 might mean the whole of V5, or the release level V5.0, as opposed to the release level V5.2, or V5.3.

The third digit in the VRMF identifies the modification level of a release. A change in the third digit does not change the release. After upgrading IBM WebSphere MQ to modification level `7.0.1`, the release of IBM WebSphere MQ remains 7.0. However the command level does change to `7.0.1`.

The significance of the distinction between release and modification level concerns migration, and the service life of a product. Queue manager objects, such as queue managers, channels, queues, and messages do not require migration to upgrade to a new modification level. Nor do they require migration if the modification level is removed[1]. Migration might be required for a version or release level change.

Reverse migration of a version/release change is not possible on UNIX, Linux, and Windows. To be able to restore an earlier version or release level of a queue manager, you must back it up before upgrading. If you do restore it, you restore the queue manager, and its data, to the state it was in when you backed it up.

The other difference concerns service life. A new version or release has a new service end date. New modification levels generally do not result in a new service end date. But if a modification level is announced, then a new service end date might be announced too.

The fourth digit in the VRMF code is the fix level. Fix levels do not affect the command level of the queue manager. No migration is required, and fix levels do not affect the service end date of a release.

Trailing zeros in the VRMF code are never significant, but are sometimes quoted for clarity. For example, you might see `7.0.0` to distinguish it from `7.0.1`, and `7.0.1.0` to distinguish it from `7.0.1.1`. `7.0.0` is no different to `7.0` or `7.0.0.0`, and `7.0.1` and `7.0.1.0` are the same level.

Modification levels and fix levels are known by three and four-digit VRMF codes. `7.0.1` is a modification level and `7.0.1.2` is a fix level. Modification levels are shipped as refresh packs, and fix levels as fix packs.

A refresh or fix pack is named using a two part name that uniquely identifies it. The first part of the name is a truncated VRMF. The second part of the name is the name new refresh or fix pack. So, for example, the name of the fix pack `7.0.1.2` for Windows is `7.0.1-WS-MQ-Windows-FP0002`, and the name of the refresh pack `7.0.1` for Windows is `7.0-WS-MQ-Windows-RP0001`.

Refresh packs and fix packs for a particular version/release are cumulative, from the initial release. You can apply any higher numbered refresh, or fix pack, of the same version/release to upgrade directly to that version level. You do not have to apply the intervening fixes. Refresh packs and fix packs are obtained as service through Fix Central.

The latest modification level is also used to refresh the version of IBM WebSphere MQ available through Passport Advantage®, or on physical media. When you order IBM WebSphere MQ you receive the latest modification level. The result of installing the manufacturing refresh is almost the same as applying the refresh pack to an earlier fix level of IBM WebSphere MQ. There is one important difference. Refresh packs are applied using a maintenance procedure, manufacturing refreshes are installed using an installation procedure. You can "unapply" a refresh pack to return to the previous fix level you had installed. You can only uninstall a manufacturing refresh, which removes IBM WebSphere MQ from your system.

In addition to fixes packaged as refresh packs and fix packs, you can also obtain interim fixes for IBM WebSphere MQ from Fix Central. Interim fixes are also known as emergency or test fixes, and are known collectively as interim fixes. The naming scheme for refresh and fix packs extends to interim fixes. Interim fixes are known either by their fix name, or by the list of APARs they fix. The interim fix for APAR IC60646 is called `7.0.0.2-WS-MQ-Windows-LAIC60646`, telling you it is based on `7.0.0.2`. .

When you apply new fix packs or refresh packs, all interim fixes are removed. The documentation with the fix pack or refresh pack tells you if the APARS associated with the interim fixes you have applied have been fixed. If they have not, check to see if there are new interim fixes, at the new level, for the APARs

[1] Applications using new functions introduced in a modification level do not work on an earlier level.

that concern you. If there are not, consult service. They might either tell you to reapply the interim fix, or supply a new interim fix.

Related concepts

"Migration paths" on page 6
Where can you find topics describing how to migrate WebSphere MQ from one release to another? Can you revert a migration to an earlier release? Can you migrate from an unsupported WebSphere MQ release to any later release directly?

"Upgrade, migration, and maintenance of IBM WebSphere MQ on UNIX, Linux, and Windows" on page 25
You can install new releases of IBM WebSphere MQ to upgrade IBM WebSphere MQ to a new maintenance, release, or version level. Multiple installations at the same or different levels can coexist on the same UNIX, Linux, and Windows server. You can apply maintenance level upgrades to upgrade the maintenance or fix level. Applying maintenance level upgrades cannot change the version or release level of IBM WebSphere MQ. Maintenance level upgrades can be reversed, installations cannot be reversed.

Internet Protocol Version 6 (IPv6) migration

This section deals with using IPv4 and IPv6 when you are thinking of installing IBM WebSphere MQ

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

IBM WebSphere MQ has the ability for queue managers to communicate using the IPv6 protocol in addition to the existing, IPv4, protocol.

Further information on IPv6 can be found at IPv6.

IBM WebSphere MQ platforms that support IPv6

This section lists the IBM WebSphere MQ platforms that support IPv6.

IPv6 is supported on the following IBM WebSphere MQ platforms:

- IBM WebSphere MQ for AIX®
- IBM WebSphere MQ for Linux
- IBM WebSphere MQ for Sun Solaris
- IBM WebSphere MQ for HP-UX
- IBM WebSphere MQ for Windows
- IBM WebSphere MQ for IBM i
- IBM WebSphere MQ for z/OS

Key points in migrating to IPv6 and using IBM WebSphere MQ

This section lists some key points to be aware of when you are thinking of installing IBM WebSphere MQ and using IPv6.

- IBM WebSphere MQ 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 (CONNAME) 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 should consider when you are thinking of installing IBM WebSphere MQ on an IPv6 network.

- To ensure consistency across the network, you should 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 doesn't 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 IBM WebSphere MQ on does not support IPv6, IBM WebSphere MQ 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 hostname that resolves to an IPv4 address only.
- If there are multiple domain name servers in a IBM WebSphere MQ network, each hostname used in a channel definition must resolve to the same address (or addresses), regardless of which DNS is used.

Migrating a queue manager to IPv6

This section deals with migrating a queue manager when you are thinking of installing IBM WebSphere MQ on an IPv6 network.

The IPv6 protocol can only be utilized by IBM WebSphere MQ or later. In order to make use of the IPv6 protocol, IBM WebSphere MQ must be installed 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 only has an effect if the hostname 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 IBM WebSphere MQ.
3. Add an entry to the DNS to resolve the hostname 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 IBM 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 the latest version of IBM WebSphere MQ.
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 as 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 queue manager on an earlier version of the product, 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, as 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 queue manager on an earlier version of the product, on the system. While it is not recommended to run this type of configuration, as long as 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 IBM WebSphere MQ.
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 IBM WebSphere MQ.
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 thinking of installing IBM WebSphere MQ on an IPv6 capable network.

The following gives an overview of approaches that can be taken when migrating a cluster to the latest version of IBM WebSphere MQ. Due to the variations that can occur within a cluster, the detail is deliberately general and should only be seen as a guide to the likely course of action you will need to take.

Migration scenarios (cluster topology)

Where an IPv6 capable system is to be added to a IBM 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 cluster from an earlier version of the product 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.

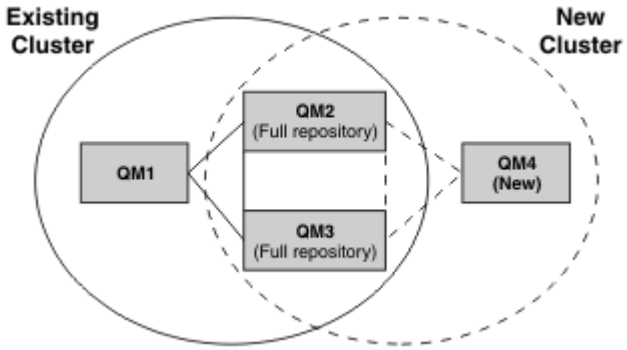
Scenario 2

A cluster from an earlier version of the product 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 hostname 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 thinking of installing clusters on IBM WebSphere MQ

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 2. Effects of CONNAME and LOCLADDR settings. Table showing the connection results for different TCP/IP stacks given the settings of CONNAME and LOCLADDR

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

Table 2. Effects of CONNAME and LOCLADDR settings. Table showing the connection results for different TCP/IP stacks given the settings of CONNAME and LOCLADDR *(continued)*

Stack Type	CONNAME setting	LOCLADDR setting	Connection result
	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
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

Table 2. Effects of CONNAME and LOCLADDR settings. Table showing the connection results for different TCP/IP stacks given the settings of CONNAME and LOCLADDR *(continued)*

Stack Type	CONNAME setting	LOCLADDR setting	Connection result
	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 4 on page 18 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 abbreviations are combined in Table 4 on page 18 to give the configuration of the systems involved in trying to establish communication. For example:

- v53 + IPv6: Represents a queue manager from an earlier version of the product on a system with a TCP/IP version 6 stack
- v6 + Dual: Represents a queue manager from the latest version of the product on system with a dual TCP/IP version 4 and version 6 stack

Table 3. Abbreviations used in system configurations. Table showing the abbreviation used in the documentation together with the meaning of the abbreviation.

Abbreviation	Meaning
v53	queue manager from an earlier version of the product
v6	queue manager from the latest version of the product
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 hostname of system holding the responding queue manager
IPv6DNS	DNS returns an IPv6 address only for hostname of system holding the responding queue manager
DualDNS	DNS returns an IPv4 and IPv6 address for hostname 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

Table 3. Abbreviations used in system configurations. Table showing the abbreviation used in the documentation together with the meaning of the abbreviation. *(continued)*

Abbreviation	Meaning
IPADDR6	IPADDRV is set to IPv6 addressing

Table 4. System configurations. Table showing the results for different settings on the originating and responding queue manager

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 v6 + Dual v6 + 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	v6 + IPv6	IPv6DNS	Unable to resolve CONNAME
v6 + IPv4	Blank or LOCLADDR4	Not specified	v53 + IPv4 or v53 + Dual or v6 + IPv4	IPv4DNS or DualDNS	IPv4 connection can be established
v6 + IPv4	LOCADD6	Not specified			Unable to resolve LOCLADDR
v6 + IPv4	Blank or LOCLADDR4	Not specified	v6 + IPv6	IPv6DNS	Unable to resolve CONNAME

Table 4. System configurations. Table showing the results for different settings on the originating and responding queue manager *(continued)*

Originating queue manager		Responding queue manager			Result
Queue manager and Stack	**LOCLADDR**	**IPADDRV**	**Queue Manager and Stack**	**DNS Return**	
v6 + IPv6	Blank or LOCLADDR6	Not specified	v53 + Dual	DualDNS	Attempts to start IPv6 channel and fails as there will be no IPv6 listener available
v6 + IPv6	Blank or LOCLADDR6	Not specified	v53 + IPv4	IPv4DNS	Attempts to start IPv6 channel and fails as there will be no IPv6 listener available
v6 + IPv6 or v6 + Dual	LOCLADDR6	Blank or IPADDR6	v6 + IPv6 or v6 + Dual	IPv6DNS or DualDNS	IPv6 connection can be established
v6 + Dual	LOCLADDR6	IPADDR4	v6 + Dual	IPv4DNS or DualDNS	IPv6 connection can be established where mapped addressing can be used
v6 + Dual	Blank or LOCLADDR4	IPADDR4	v53 + Dual	IPv4DNS or DualDNS	IPv4 connection can be established
v6 + Dual	Both LOCLADDR4 & LOCLADDR6	Blank or IPADDR4	v53 + Dual	IPv4DNS or DualDNS	IPv4 connection can be established
v6 + Dual	LOCLADDR4	IPADDR4			Unable to resolve LOCLADDR
v6 + Dual	LOCLADDR6 or both LOCLADDR4 & LOCLADDR6	Blank or IPADDR6	v6 + IPv6 or v6 + Dual	IPv6DNS or DualDNS	IPv6 connection can be established

Maintenance, upgrade, and migration

Maintenance is a reversible change to the code level of WebSphere MQ. Maintenance requires no migration. Upgrading is the process of changing the code level of WebSphere MQ. Upgrades can be backed out, as long as no migration has taken place. The process of removing an upgrade varies by platform and how the upgrade was applied. Upgrades that change the command level of WebSphere MQ require queue manager migration before applications can reconnect. Migration is the process of updating queue managers, and other objects, such as applications or administrative procedures. Typically, migration cannot be reversed.

Maintenance is the application of a fix pack, interim fix or PTF. It has one main characteristic. Those fixes, whether applied using a maintenance installation tool, or installed using a manufacturing refresh on top of an installation, are at the same command level as the existing code. No migration is required after applying maintenance. The installation can be restored to its previous level and any changed queue managers or applications will continue to work at the restored code level.

Upgrading and migration are related but distinct. Upgrading is the process of taking an existing WebSphere MQ installation and upgrading to a new level of code. Unless you are upgrading the fix level of WebSphere MQ, but not its command level, an upgrade must be followed by migration. Migration is the process of converting queue managers, applications, and other objects to run at a new command level.

An upgrade can take four different forms:

1. Application of a fix pack, interim fix, or a program temporary fix (PTF) using maintenance installation tool. Upgrades applied this way might not be called upgrades, but just fixes. Fixes, applied using a maintenance installation tool, can be rolled back completely as long as no queue manager migration has taken place, and WebSphere MQ is returned to its previous code level.
2. Installation of new code on top of existing code. You might be able to roll back an upgrade applied in this way; it depends on the platform. Generally speaking, you cannot roll back the installation of new code. To restore the old code level, you must retain the old installation media, and any fixes you applied.
3. Removal of the old level of code, followed by installation of the new level. The installers on very few platforms require you to remove an old installation first. Needless to say, to restore the old code level, you must reinstall it and any fixes.
4. Side by side installation. On UNIX, Linux, and Windows you can install different code levels are stored alongside each other on the same server. On UNIX, Linux, and Windows, you associate a queue manager with an installation, and start the queue manager. In WebSphere MQ, running multiple queue managers at different command levels on the same server is termed queue manager coexistence. You must not infer from this you can select different installations to run a queue manager at different times. Once a queue manager has been run, it is subject to the rules regarding reverting to earlier or later command levels.

Migration always follows an upgrade that changes the queue manager command level, both automatic and manual changes. Migration is the transformation of queue manager data, applications, and the environment that the queue manager runs in. Migration, and maintenance, and upgrading are described in the following topics.

IBM WebSphere MQ maintenance

Maintenance is the application of a reversible fix. Any changes to queue manager data are compatible with the previous code level.

Maintenance is the process of applying interim fixes or fix-packs. Maintenance is identified using the V.R.M.F version naming scheme. On UNIX, Linux, and Windows, maintenance is identified by either a third or fourth digit change.

An important characteristic of applying maintenance is that it must be reversible. Reversibility implies two things:

1. The previous level of code is fully restored.
2. Changes that are made to IBM WebSphere MQ objects are compatible. Changes are things like the creation or deletion of persistent messages, changes to queue managers, channels, topics, and queues. New and modified objects continue to work correctly with the restored level of code.

The reversibility of a maintenance package limits the extent of functional changes that are included in a maintenance package. No irreversible changes are included in a maintenance package. But, reversibility has limits. A maintenance package might include new programming and administrative interfaces. If you build new or modified applications to use the new interfaces, those applications do not work, if the maintenance package is removed.

Multi-instance queue managers are a good example. Should you remove the version 7.0.1 fix pack that upgraded version 7.0, then multi-instance queue manager functions no longer work. However, the queue managers continue to function correctly as single instance queue managers in version 7.0.

On a smaller scale, a fix pack or interim fix might introduce a new configuration parameter to solve a problem. If you remove the fix pack or interim fix, although the new interface introduced by the change is not available any more, IBM WebSphere MQ works with any objects that have been changed by the configuration parameter. For example, a new Java system property might introduce a parameter to set a code page for queue manager data conversion. The fix does not change any existing persistent queue manager state information. It can be removed, and the queue manager continues to work as before, but without the capability introduced in the fix.

Related concepts

"The version naming scheme for IBM WebSphere MQ on UNIX, Linux, and Windows" on page 8
On UNIX, Linux, and Windows, IBM WebSphere MQ releases have a four-digit Version, Release, Modification, and Fix (VRMF) level code. The version and release parts of the code are significant; they identify the service life of a release. To run a queue manager at a different VR level, you must migrate the queue manager, its applications, and the environment in which it runs. Depending on the migration path, the migration might require more or less effort.

IBM WebSphere MQ upgrade

The term upgrade, used loosely, covers both applying maintenance and modifying the version or release of a product. Upgrading is a change in any of the four V.R.M.F digits. After an upgrade has been applied, as long as no queue managers have been restarted, it can be backed out. The details vary by platform and how the upgrade was applied. Once a queue manager has been restarted, it depends on a number of factors whether the upgrade can be reversed. It depends on the platform, the V.R.M.F level, and the functions and command levels used by the queue manager at the new level.

Reversibility of a upgrade has two parts; backout of the installation to the previous code level, and reversion of any queue managers that have been started at the new code level, to work with the previous code level again.

Backout of an installation is not required on UNIX, Linux, and Windows. You can continue to run queue managers with the previous code level, after installing a new code level. The flexibility is due to those platforms supporting multi-installations.

On other platforms, which do not support multiple installations, it depends on how the upgrade has been applied. If an upgrade is applied as a fix pack, which is only available for maintenance and fix level changes, it can be reversed. If the upgrade is installed as a manufacturing refresh, it must be uninstalled. You require installation materials, and any maintenance levels that you applied, to restore IBM WebSphere MQ to its previous level.

The rules regarding the reversibility of an queue manager to run on a previous code level is dependent on the platform.

On UNIX, Linux, and Windows, changes in version or release level are not fully reversible, but changes in maintenance or fix level are reversible under certain conditions. If the maintenance level upgrade is applied as a normal fix pack, removing the fix pack restores the previous level. If the maintenance or fix level change is applied as a manufacturing refresh, the refresh must be removed. You can then reinstall the previous level, together with any fixes, to restore the previous level. Once the previous level of code has been restored, queue managers can be restarted, and continue with processing messages from when they were last stopped. If the maintenance level upgrade, no matter if it is a fix pack or a manufacturing refresh, contains new function, it introduces a new command level. A maintenance level upgrade is applied to a specific installation. If any queue manager associated with the installation enables the new command level, the maintenance level upgrade cannot be removed from the installation. Maintenance level upgrades applied later than the one including the new function can be removed.

An irreversible upgrade implies that you must back up the queue managers, or your system, to be able to restore your queue managers. Taking a backup of a queue manager requires you to stop the queue manager. If you do not take a backup, you are not able to restore IBM WebSphere MQ to its previous level. Any changes you make on the new level cannot be restored onto the backup system. Changes include

the creation or deletion of persistent messages, and changes to queue managers, channels, topics, and queues.

Note: The term upgrade does not imply that an IBM WebSphere MQ installation can be directly upgraded from one level to another. On some platforms, an upgrade requires that you remove the previous IBM WebSphere MQ installation. You can retain any queue managers that you have created.

Related concepts

"Upgrade, migration, and maintenance of IBM WebSphere MQ on UNIX, Linux, and Windows" on page 25
You can install new releases of IBM WebSphere MQ to upgrade IBM WebSphere MQ to a new maintenance, release, or version level. Multiple installations at the same or different levels can coexist on the same UNIX, Linux, and Windows server. You can apply maintenance level upgrades to upgrade the maintenance or fix level. Applying maintenance level upgrades cannot change the version or release level of IBM WebSphere MQ. Maintenance level upgrades can be reversed, installations cannot be reversed.

"New function in maintenance level upgrades" on page 24
On platforms other than z/OS, IBM might introduce new functions between releases in maintenance level upgrades such as fix packs. A maintenance level upgrade including new function increases the maximum command level of an installation. When you apply the maintenance level upgrade, the installation supports the new command level. A queue manager associated with the installation is not able to use the new function until you have run **strmqm** to set the queue manager to the new command level.

IBM WebSphere MQ migration

Migration is the conversion of programs and data to work with a new code level of IBM WebSphere MQ. Some types of migration are required, and some are optional. Queue manager migration is never required after applying a maintenance level update, that does not change the command level. Some types of migration are automatic, and some are manual. Queue manager migration is typically automatic and required after releases and manual and optional after a maintenance level upgrade that introduces a new function. Application migration is typically manual and optional.

Whenever you upgrade IBM WebSphere MQ to a new release that changes its command level, migration is performed by the queue manager. Whenever you upgrade IBM WebSphere MQ to a new maintenance or fix level, which introduces a new function using a new command level, you can migrate the queue manager to use the new command level and thereby the new function.

You must read "Changes that affect migration" on page 137 before upgrading your IBM WebSphere MQ installation or migrating your queue managers, to identify what migration tasks you must plan for.

Using the model in Figure 1 on page 24 you can distinguish different migration questions, which are discussed in the following topics:

Operating environment migration
: Upgrading the operating environment, or components in the environment such as installing a new level of JRE; see "IBM WebSphere MQ operating environment migration" on page 27

Queue manager migration
: Migrating a queue manager following an upgrade of the IBM WebSphere MQ installation to a new command level; see "Queue manager migration" on page 38.

IBM WebSphere MQ MQI client migration
: Migrating a client configuration following installation of a new version or release of the IBM WebSphere MQ MQI client; see "IBM WebSphere MQ MQI client migration" on page 40.

Application migration
: Relinking, recompiling, or recoding an IBM WebSphere MQ server or client application; see "Application migration and interoperation" on page 41. Application migration also includes migrating any API or channel exits

In addition, you must consider the impact of migrating one queue manager, or WebSphere MQ MQI client, on other clients or queue managers:

Compatibility, coexistence, and interoperability
See "Coexistence, compatibility, and interoperability" on page 28 for information about the compatibility of IBM WebSphere MQ applications connected to queue managers and IBM WebSphere MQ MQI clients on different command levels. The section also explains the concept of queue manager coexistence, and the interoperability of IBM WebSphere MQ JMS applications with WebSphere Application Server.

Queue manager clusters
Can a queue manager cluster contain queue managers at different command levels? See "Queue manager cluster migration" on page 42 to answer this question, and how to migrate a cluster of queue managers.

High-availability clusters
How do you migrate queue managers that are part of a high-availability cluster to a new command level, and maintain continuous and reliable service? See "Migrate a queue manager in a high availability configuration" on page 44, which covers both migration of multi-instance queue managers, and the migration of queue managers operating in high-availability clusters.

The remaining migration topics describe migration from other products or IBM WebSphere MQ features, to being part of a queue manager.

IBM WebSphere MQ publish/Subscribe broker
The version 6 publish/subscribe broker was separate from the queue manager. It used command messages to create and control publications and subscriptions. In migrating it to version 7, and integrating it with the queue manager, two major changes were introduced. The configuration and administration of publish/subscribe changed, and a new programming interfaced was introduced, integrated with the MQI. The first change requires any installation that used the version 6 publish/subscribe broker to run a migration command, **strmqbrk**. The second change is optional. You can modify existing or write new publish/subscribe programs to use the new programming interface. The changes are described in "Publish/Subscribe migration from Version 6.0" on page 46.

WebSphere Message Broker and WebSphere Event Broker Publish/Subscribe migration
The publish/subscribe broker in WebSphere Message Broker version 6.0 and 6.1, is replaced by using IBM WebSphere MQ as the publish/subscribe broker. WebSphere Event Broker version 6.0 is replaced by IBM WebSphere MQ. See "WebSphere Event Broker and WebSphere Message Broker migration tasks" on page 130.

WebSphere Message Broker SCADA migration to WebSphere MQ Telemetry
The SCADA nodes in WebSphere Message Broker version 6.0 are no longer supported in version 7.0. You can migrate your SCADA applications to use a combination of WebSphere Message Broker version 7.0 and WebSphere MQ Telemetry; see "Telemetry migration from WebSphere Message Broker" on page 51.

IBM WebSphere MQ migration concepts

Figure 1 on page 24 shows two runtime operating system environments. One environment is called `Server`, and contains an IBM WebSphere MQ server and server application. The other is called `Client`, and contains an IBM WebSphere MQ MQI client application. The server environment has one or more queue managers represented by **QM** using the installation of IBM WebSphere MQ installed on the server.

The queue manager labeled `QM-n?` coexists on the same server as `QM`, but runs at a different release level. Multiple releases of IBM WebSphere MQ installed in the same operating environment are called coexistent[2]. The IBM WebSphere MQ installations for different release levels are not shown. The question-mark in the queue manager name indicates this capability might not be present in your environment.

Queue manager coexistence is important for migration in two respects:

1. It can be used to reduce the risk involved in migrating to a new command level, and reduce the downtime during the migration process.

[2] It is not necessary, but it is usual, for coexistent installations to be at different release levels.

2. You must consider any configuration implications of running some applications or clusters on the same server with queue managers at different command levels.

For details, see “Queue manager coexistence in Version 7.5” on page 28

The queue manager, QM*, represents queue managers of various levels installed on other servers.

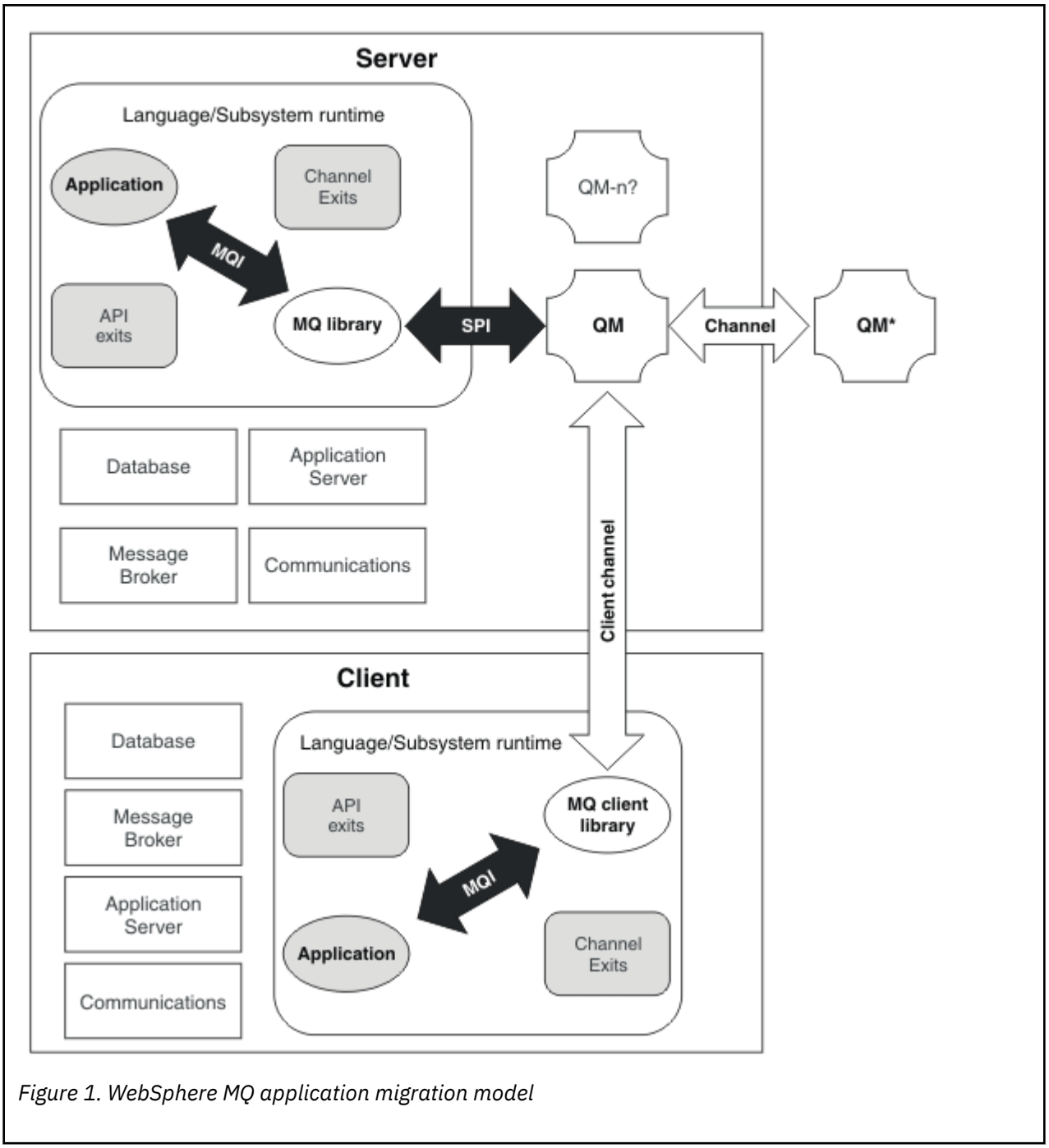


Figure 1. WebSphere MQ application migration model

New function in maintenance level upgrades

On platforms other than z/OS, IBM might introduce new functions between releases in maintenance level upgrades such as fix packs. A maintenance level upgrade including new function increases the maximum command level of an installation. When you apply the maintenance level upgrade, the installation supports the new command level. A queue manager associated with the installation is not able to use the new function until you have run **strmqm** to set the queue manager to the new command level.

A fix pack introducing a new function also includes regular fixes as a normal fix pack. If you do not want to use the function, but apply the fixes, do not enable the new function for any queue manager. The fix pack then is a fully reversible maintenance level upgrade. If you want to use a new function with a queue manager, you must change the command level of the queue manager. If one queue manager associated with an installation enables the new function, the fix pack introducing the new function and all previous fix packs cannot be removed anymore.

By applying a maintenance level upgrade that introduces a new function the IBM WebSphere MQ installation supports an additional command level. An installation can therefore support multiple command levels and multiple queue managers associated with it can run on different command levels.

To enable a new function for a queue manager, you must set the command level of queue manager to the command level supporting the new function. Because command levels are cumulative, the command level also supports all other functions introduced by previous maintenance level upgrades. By enabling a new function from a maintenance level upgrade you therefore enable all new functions from previous maintenance level upgrades, too.

Example

You have installed two fix packs including new functions:

1. Fix pack 7.1.0.2, introducing the command level 711.
2. Fix pack 7.1.0.15, introducing the command level 712.

You may have also installed other fix packs, but these did not introduce new function.

You want to use the new function from fix pack 7.1.0.15 on one of your queue managers. You enable command level 712 for that queue manager using the **strmqm** command. This enables all of the function introduced in both fix packs. It is not possible to use the function introduced by command level 712 without also enabling the function introduced by command level 711.

Once you have enabled command level 712 for a queue manager, the minimum version of product code to start the queue manager is 7.1.0.15. If you uninstall fix pack 7.1.0.15 from the installation with which the queue manager is associated, you will not be able to start the queue manager. In this situation, the **strmqm** command fails and issues error message AMQ7204 indicating that the queue manager has already been started by a newer release. If you have multiple installations of WebSphere MQ, you can start the queue manager using another installation which satisfies this minimum version requirement.

Configurable certificate validation policy on Windows and UNIX platforms

From fix pack 7.1.0.2 onwards, IBM WebSphere MQ can be configured to specify which SSL or TLS certificate validation policy is used to validate digital certificates received from remote partner systems.

If you need to ensure that your certificate validation is compliant with RFC 5280 for improved security in certificate validation, see Configuring certificate validation policies in WebSphere MQ for more formation.

See Certificate validation policies in WebSphere MQ for more information about IBM WebSphere MQ certificate validation policies.

Related tasks

“Migrating queue managers to new-function fix packs” on page 170

Upgrade, migration, and maintenance of IBM WebSphere MQ on UNIX, Linux, and Windows

You can install new releases of IBM WebSphere MQ to upgrade IBM WebSphere MQ to a new maintenance, release, or version level. Multiple installations at the same or different levels can coexist on the same UNIX, Linux, and Windows server. You can apply maintenance level upgrades to upgrade the maintenance or fix level. Applying maintenance level upgrades cannot change the version or release level of IBM WebSphere MQ. Maintenance level upgrades can be reversed, installations cannot be reversed.

On different platforms, you employ different mechanisms to install and maintain software releases. Installing a release at a new maintenance level, and applying maintenance level upgrades to upgrade an earlier release to the same maintenance level, have different results.

If you start a queue manager running on a later release level, then migration of the queue manager to that release level is required. The migration tasks you must perform to migrate from one release to another are documented in “Migrating IBM WebSphere MQ to a later version on UNIX, Linux, and Windows ” on page 83. A list of all the changes, by migration path, are listed in “Changes that affect migration” on page 137.

When you upgrade the maintenance or fix level of IBM WebSphere MQ by applying a regular maintenance level upgrade, you can reverse the upgrade by removing the fix. When you upgrade the maintenance or fix level of IBM WebSphere MQ by applying a maintenance level upgrade containing a new function, you can reverse that upgrade and all previously reversible upgrades until a queue manager associated with the installation enables the new function.

On UNIX, Linux, and Windows, you cannot easily revert to a previous level of IBM WebSphere MQ after installation. If you install a copy of IBM WebSphere MQ obtained from Passport Advantage or from physical media, the installer uninstalls IBM WebSphere MQ, if it is present. It then installs the new level of IBM WebSphere MQ. To revert to the previous level of IBM WebSphere MQ, you must keep the earlier installation image and any fixes you applied. Then you must uninstall the new level, reinstall the previous release level, and reapply the required fixes. If you have started any queue managers at the later level, they will not work with the restored level of IBM WebSphere MQ[3]. To restore IBM WebSphere MQ to its previous level, after starting any queue managers, you must first back up the queue managers. You can then restore the backup queue managers after restoring the previous level of IBM WebSphere MQ.

Maintenance levels and fix levels are both supplied from the service site, Fix Central. Fix central has a function to tell you what upgrades you can apply to the current level of your system. “Applying and removing maintenance level updates on UNIX, Linux, and Windows” on page 141 tells you how to apply the maintenance level upgrade and back it out. If you back out a maintenance level upgrade, it returns IBM WebSphere MQ code to the same level of code as before applying the maintenance level upgrade.

Characteristics of different types of upgrade

When you upgrade from one release to another, or apply maintenance refresh packs, fix packs, or interim fixes, the impact of the change depends on the extent of the change in VRMF level. The VRM codes are explained in “The version naming scheme for IBM WebSphere MQ on UNIX, Linux, and Windows” on page 8.

Table 5 on page 27 describes characteristics of different upgrade paths for IBM WebSphere MQ. In particular, notice that migration is required only if the version or release number changes. Other types of upgrade do not require migration, and are reversible if the upgrade is applied using a maintenance procedure.

[3] Unless you installed a later maintenance level upgrade, not a new release or version: then you could revert to an earlier maintenance level by reinstalling the earlier maintenance level upgrade. Queue manager data is compatible between maintenance levels.

Table 5. Types of upgrade on UNIX, Linux, and Windows

Upgrade type	Examples	Suggested testing	Type of installation	Manufacturing refresh	New features	IBM WebSphere MQ Migration
Version	6.0, 7.0	Full testing of all applications	Full or upgrade installation	Yes	Yes	Yes
Release	5.3					
Modification	7.0.1	Regression test critical applications	Full, or upgrade installation, or maintenance package		Minor	No
Enabled new-function fix pack	No example yet	Regression test critical applications	Maintenance package & queue manager migration	No		
No-function or not enabled new-function fix pack	6.0.1.1, 7.0.1.3	Brief test of critical applications	Maintenance package		No	
Interim fix	7.0.0.1-WS-MQ-Windows-LAIZ50784	Test affected applications	Manual			

Related concepts

"Queue manager coexistence in Version 7.5" on page 28
Queue managers, with different names, can coexist on any server as long as they use the same IBM WebSphere MQ installation. On UNIX, Linux, and Windows, different queue managers can coexist on the same server and be associated with different installations.

"Multi-installation queue manager coexistence on UNIX, Linux, and Windows" on page 31
You can install multiple copies of IBM WebSphere MQ for UNIX, Linux, and Windows on the same server. The installations must be at Version 7.1 or later, with one exception. One Version 7.0.1 installation, at fix pack level 6, or later, can coexist with multiple Version 7.1, or later installations.

"The version naming scheme for IBM WebSphere MQ on UNIX, Linux, and Windows" on page 8
On UNIX, Linux, and Windows, IBM WebSphere MQ releases have a four-digit Version, Release, Modification, and Fix (VRMF) level code. The version and release parts of the code are significant; they identify the service life of a release. To run a queue manager at a different VR level, you must migrate the queue manager, its applications, and the environment in which it runs. Depending on the migration path, the migration might require more or less effort.

Related information

Backing up and restoring a queue manager

IBM WebSphere MQ operating environment migration

You might perform some migration tasks for IBM WebSphere MQ as a result of upgrading the operating environment.

To find out what operating environment upgrades you must make before upgrading IBM WebSphere MQ, compare the requirements for different releases. For more information about system requirements, see System Requirements for IBM MQ.

If an operating environment change directly affects the migration to a new version of IBM WebSphere MQ, it is listed in "Changes that affect migration" on page 137.

The change might affect IBM WebSphere MQ migration indirectly. For example, the runtime linkage conventions for applications, or the way memory is allocated, might change.

Coexistence, compatibility, and interoperability

Coexistence is being able to install and run two or more versions of the same program on the same server. For WebSphere MQ, it normally means installing and running multiple versions of WebSphere MQ on a server. Compatibility is the ability to run applications from one level of queue manager with an earlier or previous level of the queue manager. It is also the ability to run client applications with different versions of the WebSphere MQ MQI client, and different levels of the queue manager. Interoperability is mainly the ability to exchange messages between different versions of WebSphere MQ. It can also mean the interoperability between others things, such as publish/subscribe brokers, or between components such as the WebSphere MQ classes for JMS and WebSphere Application Server.

Maintaining the compatibility, coexistence, and interoperability of WebSphere MQ is important in order to preserve the investment you make in applications and administrative procedures. Three areas to which this objective does not apply to as rigidly, are GUI interfaces, such as WebSphere MQ Explorer, information for service, such as FFST files and traces, and error messages. The text in an error message might change, to make the wording clearer or more accurate.

Queue manager coexistence in Version 7.5

Queue managers, with different names, can coexist on any server as long as they use the same IBM WebSphere MQ installation. On UNIX, Linux, and Windows, different queue managers can coexist on the same server and be associated with different installations.

Single installation queue manager coexistence on all platforms

Single installation queue manager coexistence is useful in development and production environments. In development environments, you can set up different queue manager configurations to support different development activities. You can also work with multiple queue manager configurations on a single server, connected by channels, as if deployed on a network.

In production environments configuring multiple queue manager on a single server is less common. It has no performance or functional advantage over a single queue manager configuration. Sometimes, you must deploy multiple queue managers on server. It might be essential to meet the requirements of a particular software stack, governance, administration, or as a consequence of the consolidation of servers.

Multi-installation queue manager coexistence

Multi-installation[4] queue manager coexistence has always been supported on z/OS. It is new in Version 7.1 on UNIX, Linux, and Windows. With multi-installation queue manager coexistence on the same server you can run queue managers at different commands levels on the same server. You can also run multiple queue managers at the same command level, but associate them with different installations.

Multi-installation adds more flexibility to the coexistence of queue managers using a single installation. Any of the reasons behind running multiple queue managers, such as supporting different software stacks, might require different versions of IBM WebSphere MQ.

[4] Do not confuse multi-installation queue manager coexistence with multi-instance queue managers. They are completely different, though they sound similar in English.

The biggest benefit of multi-installation identified by early users, is in upgrading from one version of IBM WebSphere MQ to another. Multi-installation makes upgrading less risky, less costly, and is more flexible in meeting the migration needs of applications running on a server.

The key to migration flexibility is being able to install a new version alongside an existing installation; see Figure 2 on page 29, which is extracted from “UNIX, Linux, and Windows: Side-by-side migration to a later version ” on page 65.

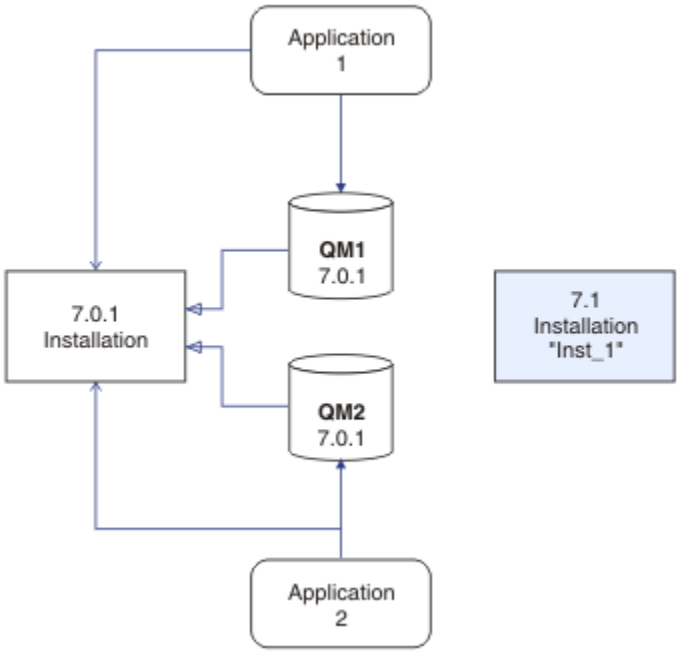


Figure 2. Side-by-side installation - step 2

When the installation is complete, and verified, migrate queue managers and applications to the new installation; see Figure 3 on page 29. When migration is complete, uninstall the old installation.

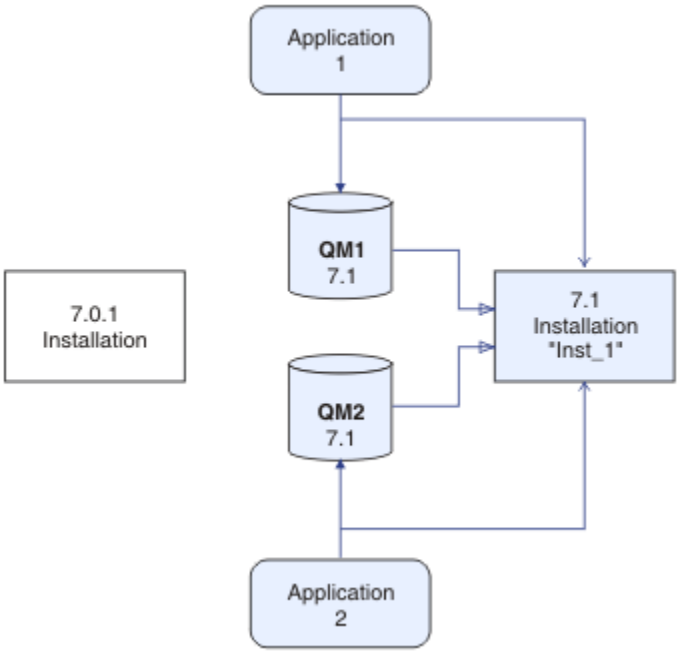


Figure 3. Side-by-side installation - step 4

Think of multi-installation as being the basis for a range of migration strategies. At one end is "Single-stage", in which you only have one installation on a server at a time. At the other end is multi-stage migration, in which you continue to run multiple installations at the same time. In the middle is side-by-side migration. Each of the three strategies is explained in these three tasks:

1. “UNIX, Linux, and Windows: Single-stage migration to a later version” on page 60
2. “UNIX, Linux, and Windows: Side-by-side migration to a later version ” on page 65
3. “UNIX, Linux, and Windows: Multi-stage migration to a later version” on page 72

Another similar use of multi-installation is to support the migration of queue managers to a new fix level; see Figure 4 on page 30. You maintain two installations, one of which has the latest fix pack applied, and the other has the previous maintenance levels. When you have moved all queue managers to the latest fix pack level, you can replace the previous fix pack with the next fix pack to be released. The configuration allows you to stage the migrating applications and queue managers to the latest fix pack level. You can switch the primary installation designation to the latest fix pack level.

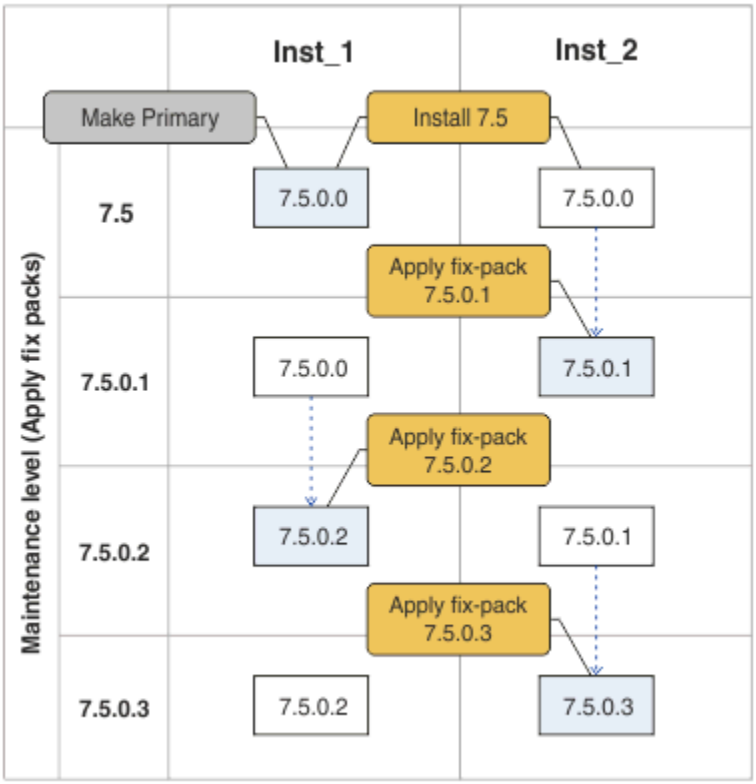


Figure 4. Rolling fix packs

Related concepts

“Multi-installation queue manager coexistence on UNIX, Linux, and Windows” on page 31
You can install multiple copies of IBM WebSphere MQ for UNIX, Linux, and Windows on the same server. The installations must be at Version 7.1 or later, with one exception. One Version 7.0.1 installation, at fix pack level 6, or later, can coexist with multiple Version 7.1, or later installations.

Related tasks

“Migrating IBM WebSphere MQ library loading from Version 7.0.1 to Version 7.5” on page 108
No change in the way IBM WebSphere MQ libraries are loaded is normally required if you upgrade from Version 7.0.1 to Version 7.5. You must have followed the instructions on building IBM WebSphere MQ applications in Version 7.0.1 and you must replace IBM WebSphere MQ Version 7.0.1 with IBM WebSphere MQ Version 7.5. If you choose to take advantage of multi-installation in IBM WebSphere MQ Version 7.5, based on the side-by-side or multi-stage migration scenarios, you must modify the environment for the operating system to resolve IBM WebSphere MQ dependencies for an application. Typically, you can modify the runtime environment, rather than relink the application.

“UNIX and Linux: Migrating IBM WebSphere MQ library loading from Version 7.0.1 to Version 7.5” on page 112
Investigate whether applications connecting to IBM WebSphere MQ for Windows Version 7.5 are linked to, and load libraries from, the correct installation.

“UNIX, Linux, and Windows: Staging maintenance fixes” on page 183
Use multiple installations of IBM WebSphere MQ on the same server to control the release of maintenance fixes.

“Windows: Migrating IBM WebSphere MQ library loading from Version 7.0.1 to Version 7.5” on page 110
Investigate whether applications connecting to IBM WebSphere MQ for Windows Version 7.5 are linked to, and load libraries from, the correct installation.

Multi-installation queue manager coexistence on UNIX, Linux, and Windows

You can install multiple copies of IBM WebSphere MQ for UNIX, Linux, and Windows on the same server. The installations must be at Version 7.1 or later, with one exception. One Version 7.0.1 installation, at fix pack level 6, or later, can coexist with multiple Version 7.1, or later installations.

Figure 5 on page 31 shows two IBM WebSphere MQ installations, two queue managers, and three applications. Applications 2 and 3 are connected to `QM2`, and application 1 is connected to `QM1`. Applications 1 and 3 load IBM WebSphere MQ libraries from the `Inst_1` installation, and application 2 loads libraries from the Version 7.0.1 installation.

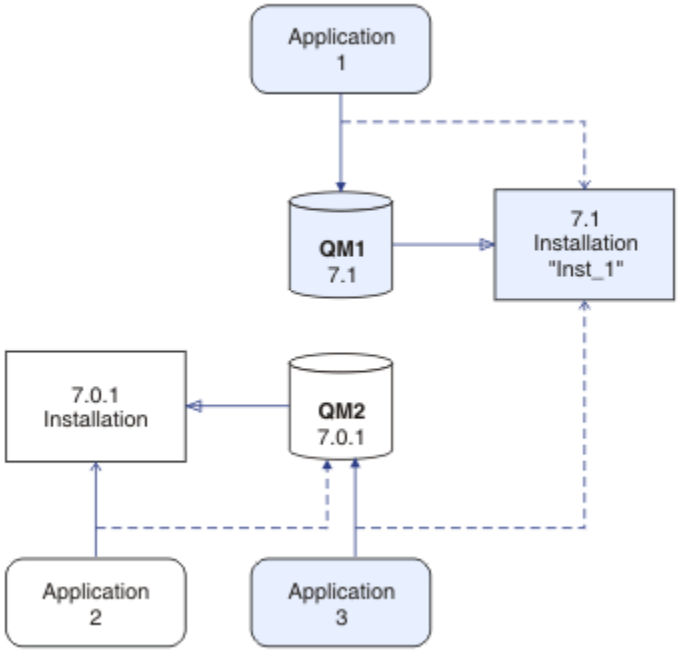


Figure 5. Coexistence of two queue managers using Version 7.0.1 and Version 7.1 installations

When you upgrade from Version 7.0.1 to Version 7.1, you can choose to run Version 7.0.1 alongside Version 7.1. The installation, illustrated in Figure 5 on page 31, is called a multi-version installation. You can also install multiple copies of Version 7.1 alongside each other. That would be called multi-installation. Multi-installation is the more general term.

Apart from on z/OS, Version 7.0.1 did not support multi-installation. Before Version 7.1 becoming available, fix pack 7.0.1.6 was shipped with some fixes to make Version 7.0.1 compatible with Version 7.1 on the same server. With 7.0.1.6 installed, you can run one copy of Version 7.0.1 alongside multiple copies of Version 7.1. You do not have to apply the fix pack to upgrade Version 7.0.1 to Version 7.1 "in-place"; see “UNIX, Linux, and Windows: Single-stage migration to a later version” on page 60.

A multi-version installation that includes Version 7.0.1, does not behave the same way as a multi-installation that does not. The differences primarily affect how you might choose to configure how

applications load IBM WebSphere MQ libraries, and run IBM WebSphere MQ commands. Because of these differences, think of the multi-version support provided in 7.0.1.6, as a migration aid to moving to a Version 7.1 multi-installation environment. The topics that explain the restrictions in Version 7.0.1 multi-version are listed in related links.

If you run multiple installations of IBM WebSphere MQ on a server you must consider three questions:

1. Which installation is a queue manager associated with; see “Queue manager association” on page 32?
2. Which installation does an application load; see “Loading IBM WebSphere MQ libraries” on page 32?
3. Which installation is a IBM WebSphere MQ command run from; see “Command association” on page 34?

Queue manager association

Before Version 7.1, queue managers on UNIX, Linux, or Windows were associated with the only installation on the server. With Version 7.1 installed on the same server as Version 7.0.1, you can change the association of a queue manager to Version 7.1 by running **setmqm**; see setmqm. You cannot change the association of a queue manager running a release of IBM WebSphere MQ earlier than Version 7.0.1 because you cannot install IBM WebSphere MQ Version 7.1 on a server with an installation of IBM WebSphere MQ earlier than Version 7.0.1.

A queue manager is permanently associated with an installation, until you choose to change the association with the **setmqm** command. You cannot associate a queue manager with an installation at a lower command level than the current command level of the queue manager.

In Figure 5 on page 31, QM1 is associated with Inst_1. The association is made by running `setmqm -m QM1 -n Inst_1`. When QM1 is first started, after running **setmqm**, if QM1 was running Version 7.0.1, it is migrated to Version 7.1. QM2 is associated with Version 7.0.1 because the association has not been changed.

Loading IBM WebSphere MQ libraries

The application connections to the queue managers are established by calling MQCONN or MQCONNX in the normal way.

Which IBM WebSphere MQ library an application loads depends on the configuration of the operating system loader and on the IBM WebSphere MQ installation the queue manager is associated with.

In Figure 5 on page 31, the operating system loads the IBM WebSphere MQ library from the Inst_1 installation for applications 1 and 3. It loads the IBM WebSphere MQ Version 7.0.1 library for application 2. The operating system has loaded the wrong library for application 3. Application 3 requires the IBM WebSphere MQ Version 7.0.1 libraries.

Figure 6 on page 33 shows what happens to application 3. Application 3 is connecting to QM2, and QM2 is associated with the IBM WebSphere MQ Version 7.0.1 installation. IBM WebSphere MQ detects that the operating system has loaded the wrong library to process calls from application 3 to QM2. IBM WebSphere MQ loads the correct library from the IBM WebSphere MQ Version 7.0.1 installation. It transfers the MQCONN or MQCONNX call to the IBM WebSphere MQ Version 7.0.1 library. Subsequent MQI calls that use the connection handle returned by MQCONN or MQCONNX, call entry points in the IBM WebSphere MQ Version 7.0.1 library.

Because IBM WebSphere MQ Version 7.0.1 libraries cannot load IBM WebSphere MQ libraries from other installations, there is no corresponding application in Figure 6 on page 33 that loads a IBM WebSphere MQ Version 7.0.1 library and connects to a queue manager running Version 7.1. If you attempt a connection to QM1 with application 2, IBM WebSphere MQ returns an error; see 2059 (080B) (RC2059): MQRC_Q_MGR_NOT_AVAILABLE .

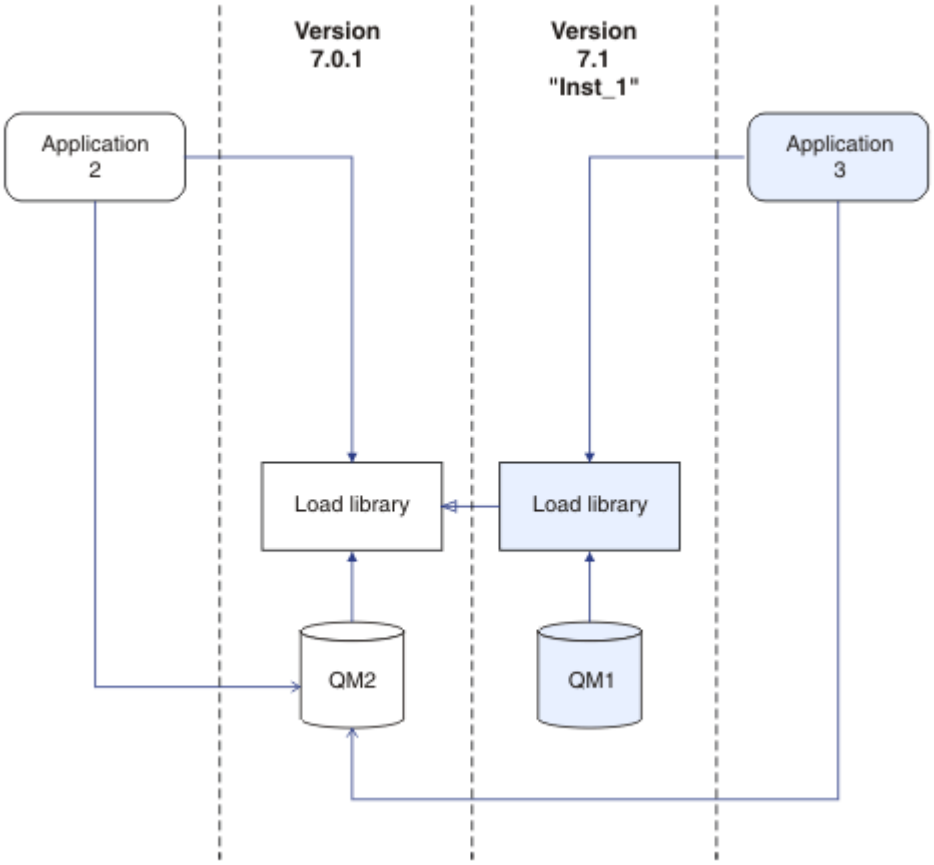


Figure 6. Loading calls in a different library

A Version 7.1 IBM WebSphere MQ library includes a routing capability that is based on the installation a queue manager is associated with. Earlier IBM WebSphere MQ libraries do not have a routing capability. The operating system can load a library from any Version 7.1 installation, or later, and IBM WebSphere MQ transfers MQI calls to the correct library.

The new loading capability of IBM WebSphere MQ libraries in Version 7.1 does not relax the restriction that an application compiled and linked at a later release level must not directly load a IBM WebSphere MQ library at an earlier release level. In practice the restriction is of less significance than in earlier releases, because as long as the operating system loads a library at the same or later level than the library the application was compiled and linked with, IBM WebSphere MQ can call any other level of IBM WebSphere MQ on the same server from Version 7.0.1 upwards.

For example, suppose you recompile and link an application that is to connect to a Version 7.0.1 queue manager using the libraries shipped with Version 7.1. At run time the operating system must load the Version 7.1 libraries for the application, even though the application connects to a Version 7.0.1 queue manager. IBM WebSphere MQ Version 7.1 detects the inconsistency and loads the Version 7.0.1 library for the application. The same applies to any future release. If the application is recompiled and linked against a later release, then the application must load an IBM WebSphere MQ library that matches the later release, even if it continues to connect to a Version 7.1 queue manager.

Your application might not be linked to an IBM WebSphere MQ library, but instead calls the operating system directly to load an IBM WebSphere MQ library. If the library that is loaded is from Version 7.1 or later, IBM WebSphere MQ checks the library is from the installation that is associated with the queue manager. If it is not, IBM WebSphere MQ loads the correct library.

Special migration considerations involving loading IBM WebSphere MQ libraries

You might have been asked to modify the installation of an earlier IBM WebSphere MQ release to satisfy the requirements of a build environment, or the IT standards in your organization. If you copied IBM WebSphere MQ libraries to other directories, or created symbolic links, you ended up with an unsupported configuration. The requirement to move IBM WebSphere MQ libraries to other directories was one of the reasons for changing the installation of IBM WebSphere MQ on UNIX and Linux. You can now install IBM WebSphere MQ into a directory of your choosing. You can also load IBM WebSphere MQ libraries from the `/usr/lib` directory, which is normally on the default load path on UNIX and Linux systems.

A common IT standard or build environment requirement is to include IBM WebSphere MQ libraries in the default load path on UNIX and Linux systems. IBM WebSphere MQ Version 7.1 has a solution. In Version 7.5 you can install IBM WebSphere MQ into a directory of your own choosing, and IBM WebSphere MQ can create symbolic links in `/usr` and its subdirectories. If you make a Version 7.1 installation primary by using the **setmqinst** command, IBM WebSphere MQ inserts symbolic links to the IBM WebSphere MQ libraries into `/usr/lib`. As a result, the operating system finds the IBM WebSphere MQ libraries in the default load path, if that includes `/usr/lib`.

Because IBM WebSphere MQ Version 7.1 libraries transfer calls to the correct installation, defining Version 7.1 installation as primary also results in the correct libraries being loaded for any application that is built with a link to `/usr/lib`, regardless of which queue manager it connects to. Unfortunately, this solution does not work if you have a Version 7.0.1 installation on the server, because then you cannot define a Version 7.1 installation as primary, and the Version 7.0.1 libraries do not load libraries from other installations. As an alternative to setting the Version 7.1 installation primary, use **setmqenv** with the `-k` or `-l` options to achieve a similar result.

You can find more information in Connecting applications in a multiple installation environment .

Command association

Examples of commands are **dspmqver**, **setmqinst**, **runmqsc**, and **strmqm**. The operating system must find a command in a IBM WebSphere MQ installation. Many commands also require a queue manager as an argument and assume the default queue manager, if a queue manager name is not provided as a parameter.

Unlike loading libraries, if a command includes a queue manager as a parameter, the command is not switched to the installation that is associated with the queue manager. You must use the **setmqenv** command to set up your environment correctly, so that any commands that you issue are run from the correct installation. You can provide a queue manager as a parameter to **setmqenv**, to set up the command environment for that queue manager; see Figure 7 on page 35.

On Windows, the **setmqinst** command sets global environment variables, and **setmqenv** local environment variables, including the PATH variable to find commands.

On UNIX and Linux, the **setmqinst** command copies symbolic links for a subset of the commands into `/usr/bin`; see External library and control command links to primary installation on UNIX and Linux. The **setmqenv** command sets up local environment variables, including the search path to the binary folder in the installation directory.

setmqenv must be on the search path in order to run it. One reason for having a Version 7.1 installation as primary is to be able to run setmqenv without having to configure the search path. If IBM WebSphere MQ Version 7.0.1 is installed on the server, no Version 7.1 installation can be primary and IBM WebSphere MQ Version 7.0.1 does not have a setmqenv command. The consequence is, you must provide a path to run the setmqenv command to set up the command environment for any of the Version 7.1 installations on the server.

Figure 7 on page 35 shows two examples of running **setmqenv** to set up the command environment for the copy of IBM WebSphere MQ that is associated with the queue manager, `QM1`.

IBM WebSphere MQ for Windows Version 7.1

```
"MQ_INSTALLATION_PATH\bin\setmqenv" -m QM1
```

IBM WebSphere MQ Version 7.1 for UNIX and Linux

```
. MQ_INSTALLATION_PATH/bin/setmqenv -m QM1
```

*Figure 7. Running **setmqenv***

Related tasks

“Migrating IBM WebSphere MQ library loading from Version 7.0.1 to Version 7.5” on page 108
No change in the way IBM WebSphere MQ libraries are loaded is normally required if you upgrade from Version 7.0.1 to Version 7.5. You must have followed the instructions on building IBM WebSphere MQ applications in Version 7.0.1 and you must replace IBM WebSphere MQ Version 7.0.1 with IBM WebSphere MQ Version 7.5. If you choose to take advantage of multi-installation in IBM WebSphere MQ Version 7.5, based on the side-by-side or multi-stage migration scenarios, you must modify the environment for the operating system to resolve IBM WebSphere MQ dependencies for an application. Typically, you can modify the runtime environment, rather than relink the application.

“UNIX, Linux, and Windows: Single-stage migration to a later version” on page 60
Single-stage migration is the term used to describe replacing the only installation of IBM WebSphere MQ on a server, with a later release. Single stage migration is also known as `upgrading in place` or `in place upgrade`. Until Version 7.0.1.6, single-stage was the only migration scenario. Single-stage migration preserves existing scripts and procedures for running IBM WebSphere MQ the most. With other migration scenarios you might change some scripts and procedures, but you can reduce the effect queue manager migration has on users.

“UNIX, Linux, and Windows: Side-by-side migration to a later version ” on page 65

“UNIX, Linux, and Windows: Multi-stage migration to a later version” on page 72

“UNIX, Linux, and Windows: Staging maintenance fixes” on page 183
Use multiple installations of IBM WebSphere MQ on the same server to control the release of maintenance fixes.

“UNIX and Linux: Migrating IBM WebSphere MQ library loading from Version 7.0.1 to Version 7.5” on page 112
Investigate whether applications connecting to IBM WebSphere MQ for Windows Version 7.5 are linked to, and load libraries from, the correct installation.

“Windows: Migrating IBM WebSphere MQ library loading from Version 7.0.1 to Version 7.5” on page 110
Investigate whether applications connecting to IBM WebSphere MQ for Windows Version 7.5 are linked to, and load libraries from, the correct installation.

Related reference

“Coexistence” on page 137
Queue managers, with different names, can coexist on any server as long as they use the same IBM WebSphere MQ installation. On UNIX, Linux, and Windows, different queue managers can coexist on the same server and be associated with different installations. In addition to queue managers coexisting on a server, objects, and commands must work correctly with different queue managers running at different command levels.

setmqenv

setmqinst

setmqm

strmqm -e CMDLEVEL

Related information

AIX linking 101

Changing the primary installation

Connecting applications in a multiple installation environment
Dynamic-Link Library Search Order
External library and control command links to primary installation on UNIX and Linux
Features that can be used only with the primary installation on Windows
Installation configuration file, mqinst.ini
Learn Linux, 101: Manage shared libraries

Application compatibility and interoperability with earlier versions of IBM WebSphere MQ

Connecting an application that is built against libraries shipped with a later version of IBM WebSphere MQ to an earlier version IBM WebSphere MQ is not supported. Avoid building applications against a later version, and redeploying them to a queue manager running at an earlier version, although some applications do work in practice.

IBM WebSphere MQ applications do interoperate with applications running on earlier versions of IBM WebSphere MQ, as long as they use no new function. IBM WebSphere MQ clients can connect to queue managers running at an earlier version than the client, as long as the client uses no new functions.

An IBM 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. It does not matter what version of IBM WebSphere MQ an application is built on and connected to. It can exchange messages with an application connected to an earlier version of IBM WebSphere MQ, as long as it does not use new function.

Consider these four cases; the first two cases are not supported though they might work in practice, the last two cases are supported. The first two cases require compatibility with an earlier version of IBM WebSphere MQ. The last two cases rely on the interoperability between all versions of IBM WebSphere MQ

1. Running an IBM WebSphere MQ server application, built with a later version of IBM WebSphere MQ, connecting to a queue manager running on a server with an earlier version of IBM WebSphere MQ installed.
2. Running an IBM WebSphere MQ client application, built with a later version of IBM WebSphere MQ, on a client platform with an earlier client installation, connecting to a queue manager running on a server with a later version of IBM WebSphere MQ installed.
3. Running an IBM WebSphere MQ client application, built with a later version of IBM WebSphere MQ, on a client platform with the later client installation, connecting to a queue manager running on a server with an earlier version of IBM WebSphere MQ installed.
4. Exchanging messages between an IBM WebSphere MQ client or server application, connected to a queue manager running on a server with a later version of IBM WebSphere MQ installed, with applications connected to a queue manager running on a server with an earlier version of IBM WebSphere MQ installed.

Plan to avoid the first two cases, as they are not guaranteed to work all the time. If you are running an incompatible configuration and you encounter a problem, you must rebuild your applications with the correct level of IBM WebSphere MQ. You can then continue with problem diagnosis.

Multi-installation and application loading

The new loading capability of IBM WebSphere MQ libraries in Version 7.1 does not relax the restriction that an application compiled and linked at a later release level must not directly load a IBM WebSphere MQ library at an earlier release level. In practice the restriction is of less significance than in earlier releases, because as long as the operating system loads a library at the same or later level than the library the application was compiled and linked with, IBM WebSphere MQ can call any other level of IBM WebSphere MQ on the same server from Version 7.0.1 upwards.

For example, suppose you recompile and link an application that is to connect to a Version 7.0.1 queue manager using the libraries shipped with Version 7.1. At run time the operating system must load the Version 7.1 libraries for the application, even though the application connects to a Version 7.0.1 queue

manager. IBM WebSphere MQ Version 7.1 detects the inconsistency and loads the Version 7.0.1 library for the application. The same applies to any future release. If the application is recompiled and linked against a later release, then the application must load an IBM WebSphere MQ library that matches the later release, even if it continues to connect to a Version 7.1 queue manager.

Examples

1. 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 answer is no, you must upgrade all the client workstations you deploy to, at least to the version of the client you have built. The queue managers running on earlier versions of IBM WebSphere MQ do not have to be upgraded. In practice all the clients are likely to work, but for maintainability you must avoid running incompatible levels of an application and the IBM WebSphere MQ client.
2. You deploy some IBM WebSphere MQ queue managers at a new version level. You have an existing IBM WebSphere MQ application that you use to send messages between the servers. Do you rebuild the application to deploy it onto the new servers? Can 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. IBM WebSphere MQ supports running the existing application on later servers and sending messages from later application versions to earlier ones. What you must not do is to rebuild the application on the later version and redeploy it onto both the earlier and newer servers. IBM WebSphere MQ does not support compatibility with earlier versions.

Application compatibility and interoperability with later versions of IBM WebSphere MQ

IBM WebSphere MQ applications run against later versions of a queue manager without recoding, recompiling, or relinking. You can connect an application that is built against libraries shipped with an earlier version of WebSphere MQ to a queue manager running at a later version of IBM WebSphere MQ.

If you upgrade a queue manager to a later version, existing applications built against its earlier version work without change. Exceptions are noted in “Changes that affect migration” on page 137. Likewise applications connected to the IBM WebSphere MQ Client, run against later versions of the client without recoding, recompiling, or relinking. You can deploy client applications built against earlier versions of the IBM WebSphere MQ Client libraries to connect using later versions of the libraries.

All the following four cases are supported. The first two cases rely on the compatibility of later version of WebSphere MQ with applications built against earlier versions. The last two cases rely on the interoperability between all versions of IBM WebSphere MQ.

You might change the operating environment as a prerequisite of migrating to a new level of queue manager. The operating environment changes, rather than changes in IBM WebSphere MQ itself, might require application change, recompilation, or relinking. Sometime the operating environment change affects only the development environment, and the operating environment supports applications built at an earlier level. In which case, you might be able to run existing applications built at the older level of the operating environment. You might not be able to build any new applications until the operating environment is upgraded.

In the future, after you have migrated queue managers and clients to the latest release level, consider changing your applications to take advantage of new capabilities.

Compatibility between different versions of a IBM WebSphere MQ MQI client and a queue manager

Any version and release of a WebSphere MQ MQI client can connect to any version and release of a WebSphere MQ queue manager. The MQI channel is automatically configured to the latest version that

both the client and server support. If the client and server are different versions, the client application must use only the functions in the earlier version.

The compatibility between clients and queue managers applies only to the version and release (`V.R`) of the product. The statement of compatibility does not necessarily apply to the modification and fix pack level (`M.F`) of the product.

If there are known problems at a specific `V.R.M.F` of the product, an upgrade to a more recent fix pack for the same `Version.Release` is necessary.

When you upgrade a queue manager to a different version, you automatically upgrade WebSphere MQ libraries. The libraries are used by WebSphere MQ MQI client and server applications running on the same server as the queue manager. To access new functions from remote clients, you must also upgrade the IBM WebSphere MQ MQI client installation on remote workstations. The WebSphere MQ MQI client includes the WebSphere MQ MQI client libraries.

Remote clients that have not been upgraded continue to work with an upgraded queue manager. The behavior of the client application might, in rare cases change. You must consult “Changes that affect migration” on page 137, to find out whether the changes from one version to another affect your client applications.

Remote clients that are connected to upgraded queue managers can use the new functions in the release. If an upgraded remote client is connected to a queue manager that has not been upgraded, it must not use new functions. In rare cases, the behavior of the client might change; see “Changes that affect migration” on page 137.

You can generally assume that upgrading the WebSphere MQ MQI client does not require you to recompile or relink the client application. You can also continue to use the same connection to the queue manager. If changes are required, they are identified in “Migrating IBM WebSphere MQ to a later version on UNIX, Linux, and Windows ” on page 83, for the particular migration path and platform you are concerned with.

The Client Channel Definition Table (CCDT) is an interface to customize the connection between an IBM WebSphere MQ Client and a queue manager. Entries in the tables are client connections, which are defined using a queue manager. The version of a CCDT is the version of the queue manager used to define the client connections. If an IBM WebSphere MQ MQI client uses CCDT to connect to a queue manager, the CCDT can be at a version greater than, less than, or equal to that of the client.

You can connect to a queue manager with an earlier WebSphere MQ Client or an earlier CCDT. If you are using a CCDT, and you plan to use new client channel configuration options, such as shared conversations, you must upgrade the CCDT, and therefore the WebSphere MQ MQI client installation to the new version.

Queue manager migration

After upgrading an installation, queue manager migration might be required. Migration takes place when you start a queue manager.

On UNIX, Linux, and Windows, queue manager migration is always required for changes in the first two digits of the VRMF code. Changes in the maintenance and fix level, M and F in the VRMF code, never cause automatic queue manager migration. No migration was required for the upgrade from Version 7.0 to Version 7.0.1. The change from Version 7.0 to Version 7.0.1 did change the command level from 700 to 701. From Version 7.1 onwards, a change in the command level always requires queue manager migration, but if the change is shipped in a maintenance or fix pack, you have the choice of whether to increase the command level, and cause queue manager migration.

Command level always increases with a change in version or release. If you decide to use new function introduced in a maintenance level upgrade, you must change the command level. The converse is not the case. You do not have to change the command level when the fix level changes. You can decide to install the fix pack, but not use the new function. Whether or not you use the new function, the installation of the fix pack increases the maximum command level supported by the installation. Run the **dspmqver** command to display the current maximum supported command level.

Queue manager migration is the process of converting persistent queue manager data from one version to another. Persistent queue manager data includes log files and data in the queue manager directory. The data records changes to objects such as messages, subscriptions, publications, queue managers, channels, queues, and topics.

Queue manager migration is required and largely automatic.

After migrating to a new release the queue manager cannot be restored to an earlier release level.

See “Migrating IBM WebSphere MQ to a later version on UNIX, Linux, and Windows ” on page 83 for any queue manager changes you must perform manually.

You can reduce the downtime and risk caused by queue manager migration, by verifying the new version first, using a different queue manager. Unless the platform supports queue manager coexistence, you need to perform the verification on a different server, or in a virtualized environment on the same server. If the platform you are upgrading supports queue manager coexistence, you can install the new version of IBM WebSphere MQ on the same server, verify it, and minimize downtime to the time required to stop, backup, and restart the queue manager.

Note: If you are migrating a queue manager through multiple release levels, one level at a time, you must start the queue manager after each upgrade to migrate it. You must also start all the channels, to ensure they are migrated. For example, if you are upgrading from version 5.3 to version 7.0.1, you might choose to upgrade to version 6, and then to version 7.0.1; see “Migration paths” on page 6.

Related concepts

“Migration paths” on page 6
Where can you find topics describing how to migrate WebSphere MQ from one release to another? Can you revert a migration to an earlier release? Can you migrate from an unsupported WebSphere MQ release to any later release directly?

“The version naming scheme for IBM WebSphere MQ on UNIX, Linux, and Windows” on page 8
On UNIX, Linux, and Windows, IBM WebSphere MQ releases have a four-digit Version, Release, Modification, and Fix (VRMF) level code. The version and release parts of the code are significant; they identify the service life of a release. To run a queue manager at a different VR level, you must migrate the queue manager, its applications, and the environment in which it runs. Depending on the migration path, the migration might require more or less effort.

“Upgrade, migration, and maintenance of IBM WebSphere MQ on UNIX, Linux, and Windows” on page 25
You can install new releases of IBM WebSphere MQ to upgrade IBM WebSphere MQ to a new maintenance, release, or version level. Multiple installations at the same or different levels can coexist on the same UNIX, Linux, and Windows server. You can apply maintenance level upgrades to upgrade the maintenance or fix level. Applying maintenance level upgrades cannot change the version or release level of IBM WebSphere MQ. Maintenance level upgrades can be reversed, installations cannot be reversed.

Related tasks

“Migrating a queue manager to a later version on UNIX, Linux, and Windows” on page 83
The procedures for migrating a queue manager from Version 7.1 to Version 7.5 are listed by platform.

Related reference

dspmqver

Reverting a queue manager to a previous version

On platforms other than z/OS it is not possible to reverse a migration. You can remove an upgrade before you have started a queue manager. After a queue manager has been started, if you remove the upgrade, the queue manager will not work.

On platforms other than z/OS, for which migration cannot be reversed, you must back up your system before starting migration. You can either back up queue manager data, or use a backup queue manager; see Backing up and restoring WebSphere MQ. To back up, you must stop the queue manager.

You can reduce the downtime and risk caused by queue manager migration, by verifying the new version first, using a different queue manager. Unless the platform supports queue manager coexistence, you need to perform the verification on a different server, or in a virtualized environment on the same server.

If the platform you are upgrading supports queue manager coexistence, you can install the new version of IBM WebSphere MQ on the same server, verify it, and minimize downtime to the time required to stop, backup, and restart the queue manager.

Related concepts

"Upgrade, migration, and maintenance of IBM WebSphere MQ on UNIX, Linux, and Windows" on page 25
You can install new releases of IBM WebSphere MQ to upgrade IBM WebSphere MQ to a new maintenance, release, or version level. Multiple installations at the same or different levels can coexist on the same UNIX, Linux, and Windows server. You can apply maintenance level upgrades to upgrade the maintenance or fix level. Applying maintenance level upgrades cannot change the version or release level of IBM WebSphere MQ. Maintenance level upgrades can be reversed, installations cannot be reversed.

Related tasks

Backing up and restoring WebSphere MQ

IBM WebSphere MQ MQI client migration

IBM WebSphere MQ MQI client migration is the process of converting IBM WebSphere MQ MQI client client configurations, and client and server channels from one version to another. Client migration can take place after upgrading the IBM WebSphere MQ MQI client. IBM WebSphere MQ MQI client migrations are reversible.

Client migration on the client workstation is optional and manual. Client migration on the server is required and automatic. See "Changes that affect migration" on page 137 for a list of any client changes. You must upgrade a IBM WebSphere MQ MQI client before migrating a client workstation to make use of new configuration options. You can make configuration changes to client and server connection channels on the server, but they have no effect on a client workstation, until the client is upgraded.

An example of client migration performed at the client workstation is to manually migrate configuration settings to the `mqclient.ini` configuration file.

An example of client migration at the server is the automatic conversion of server-connection channels between Version 6.0 and Version 7.1 to use shared conversations. You must upgrade the client to Version 7.1 to use shared conversations.

An example of combined client and server migration is the deployment of a new client connection definition table (CCDT). To use a new version of the CCDT, generate the table on a queue manager that is at the new code level. Deploy the table to clients that are going to use it. To deploy the table to a client, you first must update the client to at least the same level as the queue manager that created the table.

The IBM WebSphere MQ MQI client is interoperable with earlier and later versions of IBM WebSphere MQ. Upgrading the IBM WebSphere MQ MQI client makes new function available to client applications, and is important to maintain the service level. Migrating a IBM WebSphere MQ MQI client gives it access to new configuration options, such as shared conversations.

The IBM WebSphere MQ MQI client libraries, such as `mqic.dll`, are dynamic, and the application linkages to the libraries do not normally change. You do not relink a client application to pick up new IBM WebSphere MQ MQI client libraries. The client picks up the new library next time the library is loaded by the client application. Do not move libraries from their installed directory. Linking to libraries in anything other than their installed directory is an unsupported configuration.

Related concepts

"Application compatibility and interoperability with earlier versions of IBM WebSphere MQ" on page 36
"Application compatibility and interoperability with later versions of IBM WebSphere MQ" on page 37

IBM WebSphere MQ applications run against later versions of a queue manager without recoding, recompiling, or relinking. You can connect an application that is built against libraries shipped with an earlier version of WebSphere MQ to a queue manager running at a later version of IBM WebSphere MQ.

Application migration and interoperation

IBM WebSphere MQ supports running applications compiled and linked against previous versions of IBM WebSphere MQ with later levels of IBM WebSphere MQ.

To migrate an application to run with a new level of IBM WebSphere MQ, disconnect an application from the queue manager. Reconnect it when the queue manager is running again. However, it takes only one small difference in the interface between IBM WebSphere MQ and the application to break an application, or make it behave wrongly. Sometimes a problem does not show up for a long time. For this reason, you must always test your applications against a new version of IBM WebSphere MQ. The suggested extent of testing varies depending on the extent of the changes in IBM WebSphere MQ; see “Characteristics of different types of upgrade” on page 26.

Application migration refers to four kinds of changes.

1. Application changes that are consequent to upgrading the operating environment along with the queue manager. Rarely, linkage conventions change. The most likely reason for a linkage change is switching from 32 bit to a 64 bit environment. If you are using SSL or TLS you might have to relink with a new secure library.
2. Changes that you must make to the application in order to run an application against a new level of queue manager. Changes of this sort are uncommon. However, you must check “Changes that affect migration” on page 137 to see if any changes might affect your applications.
3. Changes that are not required, but you might want to make in the future, perhaps if there is some business reason to modify an application.
4. Changes to applications that are supplied by IBM, or other vendors, that require you to run migration utilities. The utilities convert the applications to running on the new version of IBM WebSphere MQ. See “Publish/Subscribe migration from Version 6.0” on page 46 and “Migration of the publish/subscribe broker in WebSphere Event Broker and WebSphere Message Broker” on page 48.

Do not load IBM WebSphere MQ libraries from an earlier level. IBM WebSphere MQ does not support connecting server applications loading libraries from the earlier level to connect to a later level of queue manager. On UNIX, Linux, and Windowsother platforms, the application load path must be set up to the location of the IBM WebSphere MQ server libraries. You do not have to recompile and relink an application. Applications compiled and linked against an earlier version of WebSphere MQ can load libraries from a later version.

On UNIX, Linux, and Windows, from Version 7.1 onwards, IBM WebSphere MQ loads the library from the installation the application is connecting to. An application must initially load a library of at least the same level as the application linked to. IBM WebSphere MQ then loads the correct version of the library from the installation that the queue manager is associated with. If you have two installations of the same version, but at different fix levels, IBM WebSphere MQ chooses which library to load. The choice is based on the queue manager the application is connected to. If an application is connected to multiple queue managers, it is possible that multiple libraries are loaded.

To help you write applications that can exchange messages with earlier versions of the product, IBM WebSphere MQ provides data type versioning. Data type versioning assists you in exchanging messages that are compatible with target queue managers. A good programming practice is to set the version number of a data structure explicitly. Do not assume that the default version is the one you require. By setting the version explicitly, you are forced to look up what version to use. The description of the data type version tells you what level of queue manager supports that version.

It is poor practice to set the data type version to the current version. If you recompile your program against a new version of IBM WebSphere MQ, the data type version might change with unexpected consequences.

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 to queue managers on different platforms. The default values of some WebSphere constants, such as `MQPMO_SYNCPOINT, MQPMO_NO_SYNCPOINT` differ between platforms. Some functions are not available on all platforms.

You must 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 IBM 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 functions programs use.

Related concepts

"Application compatibility and interoperability with earlier versions of IBM WebSphere MQ" on page 36

"Application compatibility and interoperability with later versions of IBM WebSphere MQ" on page 37
IBM WebSphere MQ applications run against later versions of a queue manager without recoding, recompiling, or relinking. You can connect an application that is built against libraries shipped with an earlier version of WebSphere MQ to a queue manager running at a later version of IBM WebSphere MQ.

"Migration paths" on page 6
Where can you find topics describing how to migrate WebSphere MQ from one release to another? Can you revert a migration to an earlier release? Can you migrate from an unsupported WebSphere MQ release to any later release directly?

Queue manager cluster migration

You can migrate queue managers in a cluster all at once, or one at a time, which is called a staged migration. Migrate full repository queue managers in a cluster before partial repository queue managers.

Cluster queue managers can participate in clusters with other queue managers running at different versions, which is why a staged migration is possible. Being able to stage a migration is important, as migrating each queue manager in a cluster takes time. By staging the migration, which leaves other queue managers that are in the cluster running, you reduce the effect of queue manager downtime on applications.

Migrate queue managers with full repositories first. Then migrate the other queue managers, which have partial repositories, one at a time. Complete migration of the entire cluster before starting to use new functions.

If you do have to start using new functions before completing migration of the entire cluster, you might need to refresh the partial repositories. After each migration of a queue manager with a partial repository, issue the **REFRESH CLUSTER** command on the newly migrated queue manager. The command updates the cluster records in the newly migrated queue manager, potentially receiving updates for any new attributes. Do not do this step if you migrated the entire cluster before using new function. The **REFRESH CLUSTER** command takes a long time for all the changes to work through the cluster.

Note: For large clusters, use of the **REFRESH CLUSTER** command can be disruptive to the cluster while it is in progress, and again at 27 day intervals thereafter when the cluster objects automatically send status updates to all interested queue managers. See Refreshing in a large cluster can affect performance and availability of the cluster.

If full repositories are not migrated before partial repositories, the cluster continues to work, but without all the new features in a version working as expected. To work predictably, the full repository queue managers must be at the new command level to be able to store information from the rest of the cluster that arises from using new features.

For example, the information might be a new channel attribute, such as shared conversations, which were introduced in Version 7.0. Information about the shared conversation attribute of a channel between two other Version 7.0.1 queue managers can be stored in a version Version 7.0 full repository, but not in a Version 6.0 repository. If information about a channel with the shared conversation attribute is updated from the version 6.0 full repository, the definition loses its shared conversation attribute. "How mixed version cluster repositories are updated" on page 43 explains how information is updated in a mixed-version cluster.

Note: If a queue manager is a member of a cluster, and is running at a release earlier than Version 6.0, you must migrate the queue manager to Version 6.0, Version 7.0, or Version 7.0.1, before migrating it to Version 7.5. You must start the queue manager after the first migration step, before proceeding to Version 7.5.

How mixed version cluster repositories are updated

Repositories store records for an object in a cluster in the version of the record format that matches the version of the queue manager hosting the repository. Repository queue managers forward object records, before they are stored, in the format that they are received in. The recipient ignores fields from a newer version, and uses default values for fields that are not present in the record.

Cluster repositories hold records that represent objects, for example, a queue record represents a cluster queue. A full repository holds records for all objects in the cluster. Partial repositories hold records for local objects and remote objects that are used locally. A repository record can hold information only about attributes at the same command level as the queue manager holding that repository. So for example, a Version 6.0 repository contains only Version 6.0 level attribute information. A Version 7.5 repository contains all Version 6.0 records, plus Version 7.5 records containing additional Version 7.5 attributes.

A repository stores a record it receives in its own version. If the record it receives is at a later version, the later version attributes are discarded when the record is stored. A Version 6.0 queue manager receiving information about a Version 7.5 queue manager stores only Version 6.0 information. A Version 7.5 repository receiving a version 6 record stores default values for attributes introduced in the version 7. The defaults define the values for the attributes that are not included in the record it receives.

A repository normally sends records in its own version format, which is the same as the format it has stored them in. There is one exception to this rule. When a full repository receives a record from a partial repository, it is immediately forwarded in the same format. So if a Version 6.0 full repository were to receive a record from a Version 7.5 partial repository, it would forward the Version 7.5 record. It sends the record to any other full repositories, and any other partial repositories that have subscriptions that match the record.

A partial repository reflects whichever full repository sent it the latest update to a record. As a consequence, you might see the information held by a Version 7.5 partial repository for new Version 7.5 attributes changing unexpectedly. The values might change from actual Version 7.5 information to default values. The changes occur if the full repositories in the cluster are at different levels. Migrate full repositories first to avoid instability.

A partial repository sends information about its objects to a full repository periodically at least once every 27 days. Information is sent about any object when it is altered or defined.

After migrating all full repositories to Version 7.5, some attributes might hold default values. The attributes might hold default values in place of actual values, if a repository has not received an update. You can refresh the repository in either of two ways:

- Alter the object which the record containing default values represents, for example, using ALTER QL for a local queue. The alteration forces the local repository to send the record again.
- Issue the **REFRESH CLUSTER** command on the partial repository which holds the record containing default values. **REFRESH CLUSTER** forces the partial repository to discard the record containing default values and get a new record as required.

  Note: For large clusters, use of the **REFRESH CLUSTER** command can be disruptive to the cluster while it is in progress, and again at 27 day intervals thereafter when the cluster objects automatically send status updates to all interested queue managers. See Refreshing in a large cluster can affect performance and availability of the cluster.

In summary, for the most predictable, and fastest migration, when you stage cluster migration do these steps in the following order:

1. Migrate the queue managers with full repositories.
2. Migrate the queue managers with partial repositories.
3. Start using new function in the cluster.

Related concepts

How long do the queue manager repositories retain information?

Migrate a queue manager in a high availability configuration

Follow standard procedures to migrate a queue manager that is part of a high availability configuration on platforms other than z/OS.

High availability configurations of queue managers can increase the availability of IBM WebSphere MQ applications. If a queue manager, or server fails, it is restarted automatically on another server. You can arrange for IBM WebSphere MQ MQI client applications to automatically reconnect to the queue manager. Server applications can be configured to start when the queue manager starts.

High availability configurations on platforms other than z/OS are implemented either by using a high availability cluster solution or by using multi-instance queue managers. Red Hat Cluster Suite or Microsoft Cluster Service (MSCS) are examples of high availability cluster solutions.

The overall principles involved in queue manager migration in a high availability configuration are the same, whether you are migrating a multi-instance queue manager or a high availability cluster. In either case, the principles are as follows:

1. You must not restart a queue manager at a lower command level than the one it was previously running.
2. You cannot upgrade the code an active queue manager is running.
3. You cannot back up an active queue manager.

Overall steps to migrate a queue manager in a multi-instance queue manager configuration

The following terms are relevant:

active queue manager instance
A queue manager instance that has been started permitting standby instances, and is running.

standby queue manager instance
A queue manager instance that has been started permitting standby instances, and is in standby. It is ready to take over from the active instance automatically.

Base your migration procedure on the following steps.

1. If a multi-instance configuration is using a common location for WebSphere MQ code, you must create a second location for the upgraded code.
2. Before you start the migration process, create a different queue manager on a server on which you have installed the upgrade. Test the upgrade by performing whatever verification checks that your organization requires.
3. If you have a pool of servers that you pick from when starting a queue manager instance, upgrade IBM WebSphere MQ on the servers that are in the pool and are neither active or acting as a standby.
4. Stop the standby queue manager instance. Make sure that you have no system management procedure running that restarts the instance automatically.
5. If you do not have a pool of servers, upgrade WebSphere MQ on the server that was running the standby instance.
6. Decide whether downtime or recoverability is more important in the migration:

   Follow these steps if recoverability is more important, and you must take a backup:

   a. Stop the active queue manager instance, without switching to any standby.
   b. Back up the queue manager.
   c. Start a queue manager instance, permitting standbys, on one of the upgraded servers.
   d. If you have a pool of upgraded servers, start another one, permitting standbys.

If availability is more important, follow this procedure; you do not take a backup.

a. Start a queue manager instance as a standby on one of the upgraded servers.

b. Stop the active queue manager instance, switching to the standby.

c. If you have a pool of upgraded servers, start another one, permitting standbys.

7. Upgrade the IBM WebSphere MQ code on the server that was the active queue manager instance, and start it as the standby instance if you have not already started a standby.

Overall steps to migrate a queue manager in a high availability cluster

The following terms are relevant:

active server
The running server or active queue manager instance

passive server
A server that is ready to take over from the active server automatically.

inactive server
A server that is not prepared to take over automatically. The server might have been removed from the cluster, or taken offline in some way.

Base your migration procedure on the following steps. The details depend on the specific commands in the cluster concerned.

1. Before you start the migration process, create a different queue manager on a server on which you have installed the upgrade. Test the upgrade by performing whatever verification checks that your organization requires.
2. If you have four servers available, you can form two cluster pairs.

   With two pairs, the queue manager can continue to run in a cluster-pair at the old command level. When you are ready, you can transfer the queue manager to the pair of servers at the new command level.
3. Remove a passive server from the cluster. Make sure that the cluster cannot automatically restart the server. The server is made inactive.
4. If a high availability cluster is using a common location for WebSphere MQ code, you must create a second location for the upgraded code.
5. Install, or upgrade, IBM WebSphere MQ code using the server that is not now running the queue manager.
6. Verify the upgrade by creating a different queue manager on the server, and performing whatever verification checks that your organization requires.
7. If more than half the servers remain in the cluster, remove a server, upgrade WebSphere MQ, and verify the upgrade. Each server is made inactive as part of the process. Continue until half the servers are upgraded.
8. If your active server is part of a remaining cluster, deactivate the passive servers so that the cluster cannot reactivate them automatically.
9. Decide whether downtime or recoverability is more important in the migration:

   Follow these steps if recoverability is more important:

   a. Stop the queue manager and remove the server from the cluster.

   b. Back up the queue manager.

   Or this step, if downtime is more important:

   a. Add the migrated servers back into the cluster, as passive servers. Then switch the remaining server in the high availability server cluster over to one of the passive servers. The switch causes the running queue manager to stop, restarts it on one of the passive servers.
10. Upgrade any remaining high availability servers, and add them back into the cluster.

Related tasks

"Windows: Migrating an MSCS configuration" on page 120
Migrate queue managers in MSCS configuration one node at a time, following these instructions.

Publish/Subscribe migration from Version 6.0

The publish/subscribe programming interface is integrated into the MQI. It is sometimes known as integrated publish/subscribe to distinguish it from queued publish/subscribe. Queued publish/subscribe is name given to the implementation of the Version 6.0 publish/subscribe command interface. You can use both queued and integrated publish/subscribe, and use them together with the same topics and subscriptions.

Connections between the MQTT client and the queue manager are always initiated by the MQTT client. The MQTT client is always the SSL client. Client authentication of the server and server authentication of the MQTT client are both optional.

In IBM WebSphere MQ Version 6.0, applications perform publish and subscribe operations by placing special request messages on certain queues. The IBM WebSphere MQ Version 6.0 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. The broker is started and stopped independently from the queue manager.

In later versions of IBM WebSphere MQ, integrated publish/subscribe applications do not communicate with the broker in order to publish or subscribe. They use the new publish/subscribe API directly. The MQI call, 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 IBM WebSphere MQ Version 6.0 to a later version, the publish/subscribe broker is not upgraded. State information must be migrated from the IBM WebSphere MQ publish/subscribe broker into the queue manager. Data that is migrated includes subscriptions, retained publications, hierarchy relations, and authorities. You migrate publish/subscribe resources from the broker to the queue manager by using the **strmqbrk** command, which previously started the publish/subscribe broker.

JMS can use either the queued or the integrated publish/subscribe interface. In most circumstances, JMS defaults to using the integrated interface. You can control which interface it uses by setting the JMS property **PROVIDERVERSION** to 6 or 7.

Publish/subscribe hierarchies and clusters

IBM WebSphere MQ Version 6.0 publish/subscribe brokers are connected into hierarchies so that publications and subscriptions can flow between them. After migrating, using **strmqbrk**, these hierarchies continue to function in later versions. IBM WebSphere MQ also contains a new method of allowing publications and subscriptions flow between queue managers; publish/subscribe clusters. To migrate to a publish/subscribe cluster, first migrate the queue managers in the hierarchy to a later version using **strmqbrk**. After checking publish/subscribe works, convert to using a cluster by creating cluster topics and altering the parent/child relations.

Publish/subscribe clusters are the same as queue manager clusters. You can use a queue manager cluster for publish/subscribe and to route point-to-point messages. You might decide to create separate publish/subscribe clusters, with different queue managers, because you want to manage the clusters separately.

New publish/subscribe applications are more likely to use clusters than hierarchies. What advantages do clusters have over hierarchies that might prompt you to convert from a hierarchy to a cluster?

Robustness

The failure of a queue manager can affect the transmission of publications and subscriptions between other queue managers in the hierarchy. In a cluster, every queue manager is connected to every other queue manager. A failure in one queue manager does not affect the connection between two other queue managers.

Simpler channel administration
A hierarchy uses manually configured channel connections between queue managers in the hierarchy. You must maintain these connections, adding, and removing channels as queue managers are added and removed from the hierarchy. In a publish/subscribe cluster, queue managers are connected by automatically maintained cluster connections.

Ease of use
Cluster publication topics, subscriptions, and their attributes are replicated to every member of a cluster. You can list and modify cluster topics and subscriptions attributes using the IBM WebSphere MQ Explorer. Your changes are replicated to other members of the cluster.

Consistency
In general, you must not connect the same queue managers together with hierarchies and clusters. If you have decided that new publish/subscribe applications are to use publish/subscribe clusters, avoid queue managers being part of both a hierarchy and a cluster. You must convert existing hierarchies to clusters.

The principal reasons for continuing to use a hierarchy, despite the advantages of publish/subscribe clusters, are threefold:

1. **strmqbrk** performs the migration to a hierarchy automatically. To convert the hierarchy to a cluster, you must do several manual tasks in addition to running **strmqbrk**.
2. You do not need to complete the migration of the whole hierarchy, before resuming your publish/subscribe applications using a hierarchy. In contrast, the whole hierarchy must be converted to a cluster before resuming your publish/subscribe applications using a cluster.
3. You might want to compare the performance of publish/subscribe in a cluster to a hierarchy, before switching your production systems to using publish/subscribe clusters.

Treat the conversion of a Version 6.0 publish/subscribe hierarchy to using clusters as a two-stage process. First, migrate all the queue managers in the hierarchy, and then convert the hierarchy to a cluster.

Publish/subscribe application migration

Publish/subscribe application migration from Version 6.0 to use integrated publish/subscribe in later versions is not necessary. Queued publish/subscribe continues to work alongside integrated publish subscribe. It can work in two ways. Set the **PSMODE** queue manager attribute to ENABLED for the queue manager to process queued publish subscribe command messages. If you set **PSMODE** to COMPAT, the queue manager runs integrated publish/subscribe, but not queued publish/subscribe. Another broker, such as WebSphere Message Broker, can process the queued publish/subscribe command messages while connected to the same queue manager.

Note: Publications and subscriptions cannot be exchanged between WebSphere Message Broker version 6.0 or 6.1, or WebSphere Event Broker version 6.0 and versions of IBM WebSphere MQ later than Version 6.0. Messages can be exchanged.

Queued publish/subscribe coexists on the same queue manager as integrated publish/subscribe, sharing publish/subscribe topics. Queued and integrated publish/subscribe also interoperate, exchanging subscriptions and publications between different queue managers.

When faced with a requirement to create another publish/subscribe application, or extend an existing application, which interface must you choose? You can choose either interface. The main reasons for continuing to develop a Version 6.0 application using the queued interface are as follows:

1. Continuing requirement to run the application on Version 6.0 queue managers.
2. Continuing requirement to run the application on version 6.0 or 6.1 WebSphere Message Broker or WebSphere Event Broker.
3. A small enhancement is required, and it is cheaper to extend the existing application than to migrate the existing application to use the integrated publish/subscribe interface, or to develop a new application to implement the enhancement.

4. The differences between the behavior of integrated and queued publish/subscribe mean that migration is a costly option.

The main reasons for developing new applications, and migrating existing applications to call the integrated publish/subscribe interface are as follows:

1. You find that the integrated publish/subscribe interface is easier to understand, and applications are quicker to develop.
2. You want to take advantage of clustering.

If you want to convert queued publish/subscribe commands into integrated publish/subscribe calls, note that the publish/subscribe interfaces have different concepts of a publisher, subscriber, the identity of a subscription and publication, and the nature of streams, topics, and access control. Consequently, a straightforward mapping changing your application from creating a command message to calling a publish/subscribe function, works only for the simplest of applications.

You can combine queue publish/subscribe command messages and integrated publish/subscribe calls in the same application on the same queue manager connection. However, because the publication and subscription models are different, matching commands and calls do not necessarily produce the same results.

Migration of the publish/subscribe broker in WebSphere Event Broker and WebSphere Message Broker

The publish/subscribe broker used in WebSphere Message Broker version 6.0 and 6.1 is replaced by the integrated publish/subscribe broker in IBM WebSphere MQ. IBM WebSphere MQ replaces WebSphere Event Broker version 6.0. The migration to IBM WebSphere MQ has two parts. The migration of publish/subscribe configuration data, and conversion of publish/subscribe collectives to publish/subscribe clusters.

Note: IBM WebSphere MQ does not support the content-based filtering provided in WebSphere Event Broker version 6.0. If you use this function and intend to continue to do so, you must migrate to WebSphere Message Broker version 7.0.

Chapter 9 of the IBM Redbook, contains a worked example of migrating the publish/subscribe broker to IBM WebSphere MQ; see the related information.

The migration of the publish/subscribe configuration is carried out by **migmqbrk**.

The programs migrate the following publish/subscribe configuration data:

- Subscriptions
- Subscription points. Subscription points are supported only when MQRFH2 messages are used.
- Streams
- Retained publications

The task, “Migrating publish/subscribe configuration data from WebSphere Event Broker or WebSphere Message Broker version 6.0 or 6.1” on page 131, guides you through the migration of publish/subscribe configuration data. The migration is in three phases; see “Migration phases during the migration of WebSphere Message Broker or WebSphere Event Broker” on page 49. It is partially automatic. You must set up access control on IBM WebSphere MQ to reproduce the security environment on the broker. The migration procedure creates security commands for you to review and run to set up the access control on IBM WebSphere MQ.

A WebSphere Message Broker and WebSphere Event Broker organize networks of publish/subscribe brokers that exchange publications and subscriptions. In IBM WebSphere MQ, the matching topology is a publish/subscribe cluster. If you have defined publish/subscribe collectives, follow the task “Migrating a WebSphere Message or Event Broker publish/subscribe collective to a WebSphere MQ publish/subscribe cluster” on page 133.

Note:

You must migrate all the Message or Event Brokers in a collective to an IBM WebSphere MQ cluster at the same time. Any Message or Event Brokers that remain are not able to connect to the migrated brokers.

The WebSphere Event Broker or WebSphere Message Broker publish/subscribe information is not deleted by the migration process. This information is stored in the subscription database tables, and is available to use until you explicitly delete it.

Related information

IBM Redbook: Connecting Your Business Using IBM WebSphere Message Broker V7 as an ESB

Migration phases during the migration of WebSphere Message Broker or WebSphere Event Broker

The migration of the publish/subscribe broker from WebSphere Message Broker or WebSphere Event Broker to WebSphere MQ has three phases. The rehearsal phase, the initial phase, and the completion phase.

The rehearsal phase

This phase creates a migration log, reporting any errors that might be found, but does not change the current configurations. You can use this phase to observe what the result of a real migration would be. Rehearsing the migration also produces a file containing security commands. The commands are suggestions to guide you setting up a security environment in the queue manager. The commands define a security environment for the queue manager that is equivalent to the one that existed in the broker. Review the security command file and determine the actions that you need.

The initial phase

This phase creates topic objects that might be needed in the queue manager. The topic objects are based on the Access Control List (ACL) entries that are defined in the broker. You must run this phase before you run the completion phase. The initial phase also produces a file containing security commands. The topic objects are created in anticipation of you running the security commands to create ACLs for the topic objects. Before you run the completion phase, you must review and modify the security command file as required and run the commands that you need. The initial phase also creates a migration log.

Note: If the migration process attempts to create a topic object, it first checks whether a suitable topic object exists in the queue manager. If it does, it uses that existing topic object. The check ensures that if the migration process is run multiple times, it does not attempt to create multiple topic objects for the same purpose.

If you modify the properties of one of the generated topic objects the migration process does not take account of the modification. For example, you might want to alter the wildcard property. The migration process uses the modified topic object, even though it no longer has the same properties that the migration process originally created for it. For this reason, with one exception, you must leave unchanged the topic objects that are created by the migration process until it has completed. The exception is you can change access permissions between the initial and completion phases,

The completion phase

The completion phase retrieves the current publish/subscribe definitions from the broker. It uses the definitions to create publish/subscribe definitions in the queue manager that is associated with the named broker. When the migration is complete, the queue manager publish/subscribe configuration is equivalent to the broker publish/subscribe configuration. The completion phase also creates a migration log containing the details of the migration.

The WebSphere Event Broker or WebSphere Message Broker publish/subscribe information is not deleted by the migration process. This information is stored in the subscription database tables, and is available to use until you explicitly delete it.

Publish/Subscribe Broker: Access Control List (ACL) migration

An automated process migrates the publish/subscribe broker from WebSphere Event Broker and WebSphere Message Broker to IBM WebSphere MQ. The process does not migrate the ACLs automatically. The migration task provides two methods to migrate the ACL to WebSphere MQ.

The default ACL in WebSphere Event Broker and WebSphere Message Broker gives all user IDs access to any topic. In IBM WebSphere MQ, the default access control is for no user ID to have access to any topic. In the brokers, you can create ACL entries that deny access to topics. In IBM WebSphere MQ, you can permit access only, you cannot deny access. Because of the differences in access control, the automated migration process cannot migrate ACLs from the brokers to IBM WebSphere MQ.

The migration process solves this problem by generating a security command file. The file is produced by the rehearsal and initial phases of the migration process. It is a best attempt at creating security commands. The commands, when run, create similar access control for topics in the IBM WebSphere MQ queue manager, as in the ACLs in the brokers.

If the migration process finds an ACL entry that denies access, it cannot produce a IBM WebSphere MQ equivalent command. Instead, it reports it in the security command file and advises that the ACL migration must be performed manually. You must set up a security environment in the queue manager, equivalent to the one that existed in the broker. You must set up the environment in the queue manager before you run the completion phase of the migration.

To create an equivalent security environment in the queue manager, you can take one of two approaches. You can either modify the broker security settings to match the IBM WebSphere MQ security approach, or modify the security command file as needed. If you decide to modify the broker security settings, run the rehearsal phase of the migration again to produce a new security command file. Continue to run the rehearsal phase until it stops reporting that the ACL migration must be performed manually.

IBM WebSphere MQ Telemetry migration from Version 7.0.1

IBM WebSphere MQ Telemetry Version 7.0.1 is a separate feature that you install with IBM WebSphere MQ Version 7.0.1. Telemetry is integrated into IBM WebSphere MQ Version 7.5.

Existing IBM WebSphere MQ Telemetry applications are unaffected by the change. Telemetry channels are migrated to Version 7.5 when a queue manager is migrated to Version 7.5.

You can now administer IBM WebSphere MQ Telemetry from the command line and from applications, in addition to continuing to administer telemetry from IBM WebSphere MQ Explorer.

You can administer IBM WebSphere MQ Telemetry Version 7.0.1 only from the Version 7.0.1 IBM WebSphere MQ Explorer. If you connect the Version 7.5 explorer remotely to a Version 7.0.1 queue manager, no telemetry resources are displayed. You cannot connect a Version 7.5 IBM WebSphere MQ Explorer locally to a Version 7.0.1 queue manager on the same server.

With multiple installations, you can run IBM WebSphere MQ Telemetry Version 7.0.1 with IBM WebSphere MQ Version 7.0.1 on the same server as the integrated telemetry in Version 7.5.

To upgrade IBM WebSphere MQ Telemetry Version 7.0.1 from Version 7.5, you must uninstall IBM WebSphere MQ Telemetry from the Version 7.0.1 installation. You can uninstall IBM WebSphere MQ Telemetry Version 7.0.1 before or after you uninstall or upgrade IBM WebSphere MQ Version 7.0.1. Do not install the Version 7.5 telemetry component of the IBM WebSphere MQ Version 7.5 installation until you have uninstalled IBM WebSphere MQ Telemetry Version 7.0.1.

Related concepts

“Migrating IBM WebSphere MQ Telemetry” on page 80
Migrate IBM WebSphere MQ Telemetry from Version 7.0.1 to Version 7.5 by completing the tasks in this section. You must stop all IBM WebSphere MQ activity on the system before migrating.

Introduction to WebSphere MQ Telemetry

Related tasks

Administering WebSphere MQ Telemetry

Telemetry migration from WebSphere Message Broker

The Supervisory Control and Data Acquisition (SCADA) input and output nodes on WebSphere Message Broker version 6.0, have been removed in version 7.0. The SCADA nodes are an interface to the MQ Telemetry Transport (MQTT) protocol, which is used to connect small devices and small message brokers over TCP/IP. To migrate applications that use the SCADA nodes, you must install WebSphere MQ Telemetry and change your message flows. You must also migrate any MQTT client applications to use version 3 of the MQTT protocol, if they are not already at version 3. You might be using a small message broker with your MQTT applications to connect to WebSphere Message Broker. In which case, you must migrate the small message broker to the WebSphere MQ Telemetry daemon for devices.

MQTT is a publish/subscribe protocol. WebSphere MQ now acts as a publish/subscribe broker for MQTT clients, including the WebSphere MQ Telemetry daemon for devices. The WebSphere MQ Telemetry daemon for devices is both a client and a server. To connect MQTT clients to WebSphere MQ, you must install and configure WebSphere MQ Telemetry. WebSphere MQ Telemetry is a WebSphere MQ service that provides a TCP/IP connection between an MQTT client and a queue manager. To configure WebSphere MQ Telemetry you use the WebSphere MQ Explorer.

To reconnect MQTT clients to the message flows that used SCADA nodes in WebSphere Message Broker version 6, you must replace the SCADA nodes. One approach is to replace the SCADA nodes with `JMSInput` and `JMSOutput` nodes.

On the `JMSInput` node, you can provide a subscription topic. Now that WebSphere Message Broker uses WebSphere MQ as its publish/subscribe broker, the subscription is managed by a WebSphere MQ queue manager. When an MQTT client publishes to a topic, WebSphere MQ Telemetry passes the publication to the queue manager the MQTT client is connected to. You might connect the MQTT client to the same queue manager as the broker. Alternatively, you can connect the MQTT client to a different queue manager, and connect both queue managers together in a publish/subscribe cluster. Whichever topology you choose, if the MQTT publication topic matches the `JMSInput` node subscription topic, the publication is delivered to the WebSphere Message Broker application.

To reverse the flow of a publication, from the WebSphere Message Broker to the MQTT client, use a `JMSOutput` node, or a `Publication` node. On the `JMSOutput` node, define the publication topic. A message output by the `JMSOutput` node is published on that topic to the WebSphere MQ queue manager the broker is connected to. If the queue manager has a matching subscription created by an MQTT client, the publication is forwarded to the MQTT client. If the MQTT client is connected to a different queue manager, the publication is forwarded using a publish/subscribe cluster.

Related concepts

Telemetry

Related tasks

Migration of telemetry applications from using WebSphere Message Broker v6 to use WebSphere MQ Telemetry and WebSphere Message Broker v7

Related information

MQTT.org

Migration planning on UNIX, Linux, and Windows

Before migrating from one version to another, read the planning topic for the platform you are interested in. Create your own migration plan based on the outline in the planning topic.

If there are concepts about migration you do not understand, read “Introduction to IBM WebSphere MQ migration” on page 5 first.

AIX: Planning to migrate to a later version

Create a migration plan to migrate from your current version of IBM WebSphere MQ to a later version. You must back up your queue manager data before you migrate to the later version. You cannot revert a queue manager to the earlier version of the product once you start it on the later version.

Before you begin

If the queue manager is currently not on Version 7.1, see Table 1 on page 7 in "Where to find a topic about a specific migration path" on page 7.

About this task

Use the following steps as a guide to creating a migration plan.

Procedure

1. Review the WebSphere MQ system requirements for Version 7.5.

   See System Requirements for IBM WebSphere MQ.
2. Decide whether to run Version 7.1 and Version 7.5 on the same server.
3. Review all the changes in WebSphere MQ that affect you.

   See changes from "Changes in IBM WebSphere MQ Version 7.5" on page 138.
4. Review performance changes.

   Performance reports are published as Supportpacs; see WebSphere MQ - SupportPacs by Product.
5. Review the latest README file for the product you are working with.

   See WebSphere MQ and MQSeries product READMEs.
6. Plan the sequence and timing of queue manager upgrades.

   If the queue manager is part of a queue manager cluster, you must migrate the queue managers that are full repositories first.
   If the queue manager is part of a high availability cluster, plan the migration to minimize downtime and maximize availability; see "Migrate a queue manager in a high availability configuration" on page 44.
7. Plan to migrate your queue manager to Version 7.5.

   See "AIX: Migrating a queue manager to a later version" on page 84.
   Backing up queue manager data is part of the queue manager migration task.
   An alternative approach to backing up queue manager data, is to install and configure a new server. Test Version 7.5 with a new queue manager on the new server. When you are ready to go into production on Version 7.5 copy the queue manager configuration and data, to the new server.
8. Plan to update any manual or automated procedures you have written with changes to messages and codes.
9. Plan to update applications that might be affected by changes you identified in "Changes in IBM WebSphere MQ Version 7.5" on page 138.

   Consider whether the application must be able to run on both Version 7.1 and Version 7.5. You might be able to change the application to be compatible with both code levels. If you cannot, you can query the queue manager command level, and make the code conditional on the command level. Call `MQINQ` setting the `MQIA_COMMAND_LEVEL` selector.
10. Decide on what regression tests to perform before putting the queue manager into production on Version 7.5.

    Include the procedures and applications you identified in steps 6 and 7 in your regression tests.
11. Plan to upgrade your WebSphere MQ MQI client installations to Version 7.5.

12. Plan to upgrade your client and server applications to use new functions in Version 7.5.

Directory structure on UNIX and Linux systems

Changing configuration information on Windows, UNIX and Linux systems

Fix Central

IBM Passport Advantage

“Migrate a queue manager in a high availability configuration” on page 44
Follow standard procedures to migrate a queue manager that is part of a high availability configuration on platforms other than z/OS.

“Queue manager cluster migration” on page 42
You can migrate queue managers in a cluster all at once, or one at a time, which is called a staged migration. Migrate full repository queue managers in a cluster before partial repository queue managers.

“Reverting a queue manager to a previous version” on page 39
On platforms other than z/OS it is not possible to reverse a migration. You can remove an upgrade before you have started a queue manager. After a queue manager has been started, if you remove the upgrade, the queue manager will not work.

WebSphere MQ - SupportPacs by Product

“Upgrade, migration, and maintenance of IBM WebSphere MQ on UNIX, Linux, and Windows” on page 25
You can install new releases of IBM WebSphere MQ to upgrade IBM WebSphere MQ to a new maintenance, release, or version level. Multiple installations at the same or different levels can coexist on the same UNIX, Linux, and Windows server. You can apply maintenance level upgrades to upgrade the maintenance or fix level. Applying maintenance level upgrades cannot change the version or release level of IBM WebSphere MQ. Maintenance level upgrades can be reversed, installations cannot be reversed.

“IBM WebSphere MQ migration” on page 22

“IBM WebSphere MQ upgrade” on page 21
The term upgrade, used loosely, covers both applying maintenance and modifying the version or release of a product. Upgrading is a change in any of the four V.R.M.F digits. After an upgrade has been applied, as long as no queue managers have been restarted, it can be backed out. The details vary by platform and how the upgrade was applied. Once a queue manager has been restarted, it depends on a number of factors whether the upgrade can be reversed. It depends on the platform, the V.R.M.F level, and the functions and command levels used by the queue manager at the new level.

HP-UX: Planning to migrate to a later version

Create a migration plan to migrate from your current version of IBM WebSphere MQ to a later version. You must back up your queue manager data before you migrate to the later version. You cannot revert a queue manager to the earlier version of the product once you start it on the later version.

Before you begin

If the queue manager is currently not on Version 7.1, see Table 1 on page 7 in “Where to find a topic about a specific migration path” on page 7.

About this task

Use the following steps as a guide to creating a migration plan.

Procedure

1. Review the WebSphere MQ system requirements for Version 7.5.

   See System Requirements for IBM WebSphere MQ.
2. Decide whether to run Version 7.1 and Version 7.5 on the same server.
3. Review all the changes in WebSphere MQ that affect you.

   See changes from “Changes in IBM WebSphere MQ Version 7.5” on page 138.

4. Review performance changes.

   Performance reports are published as Supportpacs; see WebSphere MQ - SupportPacs by Product.

5. Review the latest README file for the product you are working with.

   See WebSphere MQ and MQSeries product READMEs.

6. Plan the sequence and timing of queue manager upgrades.

   If the queue manager is part of a queue manager cluster, you must migrate the queue managers that are full repositories first.

   If the queue manager is part of a high availability cluster, plan the migration to minimize downtime and maximize availability; see “Migrate a queue manager in a high availability configuration” on page 44.

7. Plan to migrate your queue manager to Version 7.5.

   See “HP-UX: Migrating a queue manager to a later version” on page 88

   Backing up queue manager data is part of the queue manager migration task.

   An alternative approach to backing up queue manager data, is to install and configure a new server. Test Version 7.5 with a new queue manager on the new server. When you are ready to go into production on Version 7.5 copy the queue manager configuration and data, to the new server.

8. Plan to update any manual or automated procedures you have written with changes to messages and codes.
9. Plan to update applications that might be affected by changes you identified in “Changes in IBM WebSphere MQ Version 7.5” on page 138.

   Consider whether the application must be able to run on both Version 7.1 and Version 7.5. You might be able to change the application to be compatible with both code levels. If you cannot, you can query the queue manager command level, and make the code conditional on the command level. Call `MQINQ` setting the `MQIA_COMMAND_LEVEL` selector.

10. Decide on what regression tests to perform before putting the queue manager into production on Version 7.5.

    Include the procedures and applications you identified in steps 6 and 7 in your regression tests.

11. Plan to upgrade your WebSphere MQ MQI client installations to Version 7.5.
12. Plan to upgrade your client and server applications to use new functions in Version 7.5.

Directory structure on UNIX and Linux systems

Changing configuration information on Windows, UNIX and Linux systems

Fix Central

IBM Passport Advantage

“Migrate a queue manager in a high availability configuration” on page 44
Follow standard procedures to migrate a queue manager that is part of a high availability configuration on platforms other than z/OS.

“Queue manager cluster migration” on page 42
You can migrate queue managers in a cluster all at once, or one at a time, which is called a staged migration. Migrate full repository queue managers in a cluster before partial repository queue managers.

“Reverting a queue manager to a previous version” on page 39
On platforms other than z/OS it is not possible to reverse a migration. You can remove an upgrade before you have started a queue manager. After a queue manager has been started, if you remove the upgrade, the queue manager will not work.

WebSphere MQ - SupportPacs by Product

“Upgrade, migration, and maintenance of IBM WebSphere MQ on UNIX, Linux, and Windows” on page 25
You can install new releases of IBM WebSphere MQ to upgrade IBM WebSphere MQ to a new maintenance, release, or version level. Multiple installations at the same or different levels can coexist on the same UNIX, Linux, and Windows server. You can apply maintenance level upgrades to upgrade the

maintenance or fix level. Applying maintenance level upgrades cannot change the version or release level of IBM WebSphere MQ. Maintenance level upgrades can be reversed, installations cannot be reversed.

"IBM WebSphere MQ migration" on page 22
"IBM WebSphere MQ upgrade" on page 21
The term upgrade, used loosely, covers both applying maintenance and modifying the version or release of a product. Upgrading is a change in any of the four V.R.M.F digits. After an upgrade has been applied, as long as no queue managers have been restarted, it can be backed out. The details vary by platform and how the upgrade was applied. Once a queue manager has been restarted, it depends on a number of factors whether the upgrade can be reversed. It depends on the platform, the V.R.M.F level, and the functions and command levels used by the queue manager at the new level.

Linux: Planning to migrate to a later version

Create a migration plan to migrate from your current version of IBM WebSphere MQ to a later version. You must back up your queue manager data before you migrate to the later version. You cannot revert a queue manager to the earlier version of the product once you start it on the later version.

Before you begin

If the queue manager is currently not on Version 7.1, see Table 1 on page 7 in "Where to find a topic about a specific migration path" on page 7.

About this task

Use the following steps as a guide to creating a migration plan.

Procedure

1. Review the WebSphere MQ system requirements for Version 7.5.

   See System Requirements for IBM WebSphere MQ.
2. Decide whether to run Version 7.1 and Version 7.5 on the same server.
3. Review all the changes in WebSphere MQ that affect you.

   See changes from "Changes in IBM WebSphere MQ Version 7.5" on page 138.
4. Review performance changes.

   Performance reports are published as Supportpacs; see WebSphere MQ - SupportPacs by Product.
5. Review the latest README file for the product you are working with.

   See WebSphere MQ and MQSeries product READMEs.
6. Plan the sequence and timing of queue manager upgrades.

   If the queue manager is part of a queue manager cluster, you must migrate the queue managers that are full repositories first.
   If the queue manager is part of a high availability cluster, plan the migration to minimize downtime and maximize availability; see "Migrate a queue manager in a high availability configuration" on page 44.
7. Plan to migrate your queue manager to Version 7.5.

   See "Linux: Migrating a queue manager to a later version" on page 91
   Backing up queue manager data is part of the queue manager migration task.
   An alternative approach to backing up queue manager data, is to install and configure a new server. Test Version 7.5 with a new queue manager on the new server. When you are ready to go into production on Version 7.5 copy the queue manager configuration and data, to the new server.
8. Plan to update any manual or automated procedures you have written with changes to messages and codes.

9. Plan to update applications that might be affected by changes you identified in "Changes in IBM WebSphere MQ Version 7.5" on page 138.

   Consider whether the application must be able to run on both Version 7.1 and Version 7.5. You might be able to change the application to be compatible with both code levels. If you cannot, you can query the queue manager command level, and make the code conditional on the command level. Call `MQINQ` setting the `MQIA_COMMAND_LEVEL` selector.
10. Decide on what regression tests to perform before putting the queue manager into production on Version 7.5.

    Include the procedures and applications you identified in steps 6 and 7 in your regression tests.
11. Plan to upgrade your WebSphere MQ MQI client installations to Version 7.5.
12. Plan to upgrade your client and server applications to use new functions in Version 7.5.

Directory structure on UNIX and Linux systems

Changing configuration information on Windows, UNIX and Linux systems

Fix Central

IBM Passport Advantage

"Migrate a queue manager in a high availability configuration" on page 44
Follow standard procedures to migrate a queue manager that is part of a high availability configuration on platforms other than z/OS.

"Queue manager cluster migration" on page 42
You can migrate queue managers in a cluster all at once, or one at a time, which is called a staged migration. Migrate full repository queue managers in a cluster before partial repository queue managers.

"Reverting a queue manager to a previous version" on page 39
On platforms other than z/OS it is not possible to reverse a migration. You can remove an upgrade before you have started a queue manager. After a queue manager has been started, if you remove the upgrade, the queue manager will not work.

WebSphere MQ - SupportPacs by Product

"Upgrade, migration, and maintenance of IBM WebSphere MQ on UNIX, Linux, and Windows" on page 25
You can install new releases of IBM WebSphere MQ to upgrade IBM WebSphere MQ to a new maintenance, release, or version level. Multiple installations at the same or different levels can coexist on the same UNIX, Linux, and Windows server. You can apply maintenance level upgrades to upgrade the maintenance or fix level. Applying maintenance level upgrades cannot change the version or release level of IBM WebSphere MQ. Maintenance level upgrades can be reversed, installations cannot be reversed.

"IBM WebSphere MQ migration" on page 22

"IBM WebSphere MQ upgrade" on page 21
The term upgrade, used loosely, covers both applying maintenance and modifying the version or release of a product. Upgrading is a change in any of the four V.R.M.F digits. After an upgrade has been applied, as long as no queue managers have been restarted, it can be backed out. The details vary by platform and how the upgrade was applied. Once a queue manager has been restarted, it depends on a number of factors whether the upgrade can be reversed. It depends on the platform, the V.R.M.F level, and the functions and command levels used by the queue manager at the new level.

Solaris: Planning to migrate to a later version

Create a migration plan to migrate from your current version of IBM WebSphere MQ to a later version. You must back up your queue manager data before you migrate to the later version. You cannot revert a queue manager to the earlier version of the product once you start it on the later version.

Before you begin

If the queue manager is currently not on Version 7.1, see Table 1 on page 7 in "Where to find a topic about a specific migration path" on page 7.

About this task

Use the following steps as a guide to creating a migration plan.

Procedure

1. Review the WebSphere MQ system requirements for Version 7.5.

   See System Requirements for IBM WebSphere MQ.
2. Decide whether to run Version 7.1 and Version 7.5 on the same server.
3. Review all the changes in WebSphere MQ that affect you.

   See changes from “Changes in IBM WebSphere MQ Version 7.5” on page 138.
4. Review performance changes.

   Performance reports are published as Supportpacs; see WebSphere MQ - SupportPacs by Product.
5. Review the latest README file for the product you are working with.

   See WebSphere MQ and MQSeries product READMEs.
6. Plan the sequence and timing of queue manager upgrades.

   If the queue manager is part of a queue manager cluster, you must migrate the queue managers that are full repositories first.
   If the queue manager is part of a high availability cluster, plan the migration to minimize downtime and maximize availability; see “Migrate a queue manager in a high availability configuration” on page 44.
7. Plan to migrate your queue manager to Version 7.5.

   See “Solaris: Migrating a queue manager to a later version” on page 96
   Backing up queue manager data is part of the queue manager migration task.
   An alternative approach to backing up queue manager data, is to install and configure a new server. Test Version 7.5 with a new queue manager on the new server. When you are ready to go into production on Version 7.5 copy the queue manager configuration and data, to the new server.
8. Plan to update any manual or automated procedures you have written with changes to messages and codes.
9. Plan to update applications that might be affected by changes you identified in “Changes in IBM WebSphere MQ Version 7.5” on page 138.

   Consider whether the application must be able to run on both Version 7.1 and Version 7.5. You might be able to change the application to be compatible with both code levels. If you cannot, you can query the queue manager command level, and make the code conditional on the command level. Call `MQINQ` setting the `MQIA_COMMAND_LEVEL` selector.
10. Decide on what regression tests to perform before putting the queue manager into production on Version 7.5.

    Include the procedures and applications you identified in steps 6 and 7 in your regression tests.
11. Plan to upgrade your WebSphere MQ MQI client installations to Version 7.5.
12. Plan to upgrade your client and server applications to use new functions in Version 7.5.

Directory structure on UNIX and Linux systems

Changing configuration information on Windows, UNIX and Linux systems

Fix Central

IBM Passport Advantage

“Migrate a queue manager in a high availability configuration” on page 44
Follow standard procedures to migrate a queue manager that is part of a high availability configuration on platforms other than z/OS.

“Queue manager cluster migration” on page 42

You can migrate queue managers in a cluster all at once, or one at a time, which is called a staged migration. Migrate full repository queue managers in a cluster before partial repository queue managers.

“Reverting a queue manager to a previous version” on page 39
On platforms other than z/OS it is not possible to reverse a migration. You can remove an upgrade before you have started a queue manager. After a queue manager has been started, if you remove the upgrade, the queue manager will not work.

WebSphere MQ - SupportPacs by Product

“Upgrade, migration, and maintenance of IBM WebSphere MQ on UNIX, Linux, and Windows” on page 25
You can install new releases of IBM WebSphere MQ to upgrade IBM WebSphere MQ to a new maintenance, release, or version level. Multiple installations at the same or different levels can coexist on the same UNIX, Linux, and Windows server. You can apply maintenance level upgrades to upgrade the maintenance or fix level. Applying maintenance level upgrades cannot change the version or release level of IBM WebSphere MQ. Maintenance level upgrades can be reversed, installations cannot be reversed.

“IBM WebSphere MQ migration” on page 22
“IBM WebSphere MQ upgrade” on page 21
The term upgrade, used loosely, covers both applying maintenance and modifying the version or release of a product. Upgrading is a change in any of the four V.R.M.F digits. After an upgrade has been applied, as long as no queue managers have been restarted, it can be backed out. The details vary by platform and how the upgrade was applied. Once a queue manager has been restarted, it depends on a number of factors whether the upgrade can be reversed. It depends on the platform, the V.R.M.F level, and the functions and command levels used by the queue manager at the new level.

Windows: Planning to migrate to a later version

Create a migration plan to migrate from your current version of IBM WebSphere MQ to a later version. You must back up your queue manager data before you migrate to the later version. You cannot revert a queue manager to the earlier version of the product once you start it on the later version.

Before you begin

If the queue manager is currently not on Version 7.1, see Table 1 on page 7 in “Where to find a topic about a specific migration path” on page 7.

About this task

Use the following steps as a guide to creating a migration plan.

Procedure

1. Review the WebSphere MQ system requirements for Version 7.5.

   See System Requirements for IBM WebSphere MQ.
2. Decide whether to run Version 7.1 and Version 7.5 on the same server.
3. Review all the changes in WebSphere MQ that affect you.

   See changes from “Changes in IBM WebSphere MQ Version 7.5” on page 138.
4. Review performance changes.

   Performance reports are published as Supportpacs; see WebSphere MQ - SupportPacs by Product.
5. Review the latest README file for the product you are working with.

   See WebSphere MQ and MQSeries product READMEs.
6. Plan the sequence and timing of queue manager upgrades.

   If the queue manager is part of a queue manager cluster, you must migrate the queue managers that are full repositories first.

If the queue manager is part of a high availability cluster, plan the migration to minimize downtime and maximize availability; see “Migrate a queue manager in a high availability configuration” on page 44.

7. Plan to migrate your queue manager to Version 7.5.

   See “Windows: Migrating a queue manager to a later version” on page 99

   Backing up queue manager data is part of the queue manager migration task.

   An alternative approach to backing up queue manager data, is to install and configure a new server. Test Version 7.5 with a new queue manager on the new server. When you are ready to go into production on Version 7.5 copy the queue manager configuration and data, to the new server.
8. Plan to update any manual or automated procedures you have written with changes to messages and codes.
9. Plan to update applications that might be affected by changes you identified in “Changes in IBM WebSphere MQ Version 7.5” on page 138.

   Consider whether the application must be able to run on both Version 7.1 and Version 7.5. You might be able to change the application to be compatible with both code levels. If you cannot, you can query the queue manager command level, and make the code conditional on the command level. Call `MQINQ` setting the `MQIA_COMMAND_LEVEL` selector.
10. Decide on what regression tests to perform before putting the queue manager into production on Version 7.5.

    Include the procedures and applications you identified in steps 6 and 7 in your regression tests.
11. Plan to upgrade your WebSphere MQ MQI client installations to Version 7.5.
12. Plan to upgrade your client and server applications to use new functions in Version 7.5.

Changing configuration information on Windows, UNIX and Linux systems

Directory structure on Windows systems

Fix Central

IBM Passport Advantage

“Migrate a queue manager in a high availability configuration” on page 44
Follow standard procedures to migrate a queue manager that is part of a high availability configuration on platforms other than z/OS.

“Queue manager cluster migration” on page 42
You can migrate queue managers in a cluster all at once, or one at a time, which is called a staged migration. Migrate full repository queue managers in a cluster before partial repository queue managers.

“Reverting a queue manager to a previous version” on page 39
On platforms other than z/OS it is not possible to reverse a migration. You can remove an upgrade before you have started a queue manager. After a queue manager has been started, if you remove the upgrade, the queue manager will not work.

WebSphere MQ - SupportPacs by Product

“Upgrade, migration, and maintenance of IBM WebSphere MQ on UNIX, Linux, and Windows” on page 25
You can install new releases of IBM WebSphere MQ to upgrade IBM WebSphere MQ to a new maintenance, release, or version level. Multiple installations at the same or different levels can coexist on the same UNIX, Linux, and Windows server. You can apply maintenance level upgrades to upgrade the maintenance or fix level. Applying maintenance level upgrades cannot change the version or release level of IBM WebSphere MQ. Maintenance level upgrades can be reversed, installations cannot be reversed.

“IBM WebSphere MQ migration” on page 22

“IBM WebSphere MQ upgrade” on page 21
The term upgrade, used loosely, covers both applying maintenance and modifying the version or release of a product. Upgrading is a change in any of the four V.R.M.F digits. After an upgrade has been applied, as long as no queue managers have been restarted, it can be backed out. The details vary by platform and how the upgrade was applied. Once a queue manager has been restarted, it depends on a number of factors whether the upgrade can be reversed. It depends on the platform, the V.R.M.F level, and the functions and command levels used by the queue manager at the new level.

Publish/Subscribe migration from version 6.0
The publish/subscribe programming interface is integrated into the MQI. It is sometimes known as integrated publish/subscribe to distinguish it from queued publish/subscribe. Queued publish/subscribe is name given to the implementation of the Version 6.0 publish/subscribe command interface. You can use both queued and integrated publish/subscribe, and use them together with the same topics and subscriptions.

UNIX, Linux, and Windows: Single-stage migration to a later version

Single-stage migration is the term used to describe replacing the only installation of IBM WebSphere MQ on a server, with a later release. Single stage migration is also known as `upgrading in place` or `in place upgrade`. Until Version 7.0.1.6, single-stage was the only migration scenario. Single-stage migration preserves existing scripts and procedures for running IBM WebSphere MQ the most. With other migration scenarios you might change some scripts and procedures, but you can reduce the effect queue manager migration has on users.

Before you begin

This scenario is one of three, which describe alternative ways to upgrade queue managers from an earlier version of the product. The other scenarios are as follows:

1. Install the latest version of the product alongside an earlier version ; see “UNIX, Linux, and Windows: Side-by-side migration to a later version ” on page 65.
2. Run the latest version of the product alongside an earlier version ; see “UNIX, Linux, and Windows: Multi-stage migration to a later version” on page 72.

Read these three tasks to plan how you are going to migrate to the multi-installation environment of the latest version. Even if you do not plan to have more than one version of the installation on a server, read this topic. The steps to upgrade IBM WebSphere MQ from Version 7.0.1 to Version 7.5 have changed.

These topics are for planning multi-installation migration. The planning topics guide you in deciding what other tasks you must perform to migrate queue managers and applications to the latest version. For the precise sequence of commands to upgrade a queue manager to the latest version, do the migration task for the platform you are interested in. All the tasks are listed by platform in the links at the end of this topic. As part of the queue manager migration task, back up your existing queue manager data. Even on a multi-installation server, queue managers cannot be restored to a previous command level after migration.

About this task

In the single-stage migration scenario, the installation of the latest version of the product replaces an earlier version in the same installation location. It is the same migration process that you would have used to upgrade the product prior to IBM WebSphere MQ Version 7.0.1.6 . It is now termed "single-stage" migration, in contrast to "side-by-side" and "multi-stage" migration.

The advantage of single-stage migration is that it changes the configuration of a queue manager on the earlier version as little as possible. Existing applications switch from loading the libraries from the earlier version, to loading the libraries of the latest version, automatically.

Queue managers are automatically associated with the installation on the latest version. Administrative scripts and procedures are affected as little as possible by setting the installation to be the primary installation. If you set the installation of the latest version to be the primary installation, commands such as **strmqm** work without providing an explicit path to the command.

The description of the migration scenario starts with the example in Figure 8 on page 61.

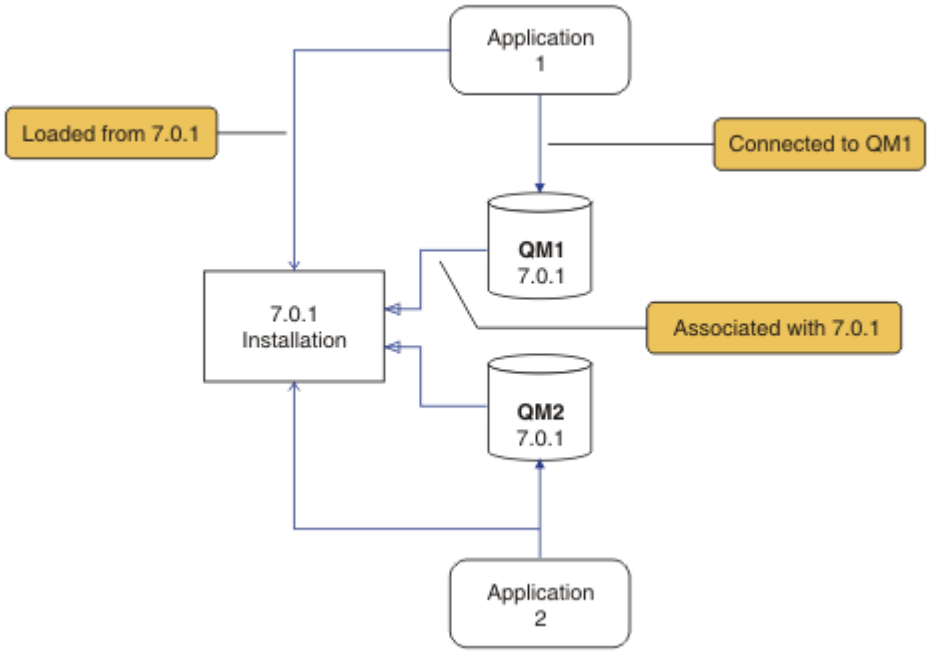


Figure 8. Existing server on an earlier version

Four types of object are important to consider during migration: installations, queue managers, administrative procedures, and applications. The diagram shows the installation an application loads IBM WebSphere MQ libraries from, connections between applications and queue managers, and associations between queue managers and installations. Administrative procedures are omitted from the diagram. Administrative procedures contain IBM WebSphere MQ commands, and scripts that use commands.

"Loading from 7.0.1" in Figure 8 on page 61, refers to the IBM WebSphere MQ installation from which the application loads the IBM WebSphere MQ library it requires; see “Loading IBM WebSphere MQ libraries” on page 32. The connection is a call to `MQCONN` or `MQCONNX`, and has not changed from the earlier version of the product to the latest version. The association is the installation that the queue manager is associated with. The association is created either by running the **`setmqm`** command, or by starting a queue manager on the earlier version; see Associating a queue manager with an installation.

To run a command, the operating system must find the command in a IBM WebSphere MQ installation. For some commands, you must run the command from the installation that is associated with the correct queue manager. IBM WebSphere MQ does not switch commands to the correct installation. For other commands, such as **`setmqinst`**, you can run the command from any installation that has the latest version of the product installed.

If an earlier version of the product is installed, the command that is run is the command for that version, unless the search path is overridden by a local setting. You can override the search path by running **`setmqenv`**. If Version 7.0.1 is not installed, you must set the correct path to run a command. If you have set a primary installation, the command that is run is the copy in the primary installation, unless you override the selection with a local search path.

Procedure

1. Upgrade the earlier version of the product to the latest version in the same installation directory; see Figure 9 on page 62.
   a) Decide on an installation naming convention. Give the installation a name of your choosing, or accept the default installation name.

      For the first installation, the default name is *Installation1*. For the second installation, the name is *Installation2*, and so on.

b) Upgrade the earlier version of the product to the latest version in place, or uninstall the earlier version, without deleting any queue managers, and install the latest version in the same default location.

Whether you have to uninstall your previous version of the product depends upon your operating system.

On:

- AIX
- Windows
- IBM i, where the process is known as a *slip* install

you do not have to uninstall a previous version of the product.

On:

- HP-UX
- Linux
- Solaris

you have to uninstall the previous version of the product.

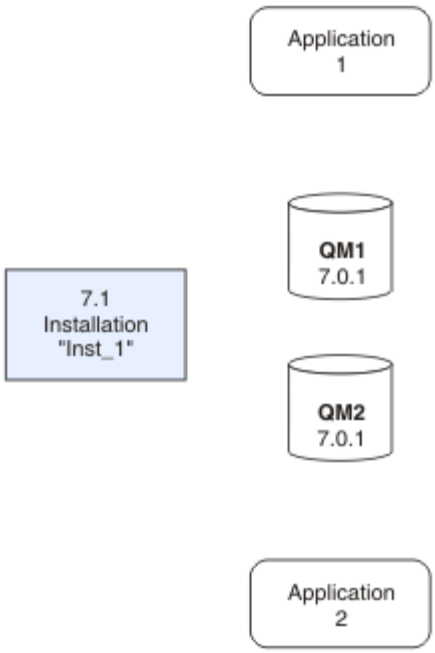


Figure 9. Upgrade of earlier version of product to latest version in the default directory

- A reason for installing into the same location is to simplify application migration. If you change the installation location, you might remove IBM WebSphere MQ libraries from an application search path. To migrate an application search path you must modify the application environment, or more rarely, the application itself.
- The default installation path is specified as a load path in the IBM WebSphere MQ build scripts for UNIX and Linux. After installation of the latest version, the load libraries of the latest version of IBM WebSphere MQ are in the same location as were the libraries of the earlier version. If you built applications by following the examples in the product documentation for the earlier version, the applications load the correct libraries in the latest version.

2. Make the latest version of the installation the primary installation; see Figure 10 on page 63.
   a) Run the **setmqinst** command

On Windows

```
"Inst_1_INSTALLATION_PATH\bin\setmqinst" -i -n Inst_1
```

On UNIX

```
Inst_1_INSTALLATION_PATH/bin/setmqinst -i -n Inst_1
```

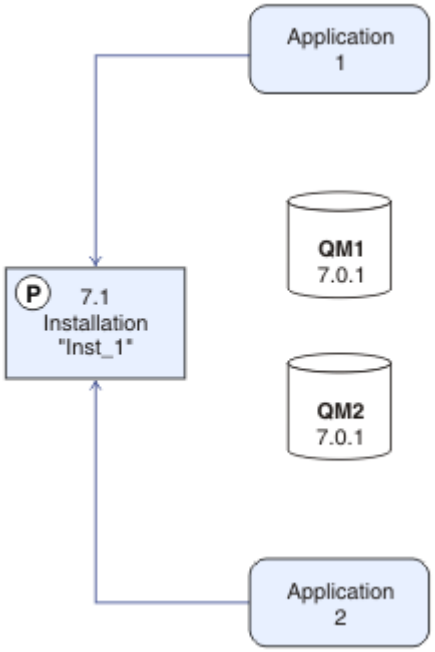


Figure 10. Prepare the installation and applications

- Make the installation primary to avoid specifying a search path to run IBM WebSphere MQ commands.
- If there is a primary installation, UNIX and Linux applications that expect to find the IBM WebSphere MQ library in `/usr/lib`, find a symbolic link to the library in `/usr/lib/32`[5]. `/usr/lib/32` is normally in the default search path. It is also specified as a load path in the IBM WebSphere MQ build scripts for UNIX and Linux.
- It is sufficient to link applications only to `/usr/lib`. With a primary installation of the latest version of the product defined on the server, an application can connect to any queue manager associated with any installation on the server. IBM WebSphere MQ loads the correct library for the application.

3. Start the queue managers and applications; see Figure 11 on page 64.
   a) Optional: Run the **setmqm** command to associate the queue managers with `Inst_1`.

```
setmqm -m QM1 -n Inst_1
setmqm -m QM2 -n Inst_1
```

   Running **setmqm** to associate the queue managers with `Inst_1` is optional, because these queue managers are being started on the latest version of the product for the first time.

   Windows If you have multiple installations, note that queue managers that were configured to start automatically, and remain after uninstalling IBM WebSphere MQ Version 7.0.1, automatically start under any other existing Version 7.1 (or laer) installation when either the machine reboots, or

[5] `/usr/lib` for 64 bit applications.

the Service for that installation is restarted. In order to avoid this, ensure that all queue managers have been moved to the desired installation before uninstalling IBM WebSphere MQ Version 7.0.1.

b) Run the **strmqm** command to start the queue managers and migrate them to the latest version of the product.

```
strmqm QM1
strmqm QM2
```

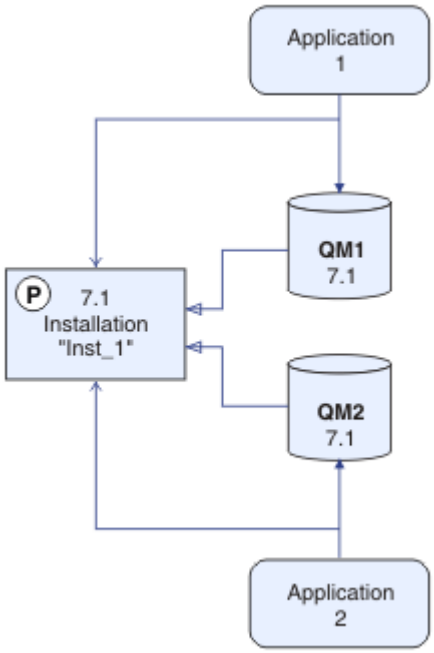


Figure 11. Restart the queue managers and applications

- When an application connects to a queue manager, the operating system searches its load path to load the IBM WebSphere MQ library[6]. A Version 7.5, or later, library contains code that checks that the queue manager is associated with an installation. If a queue manager is associated with a different installation, IBM WebSphere MQ loads the correct IBM WebSphere MQ library for the installation the queue manager is associated with.

What to do next

You cannot reinstall an earlier version of the product on a system that has the latest, or any other, version of IBM WebSphere MQ installed.

Related concepts

"Queue manager coexistence in Version 7.5" on page 28
Queue managers, with different names, can coexist on any server as long as they use the same IBM WebSphere MQ installation. On UNIX, Linux, and Windows, different queue managers can coexist on the same server and be associated with different installations.

"Multi-installation queue manager coexistence on UNIX, Linux, and Windows" on page 31

[6] On Windows, the IBM WebSphere MQ library is a DLL. A DLL is sometimes called a load library or a shared library. The entry points to a DLL are defined in a link library, with the file extension `.lib32` or `.lib`. The `.lib` library is linked at build-time and the DLL loaded at runtime.

You can install multiple copies of IBM WebSphere MQ for UNIX, Linux, and Windows on the same server. The installations must be at Version 7.1 or later, with one exception. One Version 7.0.1 installation, at fix pack level 6, or later, can coexist with multiple Version 7.1, or later installations.

Related tasks

"Migration planning on UNIX, Linux, and Windows" on page 51

"AIX: Migrating a queue manager to a later version" on page 84
Follow these instructions to migrate a queue manager from Version 7.1 to Version 7.5.

"Windows: Migrating a queue manager to a later version" on page 99
Follow these instructions to migrate a queue manager from Version 7.1 to Version 7.5.

Installing an IBM WebSphere MQ server

Associating a queue manager with an installation

"UNIX, Linux, and Windows: Side-by-side migration to a later version " on page 65

"UNIX, Linux, and Windows: Multi-stage migration to a later version" on page 72

Changing the primary installation

Choosing an installation name

"Migrating IBM WebSphere MQ library loading from Version 7.0.1 to Version 7.5" on page 108
No change in the way IBM WebSphere MQ libraries are loaded is normally required if you upgrade from Version 7.0.1 to Version 7.5. You must have followed the instructions on building IBM WebSphere MQ applications in Version 7.0.1 and you must replace IBM WebSphere MQ Version 7.0.1 with IBM WebSphere MQ Version 7.5. If you choose to take advantage of multi-installation in IBM WebSphere MQ Version 7.5, based on the side-by-side or multi-stage migration scenarios, you must modify the environment for the operating system to resolve IBM WebSphere MQ dependencies for an application. Typically, you can modify the runtime environment, rather than relink the application.

Related reference

setmqenv

setmqinst

setmqm

UNIX, Linux, and Windows: Side-by-side migration to a later version

Side-by-side migration is the term used to describe installing a new version of IBM WebSphere MQ alongside an older version on the same server. Queue managers remain running during the installation and verification of the new version of IBM WebSphere MQ. They remain associated with the older version of IBM WebSphere MQ. When you decide to migrate queue managers to the new version of IBM WebSphere MQ, you stop all queue managers, uninstall the old version , and migrate them all to the new version of IBM WebSphere MQ.

Before you begin

If you are using IBM WebSphere MQ Version 7.0.1, you must ensure that you are running IBM WebSphere MQ Version 7.0.1.6 before installing the latest version of the product on the same server. Go to Fix Central to obtain the fix pack.

This scenario is one of three, which describe alternative ways to upgrade queue managers from an earlier version of the product. The other scenarios are as follows:

1. Replace the earlier version with the latest version; see "UNIX, Linux, and Windows: Single-stage migration to a later version" on page 60.
2. Run the latest version of the product alongside an earlier version ; see "UNIX, Linux, and Windows: Multi-stage migration to a later version" on page 72.

Read these three tasks to plan how you are going to migrate to the multi-installation environment of the latest version. The step-by-step migration scenario sits half-way between the single-stage and multi-stage migration scenarios.

These topics are for planning multi-installation migration. The planning topics guide you in deciding what other tasks you must perform to migrate queue managers and applications to the latest version. For the precise sequence of commands to upgrade a queue manager to the latest version, do the migration task for the platform you are interested in. All the tasks are listed by platform in the links at the end of this topic. As part of the queue manager migration task, back up your existing queue manager data. Even on a multi-installation server, queue managers cannot be restored to a previous command level after migration.

About this task

In the "side-by-side" migration scenario, you install the latest version of IBM WebSphere MQ alongside queue managers that continue to be associated with Version 7.0.1, or later.

When you are ready to migrate the queue managers, and applications, to the latest version:

1. Stop all the queue managers.
2. Uninstall the earlier version of the product.
3. Migrate all the queue managers and applications to the latest version.

The side-by-side migration scenario is less flexible than multi-stage migration, and might not seem to have any advantages over it. However, side-by-side migration does have advantages over the multi-stage and single-stage approaches. With the side-by-side approach, because you uninstall the earlier version before starting any queue managers, you can assign an installation on the latest version to be the primary installation.

In the multi-stage approach, you cannot set an installation of the latest version to be the primary installation while you continue to run the earlier version.

Having the latest version installation as the primary installation has two benefits.

1. With the latest version having the primary installation, many applications restart without reconfiguring their environment.
2. IBM WebSphere MQ commands run against the primary installation, work without providing a local search path.

The advantage the side-by-side scenario has over the single-stage scenario is that you can install and verify the installation of the latest version of the product on the server before switching over to it.

The description of the migration scenario starts with the example in Figure 12 on page 67.

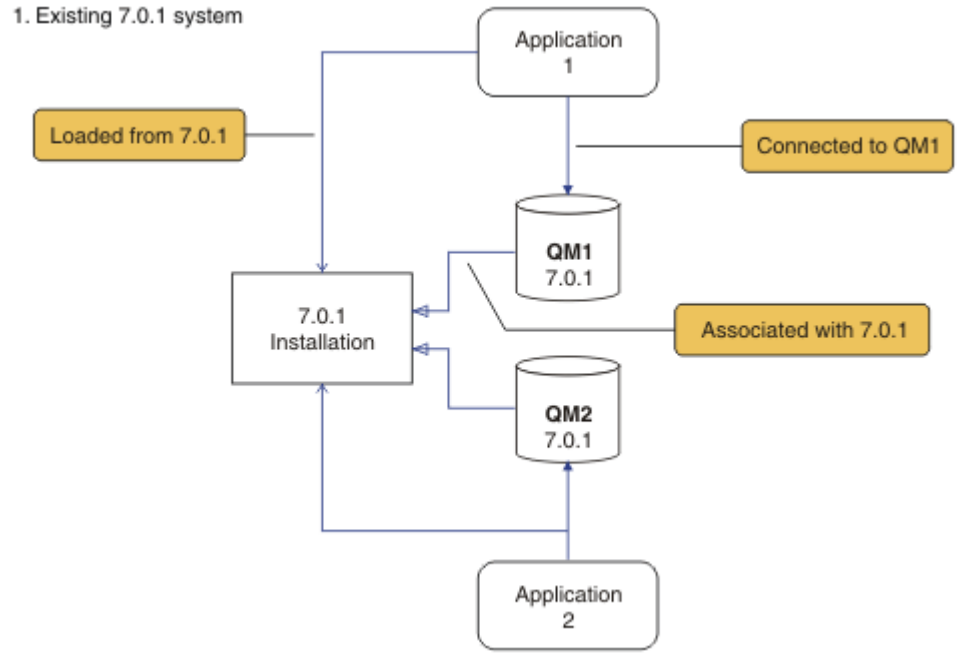


Figure 12. Existing Version 7.0.1 server

Four types of object are important to consider during migration: installations, queue managers, administrative procedures, and applications. The diagram shows the installation an application loads IBM WebSphere MQ libraries from, connections between applications and queue managers, and associations between queue managers and installations. Administrative procedures are omitted from the diagram. Administrative procedures contain IBM WebSphere MQ commands, and scripts that use commands.

"Loading from 7.0.1" in Figure 8 on page 61, refers to the IBM WebSphere MQ installation from which the application loads the IBM WebSphere MQ library it requires; see “Loading IBM WebSphere MQ libraries” on page 32. The connection is a call to `MQCONN` or `MQCONNX`, and has not changed from the earlier version of the product to the latest version. The association is the installation that the queue manager is associated with. The association is created either by running the **`setmqm`** command, or by starting a queue manager on the earlier version; see Associating a queue manager with an installation.

To run a command, the operating system must find the command in a IBM WebSphere MQ installation. For some commands, you must run the command from the installation that is associated with the correct queue manager. IBM WebSphere MQ does not switch commands to the correct installation. For other commands, such as **`setmqinst`**, you can run the command from any installation that has the latest version of the product installed.

If an earlier version of the product is installed, the command that is run is the command for that version, unless the search path is overridden by a local setting. You can override the search path by running **`setmqenv`**. If Version 7.0.1 is not installed, you must set the correct path to run a command. If you have set a primary installation, the command that is run is the copy in the primary installation, unless you override the selection with a local search path.

Procedure

1. Install the latest version in a different installation directory from the earlier version and verify the installation; see Figure 13 on page 68.
   a) Decide on an installation naming convention. Give the installation a name of your choosing, or accept the default installation name.

      For the first installation, the default name is *Installation1*. For the second installation, the name is *Installation2*, and so on.

b) Verify the installation.

Run the installation verification procedures and your own tests.

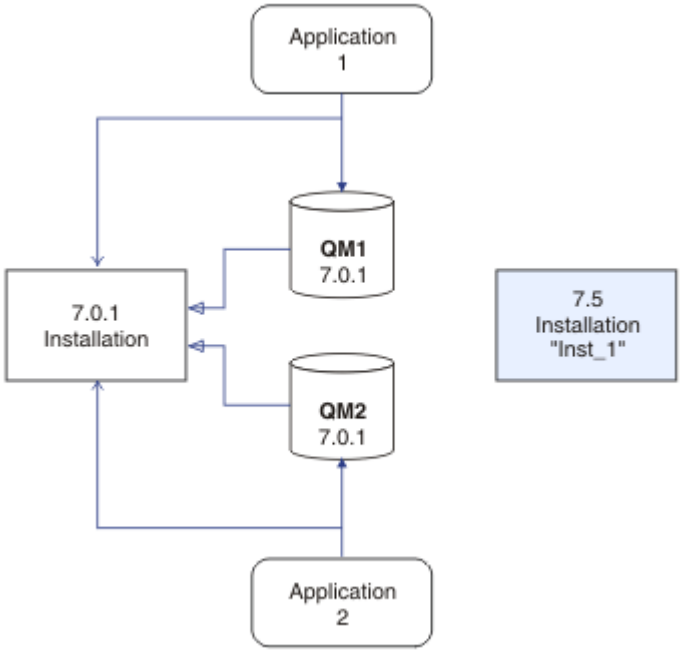


Figure 13. Install the latest version in a different directory

2. Uninstall the earlier version of the product; see Figure 14 on page 69.

   When uninstalling the earlier product, you must stop all queue managers and applications that have loaded a IBM WebSphere MQ library on the server. For this reason, you might choose to postpone uninstalling the earlier version of the product until a convenient maintenance window. When an earlier version of the product is not installed on a server, it is sufficient to stop the queue managers and applications that have loaded libraries from the installation that you are uninstalling or updating. It is not necessary to stop applications and queue managers associated with other installations.

   a) Stop all applications that have loaded IBM WebSphere MQ libraries on the server.

   b) Stop the queue managers and listeners on the server.

   c) Uninstall the earlier version of the product.

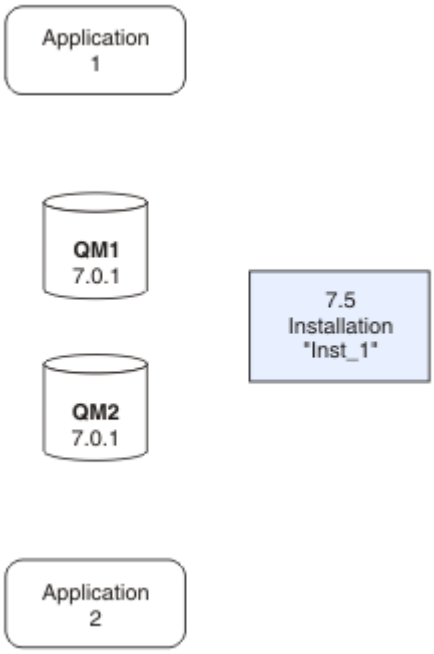


Figure 14. Uninstall an earlier version of the product

3. Make the latest version of the installation the primary installation; see Figure 15 on page 70.
   a) Run the **setmqinst** command

   On Windows

   ```
   "Inst_1_INSTALLATION_PATH\bin\setmqinst" -i -n Inst_1
   ```

   On UNIX

   ```
   Inst_1_INSTALLATION_PATH/bin/setmqinst -i -n Inst_1
   ```

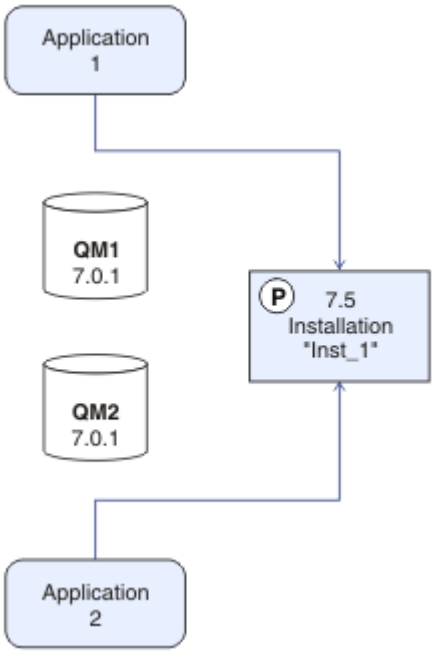


Figure 15. Prepare the installation and applications

- Make the installation primary to avoid specifying a search path to run IBM WebSphere MQ commands.
- If there is a primary installation, UNIX and Linux applications that expect to find the IBM WebSphere MQ library in `/usr/lib`, find a symbolic link to the library in `/usr/lib/32`[7]. `/usr/lib/32` is normally in the default search path. It is also specified as a load path in the IBM WebSphere MQ build scripts for UNIX and Linux.
- It is sufficient to link applications only to `/usr/lib`. With a primary installation of the latest version of the product defined on the server, an application can connect to any queue manager associated with any installation on the server. IBM WebSphere MQ loads the correct library for the application.

4. Start the queue managers and applications; see Figure 16 on page 71.
   a) Optional: Run the **setmqm** command to associate the queue managers with `Inst_1`.

```
setmqm -m QM1 -n Inst_1
setmqm -m QM2 -n Inst_1
```

   Running **setmqm** to associate the queue managers with `Inst_1` is optional, because these queue managers are being started on the latest version of the product for the first time.

   Windows If you have multiple installations, note that queue managers that were configured to start automatically, and remain after uninstalling IBM WebSphere MQ Version 7.0.1, automatically start under any other existing Version 7.1 (or laer) installation when either the machine reboots, or the Service for that installation is restarted. In order to avoid this, ensure that all queue managers have been moved to the desired installation before uninstalling IBM WebSphere MQ Version 7.0.1.

   b) Run the **strmqm** command to start the queue managers and migrate them to the latest version of the product.

```
strmqm QM1
strmqm QM2
```

[7] `/usr/lib` for 64 bit applications.

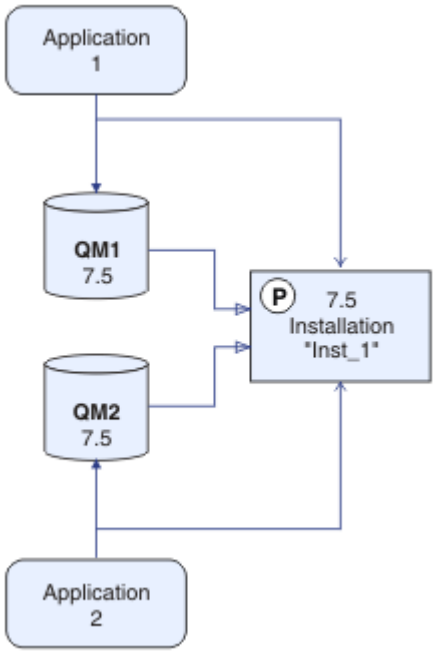


Figure 16. Restart the queue managers and applications

- When an application connects to a queue manager, the operating system searches its load path to load the IBM WebSphere MQ library[8]. A Version 7.5, or later, library contains code that checks that the queue manager is associated with an installation. If a queue manager is associated with a different installation, IBM WebSphere MQ loads the correct IBM WebSphere MQ library for the installation the queue manager is associated with.

What to do next

You cannot reinstall an earlier version of the product on a system that has the latest, or any other, version of IBM WebSphere MQ installed.

"Migration planning on UNIX, Linux, and Windows" on page 51

Installing an IBM WebSphere MQ server

Uninstalling IBM WebSphere MQ components

"UNIX, Linux, and Windows: Single-stage migration to a later version" on page 60

Single-stage migration is the term used to describe replacing the only installation of IBM WebSphere MQ on a server, with a later release. Single stage migration is also known as `upgrading in place` or `in place upgrade`. Until Version 7.0.1.6, single-stage was the only migration scenario. Single-stage migration preserves existing scripts and procedures for running IBM WebSphere MQ the most. With other migration scenarios you might change some scripts and procedures, but you can reduce the effect queue manager migration has on users.

"UNIX, Linux, and Windows: Multi-stage migration to a later version" on page 72

Changing the primary installation

Choosing an installation name

"Queue manager coexistence in Version 7.5" on page 28

[8] On Windows, the IBM WebSphere MQ library is a DLL. A DLL is sometimes called a load library or a shared library. The entry points to a DLL are defined in a link library, with the file extension `.lib32` or `.lib`. The `.lib` library is linked at build-time and the DLL loaded at runtime.

Queue managers, with different names, can coexist on any server as long as they use the same IBM WebSphere MQ installation. On UNIX, Linux, and Windows, different queue managers can coexist on the same server and be associated with different installations.

"Migrating IBM WebSphere MQ library loading from Version 7.0.1 to Version 7.5" on page 108
No change in the way IBM WebSphere MQ libraries are loaded is normally required if you upgrade from Version 7.0.1 to Version 7.5. You must have followed the instructions on building IBM WebSphere MQ applications in Version 7.0.1 and you must replace IBM WebSphere MQ Version 7.0.1 with IBM WebSphere MQ Version 7.5. If you choose to take advantage of multi-installation in IBM WebSphere MQ Version 7.5, based on the side-by-side or multi-stage migration scenarios, you must modify the environment for the operating system to resolve IBM WebSphere MQ dependencies for an application. Typically, you can modify the runtime environment, rather than relink the application.

"Multi-installation queue manager coexistence on UNIX, Linux, and Windows" on page 31
You can install multiple copies of IBM WebSphere MQ for UNIX, Linux, and Windows on the same server. The installations must be at Version 7.1 or later, with one exception. One Version 7.0.1 installation, at fix pack level 6, or later, can coexist with multiple Version 7.1, or later installations.

setmqenv
setmqinst
setmqm

UNIX, Linux, and Windows: Multi-stage migration to a later version

Multi-stage migration is the term used to describe running a new version of IBM WebSphere MQ alongside an older version on the same server. After installing the new version alongside the old, you can create new queue managers to verify the new installation, and develop new applications. At the same time, you can migrate queue managers and their associated applications from the old version to the new. By migrating queue managers and applications one-by-one, you can reduce the peak workload on staff managing the migration.

Before you begin

If you are using IBM WebSphere MQ Version 7.0.1, you must ensure that you are running IBM WebSphere MQ Version 7.0.1.6 before installing the latest version of the product on the same server. Go to Fix Central to obtain the fix pack.

This scenario is one of three, which describe alternative ways to upgrade queue managers from an earlier version of the product. The other scenarios are as follows:

1. Replace the earlier version with the latest version; see "UNIX, Linux, and Windows: Single-stage migration to a later version" on page 60.
2. Install the latest version of the product alongside an earlier version ; see "UNIX, Linux, and Windows: Side-by-side migration to a later version " on page 65.

Read these three tasks to plan how you are going to migrate to the multi-installation environment of Version 7.5. The multi-stage migration scenario is the most flexible approach to migrating from Version 7.0.1 to Version 7.5.

These topics are for planning multi-installation migration. The planning topics guide you in deciding what other tasks you must perform to migrate queue managers and applications to the latest version. For the precise sequence of commands to upgrade a queue manager to the latest version, do the migration task for the platform you are interested in. All the tasks are listed by platform in the links at the end of this topic. As part of the queue manager migration task, back up your existing queue manager data. Even on a multi-installation server, queue managers cannot be restored to a previous command level after migration.

Note:

- If an application uses COM or ActiveX it can connect to any queue manager as long as there is a primary installation and it is Version 7.5 or later.

- If you are running the IBM WebSphere MQ .NET monitor in transactional mode, the queue manager it connects to must be the primary installation.

You cannot migrate these applications to Version 7.5 until you uninstall Version 7.0.1.

About this task

In the multi-stage migration scenario, you install IBM WebSphere MQ Version 7.5 alongside running queue managers that continue to be associated with Version 7.0.1. You can create queue managers and run new applications using the IBM WebSphere MQ Version 7.5 installation. When you are ready to start migrating queue managers and applications from Version 7.0.1, you can do so, one-by-one. When migration to Version 7.5 is complete, uninstall Version 7.0.1, and make the Version 7.5 installation primary.

With the multi-stage approach, until you uninstall version 7.0.1, you must configure an environment to run applications that connect to a version 7.1 queue manager. You must also provide a path to run IBM WebSphere MQ commands. Both these tasks are accomplished with the **setmqenv** command.

Note: When you have uninstalled Version 7.0.1, and set a Version 7.5 installation primary, in most circumstances it is not necessary to run the **setmqenv** command to run applications. It is still necessary to run **setmqenv** to set the environment for commands that connect to a queue manager associated with an installation that is not primary.

The description of the migration scenario starts with the example in Figure 17 on page 73.

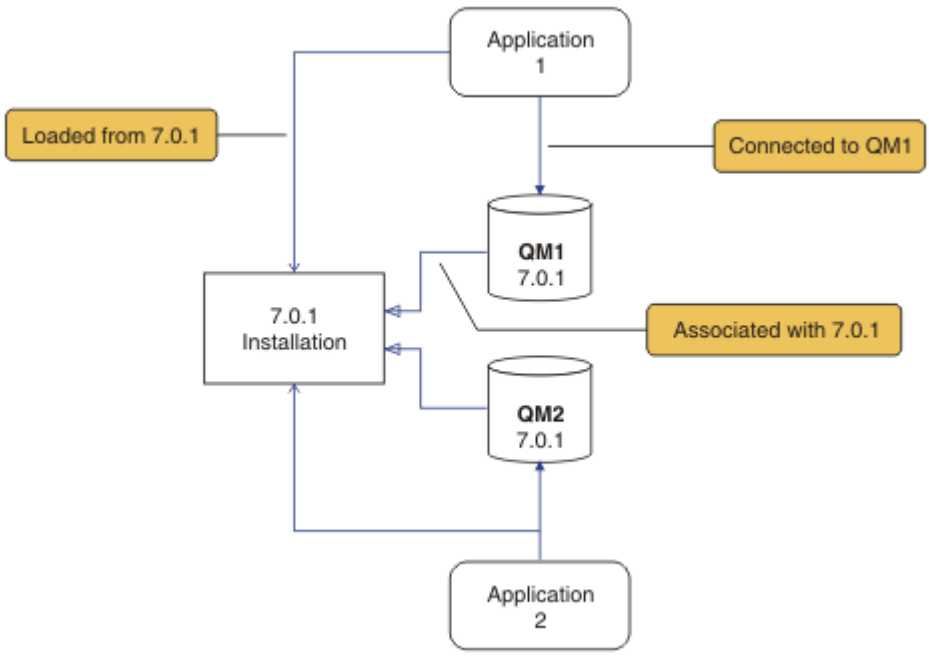


Figure 17. Existing Version 7.0.1 server

Four types of object are important to consider during migration: installations, queue managers, administrative procedures, and applications. The diagram shows the installation an application loads IBM WebSphere MQ libraries from, connections between applications and queue managers, and associations between queue managers and installations. Administrative procedures are omitted from the diagram. Administrative procedures contain IBM WebSphere MQ commands, and scripts that use commands.

"Loading from 7.0.1" in Figure 8 on page 61, refers to the IBM WebSphere MQ installation from which the application loads the IBM WebSphere MQ library it requires; see “Loading IBM WebSphere MQ libraries” on page 32. The connection is a call to `MQCONN` or `MQCONNX`, and has not changed from the earlier version of the product to the latest version. The association is the installation that the queue manager

is associated with. The association is created either by running the **setmqm** command, or by starting a queue manager on the earlier version; see Associating a queue manager with an installation.

To run a command, the operating system must find the command in a IBM WebSphere MQ installation. For some commands, you must run the command from the installation that is associated with the correct queue manager. IBM WebSphere MQ does not switch commands to the correct installation. For other commands, such as **setmqinst**, you can run the command from any installation that has the latest version of the product installed.

If an earlier version of the product is installed, the command that is run is the command for that version, unless the search path is overridden by a local setting. You can override the search path by running **setmqenv**. If Version 7.0.1 is not installed, you must set the correct path to run a command. If you have set a primary installation, the command that is run is the copy in the primary installation, unless you override the selection with a local search path.

Procedure

1. Install Version 7.5 in a different installation directory to Version 7.0.1 and verify the installation; see Figure 18 on page 74.
   a) Decide on an installation naming convention. Give the installation a name of your choosing, or accept the default installation name.

      For the first installation, the default name is *Installation1*. For the second installation, the name is *Installation2*, and so on.

   b) Verify the installation.

      Run the installation verification procedures and your own tests.

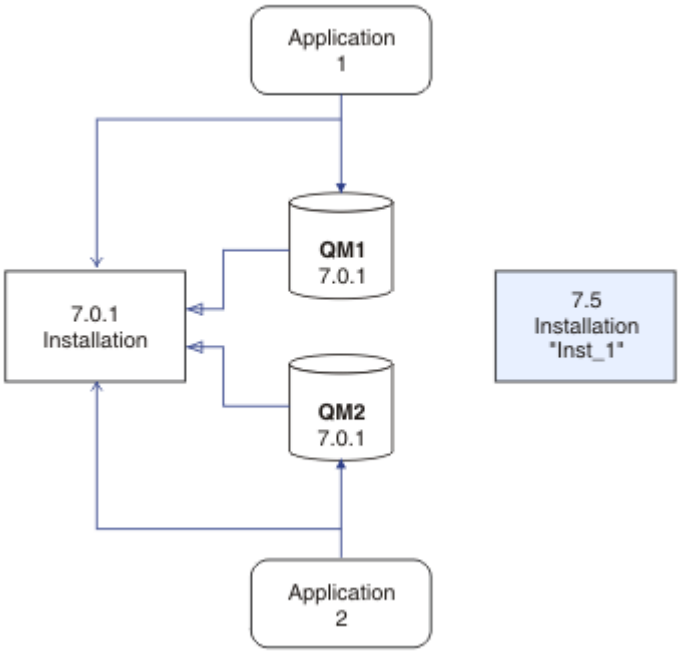


Figure 18. Install Version 7.5 in a different directory

   - You might create new queue managers running Version 7.5, and start to develop new applications before migrating applications from Version 7.0.1.

2. Configure the operating system so that applications load the Version 7.5 libraries; see Figure 19 on page 75.

Migrate queue managers one at a time. The first set of applications to load the Version 7.5 libraries are the applications that connect to the first queue manager you are going to migrate. It does not matter if those applications also connect to other queue managers on the server. If they load the Version 7.5 libraries, IBM WebSphere MQ automatically loads the Version 7.0.1 libraries for those applications that connect to Version 7.0.1. As the first step, you might either migrate the operating system environment of all applications, or just the applications that connect to the first queue manager you are going to migrate.

Some of the applications might be running as IBM WebSphere MQ MQI client applications on another workstation. When you migrate a queue manager, clients connected to it continue to run without loading a Version 7.5 client library. You can migrate these clients later, when you need to do so.

If any IBM WebSphere MQ MQI client applications are using the Version 7.0.1 library on the server, you must eventually migrate the clients to use Version 7.5 libraries, before you uninstall Version 7.0.1.

To make an application load a Version 7.5 library, you have three choices:

- Run **setmqenv** to modify the local path that is searched for IBM WebSphere MQ libraries.
- Modify the global search path that is searched for IBM WebSphere MQ libraries.
- Relink applications with an additional runtime load path.

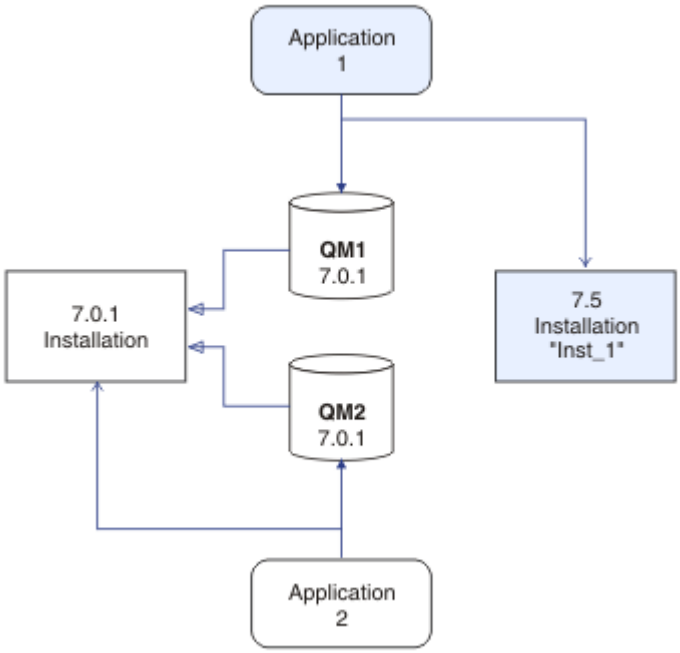


Figure 19. Application 1 loads IBM WebSphere MQ libraries from `Inst_1`

Consult operating system documentation about how to modify the global search path, or include a fixed runtime load path in the application load module.

To run **setmqenv** using the `-s` option:

Windows:

```
"Inst_1_INSTALLATION_PATH\bin\setmqenv" -s
```

The `-s` option sets up the environment for the installation that runs the **setmqenv** command.

UNIX:

```
. Inst_1_INSTALLATION_PATH/bin/setmqenv -s -k
```

The `-k` option inserts the path to the IBM WebSphere MQ load libraries at the start of the LD_LIBRARY_PATH environment variable, and adds the variable to the local environment; see “Loading IBM WebSphere MQ libraries” on page 32.

Note: On UNIX the leading `". "` is critical. The dot followed by a space instructs the command shell run **setmqenv** in the same command shell and inherit the environment set by **setmqenv**.

3. Restart the queue manager and the applications that connect to it; see Figure 20 on page 76.

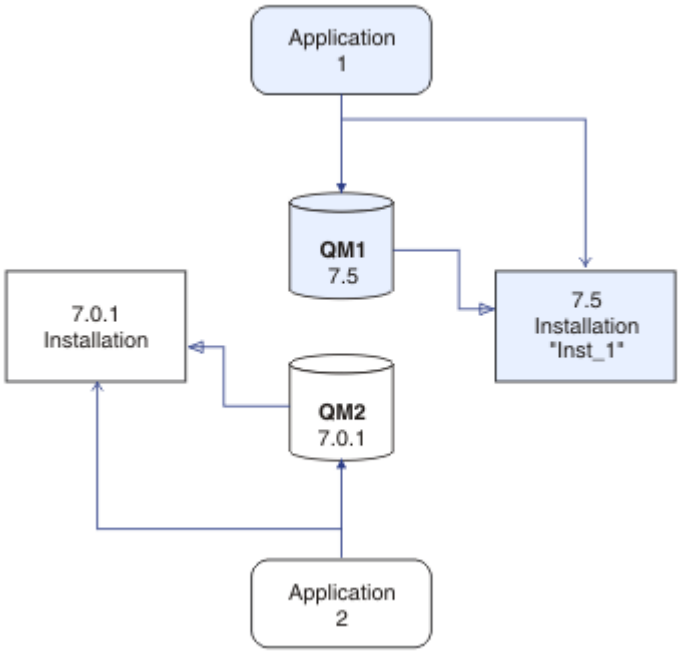


Figure 20. Restart QM1 and application 1.

a) Set up the local environment to the installation `Inst_1`.

Windows:

```
"Inst_1_INSTALLATION_PATH\bin\setmqenv" -s
```

The `-s` option sets up the environment for the installation that runs the **setmqenv** command.

UNIX:

```
. Inst_1_INSTALLATION_PATH/bin/setmqenv -s
```

b) Run the **setmqm** command to associate `QM1` with `Inst_1`.

```
setmqm -m QM1 -n Inst_1
```

c) Run the **strmqm** command to start `QM1` and migrate it to Version 7.5.

```
strmqm QM1
```

d) Restart application 1

The application loads the Version 7.5 library and connects to QM1, which is associated with Version 7.5.

4. Migrate all queue managers and applications to Version 7.5; see Figure 21 on page 77.

Repeat steps “2” on page 74 and “3” on page 76, when required, until all the queue managers and applications are migrated to Version 7.5.

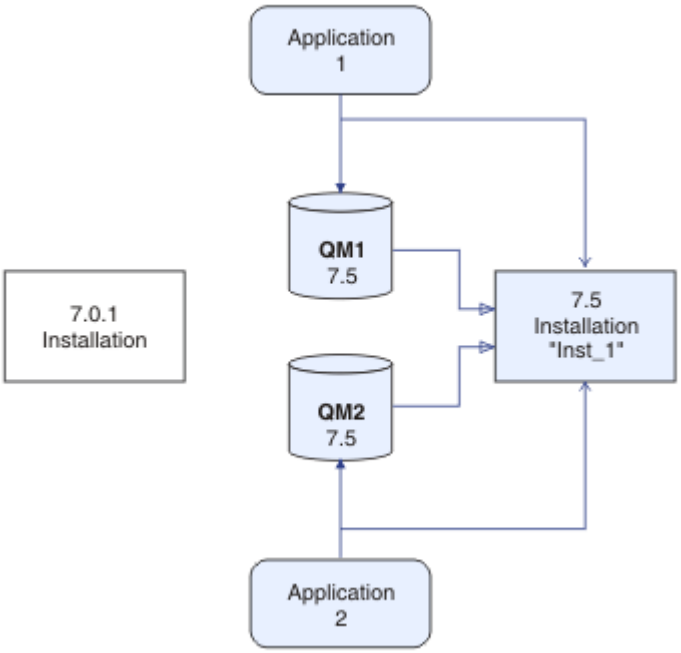


Figure 21. Migrate all queue managers and applications to Version 7.5

5. Uninstall Version 7.0.1; see Figure 22 on page 78.

When uninstalling the earlier product, you must stop all queue managers and applications that have loaded a IBM WebSphere MQ library on the server. For this reason, you might choose to postpone uninstalling the earlier version of the product until a convenient maintenance window. When an earlier version of the product is not installed on a server, it is sufficient to stop the queue managers and applications that have loaded libraries from the installation that you are uninstalling or updating. It is not necessary to stop applications and queue managers associated with other installations.

a) Stop all applications that have loaded IBM WebSphere MQ libraries on the server.

b) Stop the queue managers and listeners on the server.

c) Uninstall the earlier version of the product.

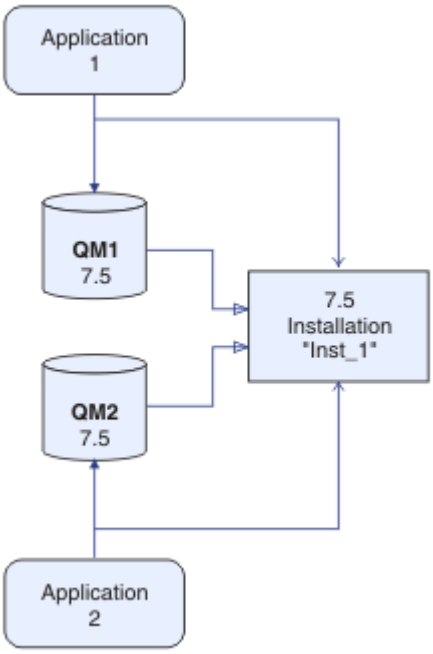


Figure 22. Uninstall Version 7.0.1

6. Make `Inst_1` the primary installation; see Figure 23 on page 79.
   a) Run the **setmqinst** command

   On Windows

   ```
   "Inst_1_INSTALLATION_PATH\bin\setmqinst" -i -n Inst_1
   ```

   On UNIX

   ```
   Inst_1_INSTALLATION_PATH/bin/setmqinst -i -n Inst_1
   ```

   - You do not have to set up a search path to run IBM WebSphere MQ commands from the primary installation.
   - If you set a Version 7.5 installation as primary on UNIX and Linux, you do not have to set up `LD_LIBRARY_PATH` in most cases. You can remove calls to **setmqenv** to set `LD_LIBRARY_PATH`.

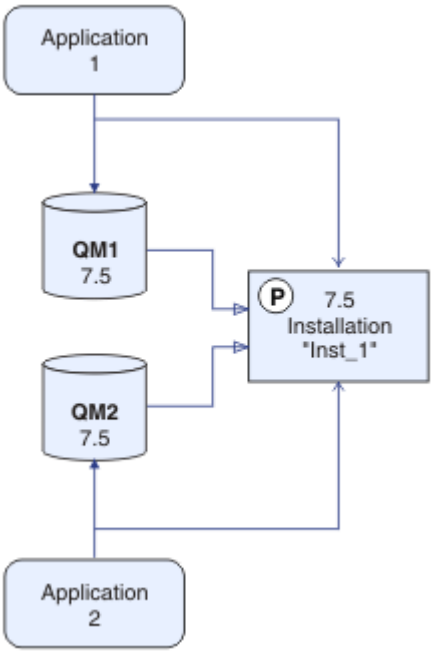


Figure 23. Make Version 7.5 primary

What to do next

You cannot reinstall an earlier version of the product on a system that has the latest, or any other, version of IBM WebSphere MQ installed.

Now that you have uninstalled Version 7.0.1, and made a Version 7.5 installation primary, you can review how the application runtime environment is set. It is no longer necessary to run **setmqenv** to set up the search path to load IBM WebSphere MQ libraries. If you have only one Version 7.5 installation it is not necessary to run **setmqenv** to run commands.

Related concepts

"Queue manager coexistence in Version 7.5" on page 28
Queue managers, with different names, can coexist on any server as long as they use the same IBM WebSphere MQ installation. On UNIX, Linux, and Windows, different queue managers can coexist on the same server and be associated with different installations.

"Multi-installation queue manager coexistence on UNIX, Linux, and Windows" on page 31
You can install multiple copies of IBM WebSphere MQ for UNIX, Linux, and Windows on the same server. The installations must be at Version 7.1 or later, with one exception. One Version 7.0.1 installation, at fix pack level 6, or later, can coexist with multiple Version 7.1, or later installations.

Related tasks

"Migration planning on UNIX, Linux, and Windows" on page 51

Installing an IBM WebSphere MQ server

"UNIX, Linux, and Windows: Single-stage migration to a later version" on page 60
Single-stage migration is the term used to describe replacing the only installation of IBM WebSphere MQ on a server, with a later release. Single stage migration is also known as `upgrading in place` or `in place upgrade`. Until Version 7.0.1.6, single-stage was the only migration scenario. Single-stage migration preserves existing scripts and procedures for running IBM WebSphere MQ the most. With other migration scenarios you might change some scripts and procedures, but you can reduce the effect queue manager migration has on users.

"UNIX, Linux, and Windows: Side-by-side migration to a later version " on page 65

"Migrating IBM WebSphere MQ library loading from Version 7.0.1 to Version 7.5" on page 108
No change in the way IBM WebSphere MQ libraries are loaded is normally required if you upgrade from Version 7.0.1 to Version 7.5. You must have followed the instructions on building IBM WebSphere MQ applications in Version 7.0.1 and you must replace IBM WebSphere MQ Version 7.0.1 with IBM WebSphere MQ Version 7.5. If you choose to take advantage of multi-installation in IBM WebSphere MQ Version 7.5, based on the side-by-side or multi-stage migration scenarios, you must modify the environment for the operating system to resolve IBM WebSphere MQ dependencies for an application. Typically, you can modify the runtime environment, rather than relink the application.

Related reference

setmqenv

setmqinst

setmqm

Related information

Associating a queue manager with an installation

Changing the primary installation

Choosing an installation name

Migrating IBM WebSphere MQ Telemetry

Migrate IBM WebSphere MQ Telemetry from Version 7.0.1 to Version 7.5 by completing the tasks in this section. You must stop all IBM WebSphere MQ activity on the system before migrating.

In IBM WebSphere MQ Version 7.0.1, IBM WebSphere MQ Telemetry was a separate feature. Because IBM WebSphere MQ Telemetry is a component of IBM WebSphere MQ Version 7.5 and later, upgrading is essentially uninstalling IBM WebSphere MQ TelemetryVersion 7.0.1 and installing IBM WebSphere MQ Version 7.5. IBM WebSphere MQ Telemetry can either be installed with the main product, or installed after Version 7.5 or later is already installed.

After the successful upgrade, Linux systems retain all telemetry data kept in `/var/mqm`, and the Windows systems retain the telemetry data in the IBM WebSphere MQ Version 7.5 installation directory, for example: `C:\Program Files (x86)\IBM\WebSphere MQ\`. Telemetry data is migrated to Version 7.5 when the queue manager is started again.

Note:

You can administer IBM WebSphere MQ Telemetry Version 7.0.1 only from the Version 7.0.1 IBM WebSphere MQ Explorer. If you connect the Version 7.5 explorer remotely to a Version 7.0.1 queue manager, no telemetry resources are displayed. You cannot connect a Version 7.5 IBM WebSphere MQ Explorer locally to a Version 7.0.1 queue manager on the same server.

The migration tasks, depending on your platform, are described in the following subtopics:

Related concepts

"IBM WebSphere MQ Telemetry migration from Version 7.0.1" on page 50
IBM WebSphere MQ Telemetry Version 7.0.1 is a separate feature that you install with IBM WebSphere MQ Version 7.0.1. Telemetry is integrated into IBM WebSphere MQ Version 7.5.

Related tasks

Installing WebSphere MQ

Installing WebSphere MQ Telemetry

Windows: Migrating IBM WebSphere MQ Telemetry

Follow these instructions to migrate IBM WebSphere MQ Telemetry from Version 7.0.1 to Version 7.5 on Windows.

Before you begin

Before proceeding with this task, ensure that you back up your existing IBM WebSphere MQ installation. You must stop the IBM WebSphere MQ Telemetry service `SYSTEM.MQXR.SERVICE` before migrating.

About this task

This task outlines the steps necessary to migrate your existing installation of IBM WebSphere MQ Telemetry to Version 7.5 on Windows systems.

Procedure

1. Uninstall IBM WebSphere MQ Version 7.0.1 using the control panel, following this procedure:
   a) From the Windows task bar, click **Start > Control Panel**. (On some systems, click **Start > Settings > Control Panel**)
   b) When uninstalling IBM WebSphere MQ Telemetry by using the control panel, on Windows 7 systems, click **Programs and Features**, on other Windows systems, click **Add or Remove Programs**.
   c) Click IBM WebSphere MQ Telemetry, then click **Change/Remove**. The uninstaller starts and summarizes items to be uninstalled.
   d) Click **Uninstall**. The uninstaller lists the items being uninstalled. The **Uninstall Complete** screen displays the status of the uninstall. Click **Done**
2. Verify that the data folders still exist. For example, they might be located at `C:\Program Files (x86)\IBM\WebSphere MQ\`.
3. IBM WebSphere MQ Telemetry is installed in one of two ways:
   - Installed as part of a **Custom** installation at the same time as IBM WebSphere MQ Version 7.5 during migration: Begin here: "Migration planning on UNIX, Linux, and Windows" on page 51
   - Added at a later date to an existing installation of IBM WebSphere MQ Version 7.5 or later: Begin here: Installing IBM WebSphere MQ Telemetry
4. Verify that the IBM WebSphere MQ Telemetry migration was successful by following the tasks at Verifying the installation of IBM WebSphere MQ Telemetry by using IBM WebSphere MQ Explorer or Verifying the installation of IBM WebSphere MQ Telemetry using the command line.

Results

After completing this task, the existing MQTT channels are still present. Previous subscriptions are also still present. System output similar to the following example is expected:

AMQ4616
A newer command level has been found when connecting to *<insert_0>*. The old level is *<insert_1>* and the new level is *<insert_2>*. The connection to the queue manager will be replaced.

Severity
0: Information

Explanation
A previous connection to this queue manager has been successful; the queue manager is the same but the command level is now higher. The version of WebSphere MQ has been changed.

Response
Message for information only.

Related information

Installing WebSphere MQ Telemetry

Verifying the installation of WebSphere MQ Telemetry

Verifying the installation of WebSphere MQ Telemetry by using WebSphere MQ Explorer

Linux: Migrating IBM WebSphere MQ Telemetry

Follow these instructions to migrate IBM WebSphere MQ Telemetry from Version 7.0.1 to Version 7.5 on Linux.

Before you begin

Before proceeding with this task, ensure that you back up your existing IBM WebSphere MQ installation. You must stop the IBM WebSphere MQ Telemetry service `SYSTEM.MQXR.SERVICE` before migrating.

About this task

This task outlines the steps necessary to migrate your existing installation of IBM WebSphere MQ Telemetry to Version 7.5 on Linux systems.

Procedure

1. Uninstall IBM WebSphere MQ Telemetry Version 7.0.1, following this procedure:
   a) Navigate to the uninstallation directory. The default location on Linux is `/opt/mqm/mqxr/Uninstall_MQTT`.
   b) Start the uninstaller, using the executable or binary file. On Linux systems, run `./Uninstall_MQTelemetry -i GUI`. The uninstaller starts and summarizes what is to be uninstalled.
   c) Click **Uninstall**. The uninstaller lists the items being uninstalled.
   d) Click **Done**.
2. Verify that the data folders still exist. Consult your *MQ_INSTALLATION_PATH* environment variable to find these folders.
3. IBM WebSphere MQ Telemetry is installed in one of two ways:
   - Installed as part of a **Custom** installation at the same time as IBM WebSphere MQ Version 7.5 during migration: Begin here: “Migration planning on UNIX, Linux, and Windows” on page 51
   - Added at a later date to an existing installation of IBM WebSphere MQ Version 7.5 or later: Begin here: Installing IBM WebSphere MQ Telemetry
4. Verify that the IBM WebSphere MQ Telemetry migration was successful by following the tasks at Verifying the installation of IBM WebSphere MQ Telemetry by using IBM WebSphere MQ Explorer or Verifying the installation of IBM WebSphere MQ Telemetry using the command line.

Results

After completing this task, the existing MQTT channels are still present. Previous subscriptions are also still present. System output similar to the following example is expected:

AMQ4616
A newer command level has been found when connecting to *<insert_0>*. The old level is *<insert_1>* and the new level is *<insert_2>*. The connection to the queue manager will be replaced.

Severity
0: Information

Explanation
A previous connection to this queue manager has been successful; the queue manager is the same but the command level is now higher. The version of WebSphere MQ has been changed.

Response

Message for information only.

Related tasks

Installing WebSphere MQ Telemetry

Verifying the installation of WebSphere MQ Telemetry

Verifying the installation of WebSphere MQ Telemetry by using WebSphere MQ Explorer

Migrating IBM WebSphere MQ to a later version on UNIX, Linux, and Windows

Follow these tasks to migrate queue managers from one version to another, and to migrate queue managers that are in clusters. You will also find assistance in migrating publish/subscribe or telemetry from WebSphere Event Broker and WebSphere Message Broker to IBM WebSphere MQ.

About this task

The upgrade from Version 7.1 to Version 7.5 requires a full migration of queue managers. If you start a queue manager on Version 7.5, you cannot restore it to Version 7.1.

Related concepts

“Migration paths” on page 6

Where can you find topics describing how to migrate WebSphere MQ from one release to another? Can you revert a migration to an earlier release? Can you migrate from an unsupported WebSphere MQ release to any later release directly?

Related reference

“Changes that affect migration” on page 137

Migrating a queue manager to a later version on UNIX, Linux, and Windows

The procedures for migrating a queue manager from Version 7.1 to Version 7.5 are listed by platform.

Before you begin

If you have installed early support program code on the server, you must delete all the queue managers created with the installation. Uninstall the code before proceeding with installing the production level code.

About this task

Migration is a complex task. It goes beyond upgrading the IBM WebSphere MQ code. The term "upgrading" applies to the process of installing a new code level. "Migrating" refers to the process of upgrading code and the to the task of migrating queue manager data, such as messages, queues, channels, and other resources. Migrating queue manager data is automatic.

Procedure

Queue manager migration follows this overall plan:

1. Review the WebSphere MQ system requirements for Version 7.5; see System Requirements for IBM WebSphere MQ.
2. Back up your system. In particular back up the queue manager.

   You cannot run a queue manager on a previous version of IBM WebSphere MQ, once you start the queue manager on the new version.
3. Carry out the platform-specific migration or upgrade procedure.

   Consult “Migration planning on UNIX, Linux, and Windows” on page 51.

a) Upgrade IBM WebSphere MQ.

b) Customize the configuration.

c) Verify the installation.

Create a queue manager to verify the installation. Verify applications and management tasks work with the new level of IBM WebSphere MQ before migrating existing queue managers. Consult “Characteristics of different types of upgrade” on page 26 to help you decide on the extent of testing required.

4. Perform any additional migration tasks that are required.

   If the queue manager is a member of a queue manager cluster, or queue-sharing group, migrate the other members of the cluster or group.

5. Perform any additional migration tasks that are required.

   If you are using publish/subscribe, you must migrate the publish/subscribe broker.
   If the queue manager is a member of a queue manager cluster, or queue-sharing group, migrate the other members of the cluster or group.

Related tasks

“Migration planning on UNIX, Linux, and Windows” on page 51

“Migration paths” on page 6

Where can you find topics describing how to migrate WebSphere MQ from one release to another? Can you revert a migration to an earlier release? Can you migrate from an unsupported WebSphere MQ release to any later release directly?

System Requirements for IBM WebSphere MQ

AIX: Migrating a queue manager to a later version

Follow these instructions to migrate a queue manager from Version 7.1 to Version 7.5.

Before you begin

1. Create a migration plan. Use the planning task, “AIX: Planning to migrate to a later version” on page 52, as a guide.
2. Review the IBM WebSphere MQ system requirements for Version 7.5; see System Requirements for IBM WebSphere MQ.
3. Back up your system before you install IBM WebSphere MQ Version 7.5 over Version 7.1. Once you have started a queue manager you cannot revert to the previous release. If you must restore the system, you cannot recover any work, such as changes to messages and objects, performed by IBM WebSphere MQ Version 7.5. For more information about backing up your system, see Backing up and restoring IBM WebSphere MQ queue manager data .
4. Review any other installed SupportPacs for their applicability to IBM WebSphere MQ Version 7.5.
5. If you are running on a server with multiple IBM WebSphere MQ installations, that is, IBM WebSphere MQ Version 7.0.1, Fix Pack 6 (or later), you must identify the installation. Make sure that the commands you enter run against the correct installation; see setmqenv.

About this task

Complete this task to migrate a queue manager to IBM WebSphere MQ Version 7.5 from Version 7.1.

Procedure

1. Log in as a user in `group mqm`.
2. Stop all applications using the IBM WebSphere MQ installation.

   If you use the IBM WebSphere MQ Managed File Transfer (MFT) component, ensure that any MFT agents have finished all of the file transfers that they were engaged in. There should be no

incomplete transfers associated with the agents, and their SYSTEM.FTE.STATE queues should contain no messages.

3. End all the activity of queue managers associated with the IBM WebSphere MQ installation.
   a) Run the **dspmq** command to list the state of all the queue managers on the system.

      Run either of the following commands from the installation that you are updating:

      ```
      dspmq -o installation -o status
      dspmq -a
      ```

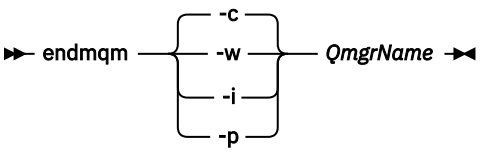
      dspmq -o installation -o status displays the installation name and status of queue managers associated with all installations of IBM WebSphere MQ.

      dspmq -a displays the status of active queue managers associated with the installation from which the command is run.

   b) Run the **MQSC** command, `DISPLAY LSSTATUS(*) STATUS` to list the status of listeners associated with a queue manager.

      ```
      echo "DISPLAY LSSTATUS(*) STATUS" | runmqsc QmgrName
      ```

   c) Run the **endmqm** command to stop each running queue manager associated with this installation.

      The **endmqm** command informs an application that the queue manager it is connected to is stopping; see Stopping a queue manager.

      For the maintenance to proceed, applications must respond to an **endmqm** command by disconnecting from the queue manager and releasing any IBM WebSphere MQ libraries they have loaded. If they do not, you must find another way to force applications to release IBM WebSphere MQ resources, such as by stopping the applications.

      You must also stop applications that are using the client libraries that are part of the installation. Client applications might be connected to a different queue manager, running a different installation of IBM WebSphere MQ. The application is not informed about queue managers in the current installation being shut down.

      Any applications that continue to have IBM WebSphere MQ shared libraries from the installation loaded prevent you applying IBM WebSphere MQ maintenance. An application might disconnect from a queue manager, or be forcibly disconnected, but keep a IBM WebSphere MQ shared library loaded.

   d) Stop any listeners associated with the queue managers, using the command:

      ```
      endmqlsr -m QMgrName
      ```

4. Back up the queue manager.
5. Log in as root.
6. Optionally, uninstall IBM WebSphere MQ Version 7.1.

   Note, that you carry out this step only if you are doing a single stage migration; see “UNIX, Linux, and Windows: Single-stage migration to a later version” on page 60
7. Install IBM WebSphere MQ Version 7.5

   See Installing WebSphere MQ Server on AIX.
8. Move the queue manager to the new IBM WebSphere MQ installation.

   You need to carry out this step, only if you did not uninstall your current version of IBM WebSphere MQ.

See “UNIX, Linux, and Windows: Side-by-side migration to a later version ” on page 65 or “UNIX, Linux, and Windows: Multi-stage migration to a later version” on page 72 for further information.

9. Start the queue manager.

```
strmqm QmgrName
```

When you first start a queue manager after migration:

- Any new attributes for existing objects are set to their default values.
- Any new default objects are created.
- Queue manager data is migrated.

Important: Do not use the `-c` option to start the queue manager, unless you explicitly want to reset or recreate the default system objects.

You must start IBM WebSphere MQ before you start any listeners.

Backing up and restoring a queue manager

“The version naming scheme for IBM WebSphere MQ on UNIX, Linux, and Windows” on page 8
On UNIX, Linux, and Windows, IBM WebSphere MQ releases have a four-digit Version, Release, Modification, and Fix (VRMF) level code. The version and release parts of the code are significant; they identify the service life of a release. To run a queue manager at a different VR level, you must migrate the queue manager, its applications, and the environment in which it runs. Depending on the migration path, the migration might require more or less effort.

AIX: Reverting a queue manager to an earlier version

You can restore a queue manager to Version 7.1 from Version 7.5, if you have made a backup of the system or queue manager. If you have started the queue manager and processed any messages, or changed the configuration, the task cannot give you any guidance on restoring the current state of the queue manager.

Before you begin

1. You must have made a backup of the system or queue manager before you upgraded to Version 7.5. For more information see Backing up and restoring IBM WebSphere MQ queue manager data
2. If any messages were processed after starting the queue manager, you cannot easily undo the effects of processing the messages. You cannot restore the queue manager to Version 7.1 in its current state. The task cannot give you any guidance how to deal with subsequent changes that have occurred. For example, messages that were indoubt in a channel, or in a transmission queue on another queue manager, might have been processed. If the queue manager is part of a cluster, then configuration messages and application messages might have been exchanged.
3. If you are running on a server with multiple IBM WebSphere MQ installations, that is, IBM WebSphere MQ Version 7.0.1, Fix Pack 6 (or later), you must identify the installation. Make sure that the commands you enter run against the correct installation; see setmqenv.

About this task

When you restore a previous version of a queue manager, you restore the queue manager to its earlier code level. Queue manager data is restored to the state it was in when the queue manager was backed up.

Procedure

1. Log in as a user in `group mqm`.
2. Stop all applications using the IBM WebSphere MQ installation.

   If you use the IBM WebSphere MQ Managed File Transfer (MFT) component, ensure that any MFT agents have finished all of the file transfers that they were engaged in. There should be no

incomplete transfers associated with the agents, and their SYSTEM.FTE.STATE queues should contain no messages.

3. End all the activity of queue managers associated with the IBM WebSphere MQ installation.
   a) Run the **dspmq** command to list the state of all the queue managers on the system.

      Run either of the following commands from the installation that you are updating:

      ```
      dspmq -o installation -o status
      dspmq -a
      ```

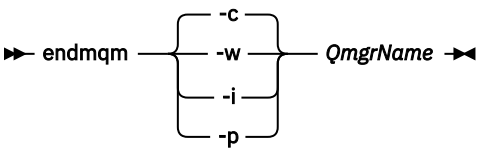
      dspmq -o installation -o status displays the installation name and status of queue managers associated with all installations of IBM WebSphere MQ.

      dspmq -a displays the status of active queue managers associated with the installation from which the command is run.

   b) Run the **MQSC** command, `DISPLAY LSSTATUS(*) STATUS` to list the status of listeners associated with a queue manager.

      ```
      echo "DISPLAY LSSTATUS(*) STATUS" | runmqsc QmgrName
      ```

   c) Run the **endmqm** command to stop each running queue manager associated with this installation.

      The **endmqm** command informs an application that the queue manager it is connected to is stopping; see Stopping a queue manager.

      For the maintenance to proceed, applications must respond to an **endmqm** command by disconnecting from the queue manager and releasing any IBM WebSphere MQ libraries they have loaded. If they do not, you must find another way to force applications to release IBM WebSphere MQ resources, such as by stopping the applications.

      You must also stop applications that are using the client libraries that are part of the installation. Client applications might be connected to a different queue manager, running a different installation of IBM WebSphere MQ. The application is not informed about queue managers in the current installation being shut down.

      Any applications that continue to have IBM WebSphere MQ shared libraries from the installation loaded prevent you applying IBM WebSphere MQ maintenance. An application might disconnect from a queue manager, or be forcibly disconnected, but keep a IBM WebSphere MQ shared library loaded.

   d) Stop any listeners associated with the queue managers, using the command:

      ```
      endmqlsr -m QMgrName
      ```

4. Restore the system, or IBM WebSphere MQ and the queue manager.

   If your backup procedure was to save the queue manager data, you must reinstall IBM WebSphere MQ.

   a) Uninstall the Version 7.5 installation.

   b) Reinstall IBM WebSphere MQ from a manufacturing refresh.

   c) Apply the fix pack and interim fixes that restore IBM WebSphere MQ to its previous level.

   d) Restore the queue manager data from the backup taken before installing Version 7.5.

5. Restart the Version 7.1 queue manager.

What to do next

You might be restoring Version 7.1 on a server with multiple IBM WebSphere MQ installations. If one of the installations is primary, after restoring Version 7.1, the Version 7.1 installation will be by default, the primary. You must review how applications connect to an installation. After restoring Version 7.1 some applications might connect to the wrong installation.

Related tasks

Backing up and restoring a queue manager

HP-UX: Migrating a queue manager to a later version

Follow these instructions to migrate a queue manager from Version 7.1 to Version 7.5.

Before you begin

1. Create a migration plan. Use the planning task, "HP-UX: Planning to migrate to a later version" on page 53, as a guide.
2. Review the IBM WebSphere MQ system requirements for Version 7.5; see System Requirements for IBM WebSphere MQ.
3. Back up your system before you install IBM WebSphere MQ Version 7.5 over Version 7.1. Once you have started a queue manager you cannot revert to the previous release. If you must restore the system, you cannot recover any work, such as changes to messages and objects, performed by IBM WebSphere MQ Version 7.5. For more information about backing up your system, see Backing up and restoring IBM WebSphere MQ queue manager data .
4. Review any other installed SupportPacs for their applicability to IBM WebSphere MQ Version 7.5.
5. If you are running on a server with multiple IBM WebSphere MQ installations, that is, IBM WebSphere MQ Version 7.0.1, Fix Pack 6 (or later), you must identify the installation. Make sure that the commands you enter run against the correct installation; see setmqenv.

About this task

Complete this task to migrate a queue manager to IBM WebSphere MQ Version 7.5 from Version 7.1.

Procedure

1. Log in as a user in `group mqm`.
2. Stop all applications using the IBM WebSphere MQ installation.

   If you use the IBM WebSphere MQ Managed File Transfer (MFT) component, ensure that any MFT agents have finished all of the file transfers that they were engaged in. There should be no incomplete transfers associated with the agents, and their SYSTEM.FTE.STATE queues should contain no messages.
3. End all the activity of queue managers associated with the IBM WebSphere MQ installation.
   a) Run the **dspmq** command to list the state of all the queue managers on the system.

      Run either of the following commands from the installation that you are updating:

      ```
      dspmq -o installation -o status
      dspmq -a
      ```

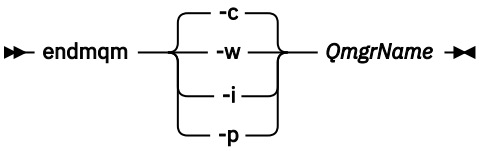
      dspmq -o installation -o status displays the installation name and status of queue managers associated with all installations of IBM WebSphere MQ.

      dspmq -a displays the status of active queue managers associated with the installation from which the command is run.

   b) Run the **MQSC** command, `DISPLAY LSSTATUS(*) STATUS` to list the status of listeners associated with a queue manager.

```
echo "DISPLAY LSSTATUS(*) STATUS" | runmqsc QmgrName
```

c) Run the **endmqm** command to stop each running queue manager associated with this installation.



The **endmqm** command informs an application that the queue manager it is connected to is stopping; see Stopping a queue manager.

For the maintenance to proceed, applications must respond to an **endmqm** command by disconnecting from the queue manager and releasing any IBM WebSphere MQ libraries they have loaded. If they do not, you must find another way to force applications to release IBM WebSphere MQ resources, such as by stopping the applications.

You must also stop applications that are using the client libraries that are part of the installation. Client applications might be connected to a different queue manager, running a different installation of IBM WebSphere MQ. The application is not informed about queue managers in the current installation being shut down.

Any applications that continue to have IBM WebSphere MQ shared libraries from the installation loaded prevent you applying IBM WebSphere MQ maintenance. An application might disconnect from a queue manager, or be forcibly disconnected, but keep a IBM WebSphere MQ shared library loaded.

d) Stop any listeners associated with the queue managers, using the command:

```
endmqlsr -m QMgrName
```

4. Back up the queue manager.
5. Log in as root.
6. Optionally, uninstall IBM WebSphere MQ Version 7.1.

   Note, that you carry out this step only if you are doing a single stage migration; see “UNIX, Linux, and Windows: Single-stage migration to a later version” on page 60
7. Install IBM WebSphere MQ Version 7.5

   See Installing WebSphere MQ Server on HP-UX.
8. Move the queue manager to the new IBM WebSphere MQ installation.

   You need to carry out this step, only if you did not uninstall your current version of IBM WebSphere MQ.

   See “UNIX, Linux, and Windows: Side-by-side migration to a later version ” on page 65 or “UNIX, Linux, and Windows: Multi-stage migration to a later version” on page 72 for further information.
9. Start the queue manager.

   ```
   strmqm QmgrName
   ```

   When you first start a queue manager after migration:

   - Any new attributes for existing objects are set to their default values.
   - Any new default objects are created.
   - Queue manager data is migrated.

   Important: Do not use the -c option to start the queue manager, unless you explicitly want to reset or recreate the default system objects.

   You must start IBM WebSphere MQ before you start any listeners.

Backing up and restoring a queue manager
“The version naming scheme for IBM WebSphere MQ on UNIX, Linux, and Windows” on page 8

On UNIX, Linux, and Windows, IBM WebSphere MQ releases have a four-digit Version, Release, Modification, and Fix (VRMF) level code. The version and release parts of the code are significant; they identify the service life of a release. To run a queue manager at a different VR level, you must migrate the queue manager, its applications, and the environment in which it runs. Depending on the migration path, the migration might require more or less effort.

HP-UX: Reverting a queue manager to an earlier version

You can restore a queue manager to Version 7.1 from Version 7.5, if you have made a backup of the system or queue manager. If you have started the queue manager and processed any messages, or changed the configuration, the task cannot give you any guidance on restoring the current state of the queue manager.

Before you begin

1. You must have made a backup of the system or queue manager before you upgraded to Version 7.5. For more information see Backing up and restoring IBM WebSphere MQ queue manager data
2. If any messages were processed after starting the queue manager, you cannot easily undo the effects of processing the messages. You cannot restore the queue manager to Version 7.1 in its current state. The task cannot give you any guidance how to deal with subsequent changes that have occurred. For example, messages that were indoubt in a channel, or in a transmission queue on another queue manager, might have been processed. If the queue manager is part of a cluster, then configuration messages and application messages might have been exchanged.
3. If you are running on a server with multiple IBM WebSphere MQ installations, that is, IBM WebSphere MQ Version 7.0.1, Fix Pack 6 (or later), you must identify the installation. Make sure that the commands you enter run against the correct installation; see setmqenv.

About this task

When you restore a previous version of a queue manager, you restore the queue manager to its earlier code level. Queue manager data is restored to the state it was in when the queue manager was backed up.

Procedure

1. Log in as a user in `group mqm`.
2. Stop all applications using the IBM WebSphere MQ installation.

   If you use the IBM WebSphere MQ Managed File Transfer (MFT) component, ensure that any MFT agents have finished all of the file transfers that they were engaged in. There should be no incomplete transfers associated with the agents, and their SYSTEM.FTE.STATE queues should contain no messages.
3. End all the activity of queue managers associated with the IBM WebSphere MQ installation.
   a) Run the **dspmq** command to list the state of all the queue managers on the system.

      Run either of the following commands from the installation that you are updating:

      ```
      dspmq -o installation -o status
      dspmq -a
      ```

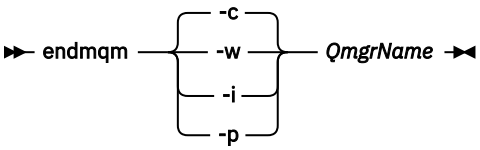
      dspmq -o installation -o status displays the installation name and status of queue managers associated with all installations of IBM WebSphere MQ.

      dspmq -a displays the status of active queue managers associated with the installation from which the command is run.

   b) Run the **MQSC** command, `DISPLAY LSSTATUS(*) STATUS` to list the status of listeners associated with a queue manager.

      ```
      echo "DISPLAY LSSTATUS(*) STATUS" | runmqsc QmgrName
      ```

c) Run the **endmqm** command to stop each running queue manager associated with this installation.



The **endmqm** command informs an application that the queue manager it is connected to is stopping; see Stopping a queue manager.

For the maintenance to proceed, applications must respond to an **endmqm** command by disconnecting from the queue manager and releasing any IBM WebSphere MQ libraries they have loaded. If they do not, you must find another way to force applications to release IBM WebSphere MQ resources, such as by stopping the applications.

You must also stop applications that are using the client libraries that are part of the installation. Client applications might be connected to a different queue manager, running a different installation of IBM WebSphere MQ. The application is not informed about queue managers in the current installation being shut down.

Any applications that continue to have IBM WebSphere MQ shared libraries from the installation loaded prevent you applying IBM WebSphere MQ maintenance. An application might disconnect from a queue manager, or be forcibly disconnected, but keep a IBM WebSphere MQ shared library loaded.

d) Stop any listeners associated with the queue managers, using the command:

```
endmqlsr -m QMgrName
```

4. Restore the system, or IBM WebSphere MQ and the queue manager.

   If your backup procedure was to save the queue manager data, you must reinstall IBM WebSphere MQ.

   a) Uninstall the Version 7.5 installation.

   b) Reinstall IBM WebSphere MQ from a manufacturing refresh.

   c) Apply the fix pack and interim fixes that restore IBM WebSphere MQ to its previous level.

   d) Restore the queue manager data from the backup taken before installing Version 7.5.

5. Restart the Version 7.1 queue manager.

What to do next

You might be restoring Version 7.1 on a server with multiple IBM WebSphere MQ installations. If one of the installations is primary, after restoring Version 7.1, the Version 7.1 installation will be by default, the primary. You must review how applications connect to an installation. After restoring Version 7.1 some applications might connect to the wrong installation.

Related tasks

Backing up and restoring a queue manager

Linux: Migrating a queue manager to a later version

Follow these instructions to migrate a queue manager from Version 7.1 to Version 7.5.

Before you begin

1. Create a migration plan. Use the planning task, “Linux: Planning to migrate to a later version” on page 55, as a guide.
2. Review the IBM WebSphere MQ system requirements for Version 7.5; see System Requirements for IBM WebSphere MQ.

3. If you are considering switching IBM WebSphere MQ for Linux System x from a 32 bit to a 64 bit installation do not migrate to Version 7.5. Your last chance to switch to 64 bit is the migration to Version 7.0.1.
4. Do not use the **rpm** -U (upgrade) or -F (freshen) options to upgrade IBM WebSphere MQ for Linux; see “Linux: Cleaning up after using the rpm freshen or upgrade options” on page 93.
5. Back up your system before you install IBM WebSphere MQ Version 7.5 over Version 7.1. Once you have started a queue manager you cannot revert to the previous release. If you must restore the system, you cannot recover any work, such as changes to messages and objects, performed by IBM WebSphere MQ Version 7.5. For more information about backing up your system, see Backing up and restoring IBM WebSphere MQ queue manager data .
6. Review any other installed SupportPacs for their applicability to IBM WebSphere MQ Version 7.5.
7. If you are running on a server with multiple IBM WebSphere MQ installations, that is, IBM WebSphere MQ Version 7.0.1, Fix Pack 6 (or later), you must identify the installation. Make sure that the commands you enter run against the correct installation; see setmqenv.

About this task

Complete this task to migrate a queue manager to IBM WebSphere MQ Version 7.5 from Version 7.1.

Procedure

1. Log in as a user in `group mqm`.
2. Stop all applications using the IBM WebSphere MQ installation.

   If you use the IBM WebSphere MQ Managed File Transfer (MFT) component, ensure that any MFT agents have finished all of the file transfers that they were engaged in. There should be no incomplete transfers associated with the agents, and their SYSTEM.FTE.STATE queues should contain no messages.
3. End all the activity of queue managers associated with the IBM WebSphere MQ installation.
   a) Run the **dspmq** command to list the state of all the queue managers on the system.

      Run either of the following commands from the installation that you are updating:

      ```
      dspmq -o installation -o status
      dspmq -a
      ```

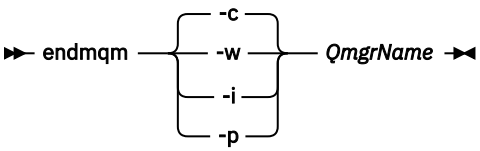
      dspmq -o installation -o status displays the installation name and status of queue managers associated with all installations of IBM WebSphere MQ.

      dspmq -a displays the status of active queue managers associated with the installation from which the command is run.

   b) Run the **MQSC** command, `DISPLAY LSSTATUS(*) STATUS` to list the status of listeners associated with a queue manager.

      ```
      echo "DISPLAY LSSTATUS(*) STATUS" | runmqsc QmgrName
      ```

   c) Run the **endmqm** command to stop each running queue manager associated with this installation.

      The **endmqm** command informs an application that the queue manager it is connected to is stopping; see Stopping a queue manager.

      For the maintenance to proceed, applications must respond to an **endmqm** command by disconnecting from the queue manager and releasing any IBM WebSphere MQ libraries they

have loaded. If they do not, you must find another way to force applications to release IBM WebSphere MQ resources, such as by stopping the applications.

You must also stop applications that are using the client libraries that are part of the installation. Client applications might be connected to a different queue manager, running a different installation of IBM WebSphere MQ. The application is not informed about queue managers in the current installation being shut down.

Any applications that continue to have IBM WebSphere MQ shared libraries from the installation loaded prevent you applying IBM WebSphere MQ maintenance. An application might disconnect from a queue manager, or be forcibly disconnected, but keep a IBM WebSphere MQ shared library loaded.

d) Stop any listeners associated with the queue managers, using the command:

```
endmqlsr -m QMgrName
```

4. Back up the queue manager.
5. Log in as root.
6. Optionally, uninstall IBM WebSphere MQ Version 7.1.

   Note, that you carry out this step only if you are doing a single stage migration; see “UNIX, Linux, and Windows: Single-stage migration to a later version” on page 60
7. Install IBM WebSphere MQ Version 7.5

   See Installing WebSphere MQ Server on Linux.
8. Move the queue manager to the new IBM WebSphere MQ installation.

   You need to carry out this step, only if you did not uninstall your current version of IBM WebSphere MQ.

   See “UNIX, Linux, and Windows: Side-by-side migration to a later version ” on page 65 or “UNIX, Linux, and Windows: Multi-stage migration to a later version” on page 72 for further information.
9. Start the queue manager.

   ```
   strmqm QmgrName
   ```

   When you first start a queue manager after migration:

   - Any new attributes for existing objects are set to their default values.
   - Any new default objects are created.
   - Queue manager data is migrated.

   Important: Do not use the `-c` option to start the queue manager, unless you explicitly want to reset or recreate the default system objects.

   You must start IBM WebSphere MQ before you start any listeners.

Backing up and restoring a queue manager

“The version naming scheme for IBM WebSphere MQ on UNIX, Linux, and Windows” on page 8
On UNIX, Linux, and Windows, IBM WebSphere MQ releases have a four-digit Version, Release, Modification, and Fix (VRMF) level code. The version and release parts of the code are significant; they identify the service life of a release. To run a queue manager at a different VR level, you must migrate the queue manager, its applications, and the environment in which it runs. Depending on the migration path, the migration might require more or less effort.

Linux: Cleaning up after using the `rpm` freshen or upgrade options

The use of **`rpm`** upgrade or freshen options is not supported. If you use the options, follow this cleanup procedure, and then install following the correct steps.

Before you begin

You have attempted to upgrade IBM WebSphere MQ for Linux using `rpm -U` or `rpm -F`

About this task

By using the freshen or upgrade options, you might have deleted your old IBM WebSphere MQ package entries from the **rpm** database without removing the product from your system. You might also have partially installed IBM WebSphere MQ

Procedure

Follow these steps to clean up your system.

1. Find out which IBM WebSphere MQ MQ packages still have entries in your RPM database.

```
rpm -qa | grep MQSeries
```

2. Remove all remaining IBM WebSphere MQ packages from your system.

```
rpm -e package-name
```

3. Remove the `/opt/mqm` directory.

```
rm -rf /opt/mqm
```

Linux: Reverting a queue manager to an earlier version

You can restore a queue manager to Version 7.1 from Version 7.5, if you have made a backup of the system or queue manager. If you have started the queue manager and processed any messages, or changed the configuration, the task cannot give you any guidance on restoring the current state of the queue manager.

Before you begin

1. You must have made a backup of the system or queue manager before you upgraded to Version 7.5. For more information see Backing up and restoring IBM WebSphere MQ queue manager data
2. If any messages were processed after starting the queue manager, you cannot easily undo the effects of processing the messages. You cannot restore the queue manager to Version 7.1 in its current state. The task cannot give you any guidance how to deal with subsequent changes that have occurred. For example, messages that were indoubt in a channel, or in a transmission queue on another queue manager, might have been processed. If the queue manager is part of a cluster, then configuration messages and application messages might have been exchanged.
3. If you are running on a server with multiple IBM WebSphere MQ installations, that is, IBM WebSphere MQ Version 7.0.1, Fix Pack 6 (or later), you must identify the installation. Make sure that the commands you enter run against the correct installation; see setmqenv.

About this task

When you restore a previous version of a queue manager, you restore the queue manager to its earlier code level. Queue manager data is restored to the state it was in when the queue manager was backed up.

Procedure

1. Log in as a user in `group mqm`.
2. Stop all applications using the IBM WebSphere MQ installation.

   If you use the IBM WebSphere MQ Managed File Transfer (MFT) component, ensure that any MFT agents have finished all of the file transfers that they were engaged in. There should be no incomplete transfers associated with the agents, and their SYSTEM.FTE.STATE queues should contain no messages.
3. End all the activity of queue managers associated with the IBM WebSphere MQ installation.
   a) Run the **dspmq** command to list the state of all the queue managers on the system.

Run either of the following commands from the installation that you are updating:

```
dspmq -o installation -o status
dspmq -a
```

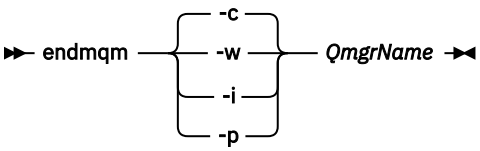
dspmq -o installation -o status displays the installation name and status of queue managers associated with all installations of IBM WebSphere MQ.

dspmq -a displays the status of active queue managers associated with the installation from which the command is run.

b) Run the **MQSC** command, `DISPLAY LSSTATUS(*) STATUS` to list the status of listeners associated with a queue manager.

```
echo "DISPLAY LSSTATUS(*) STATUS" | runmqsc QmgrName
```

c) Run the **endmqm** command to stop each running queue manager associated with this installation.



The **endmqm** command informs an application that the queue manager it is connected to is stopping; see Stopping a queue manager.

For the maintenance to proceed, applications must respond to an **endmqm** command by disconnecting from the queue manager and releasing any IBM WebSphere MQ libraries they have loaded. If they do not, you must find another way to force applications to release IBM WebSphere MQ resources, such as by stopping the applications.

You must also stop applications that are using the client libraries that are part of the installation. Client applications might be connected to a different queue manager, running a different installation of IBM WebSphere MQ. The application is not informed about queue managers in the current installation being shut down.

Any applications that continue to have IBM WebSphere MQ shared libraries from the installation loaded prevent you applying IBM WebSphere MQ maintenance. An application might disconnect from a queue manager, or be forcibly disconnected, but keep a IBM WebSphere MQ shared library loaded.

d) Stop any listeners associated with the queue managers, using the command:

```
endmqlsr -m QMgrName
```

4. Restore the system, or IBM WebSphere MQ and the queue manager.

   If your backup procedure was to save the queue manager data, you must reinstall IBM WebSphere MQ.

   a) Uninstall the Version 7.5 installation.

   b) Reinstall IBM WebSphere MQ from a manufacturing refresh.

   c) Apply the fix pack and interim fixes that restore IBM WebSphere MQ to its previous level.

   d) Restore the queue manager data from the backup taken before installing Version 7.5.

5. Restart the Version 7.1 queue manager.

What to do next

You might be restoring Version 7.1 on a server with multiple IBM WebSphere MQ installations. If one of the installations is primary, after restoring Version 7.1, the Version 7.1 installation will be by default, the primary. You must review how applications connect to an installation. After restoring Version 7.1 some applications might connect to the wrong installation.

Related tasks

Backing up and restoring a queue manager

Solaris: Migrating a queue manager to a later version

Follow these instructions to migrate a queue manager from Version 7.1 to Version 7.5.

Before you begin

1. Create a migration plan. Use the planning task, “Solaris: Planning to migrate to a later version” on page 56, as a guide.
2. Review the IBM WebSphere MQ system requirements for Version 7.5; see System Requirements for IBM WebSphere MQ.
3. Back up your system before you install IBM WebSphere MQ Version 7.5 over Version 7.1. Once you have started a queue manager you cannot revert to the previous release. If you must restore the system, you cannot recover any work, such as changes to messages and objects, performed by IBM WebSphere MQ Version 7.5. For more information about backing up your system, see Backing up and restoring IBM WebSphere MQ queue manager data .
4. Review any other installed SupportPacs for their applicability to IBM WebSphere MQ Version 7.5.
5. If you are running on a server with multiple IBM WebSphere MQ installations, that is, IBM WebSphere MQ Version 7.0.1, Fix Pack 6 (or later), you must identify the installation. Make sure that the commands you enter run against the correct installation; see setmqenv.

About this task

Complete this task to migrate a queue manager to IBM WebSphere MQ Version 7.5 from Version 7.1.

Procedure

1. Log in as a user in `group mqm`.
2. Stop all applications using the IBM WebSphere MQ installation.

   If you use the IBM WebSphere MQ Managed File Transfer (MFT) component, ensure that any MFT agents have finished all of the file transfers that they were engaged in. There should be no incomplete transfers associated with the agents, and their SYSTEM.FTE.STATE queues should contain no messages.
3. End all the activity of queue managers associated with the IBM WebSphere MQ installation.
   a) Run the **dspmq** command to list the state of all the queue managers on the system.

      Run either of the following commands from the installation that you are updating:

      ```
      dspmq -o installation -o status
      dspmq -a
      ```

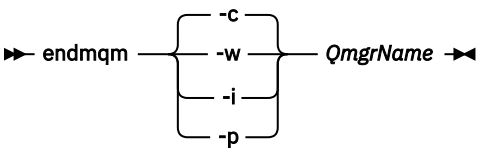
      dspmq -o installation -o status displays the installation name and status of queue managers associated with all installations of IBM WebSphere MQ.

      dspmq -a displays the status of active queue managers associated with the installation from which the command is run.

   b) Run the **MQSC** command, `DISPLAY LSSTATUS(*) STATUS` to list the status of listeners associated with a queue manager.

      ```
      echo "DISPLAY LSSTATUS(*) STATUS" | runmqsc QmgrName
      ```

   c) Run the **endmqm** command to stop each running queue manager associated with this installation.



The **endmqm** command informs an application that the queue manager it is connected to is stopping; see Stopping a queue manager.

For the maintenance to proceed, applications must respond to an **endmqm** command by disconnecting from the queue manager and releasing any IBM WebSphere MQ libraries they have loaded. If they do not, you must find another way to force applications to release IBM WebSphere MQ resources, such as by stopping the applications.

You must also stop applications that are using the client libraries that are part of the installation. Client applications might be connected to a different queue manager, running a different installation of IBM WebSphere MQ. The application is not informed about queue managers in the current installation being shut down.

Any applications that continue to have IBM WebSphere MQ shared libraries from the installation loaded prevent you applying IBM WebSphere MQ maintenance. An application might disconnect from a queue manager, or be forcibly disconnected, but keep a IBM WebSphere MQ shared library loaded.

d) Stop any listeners associated with the queue managers, using the command:

```
endmqlsr -m QMgrName
```

4. Back up the queue manager.
5. Log in as root.
6. Optionally, uninstall IBM WebSphere MQ Version 7.1.

   Note, that you carry out this step only if you are doing a single stage migration; see “UNIX, Linux, and Windows: Single-stage migration to a later version” on page 60
7. Install IBM WebSphere MQ Version 7.5

   See Installing WebSphere MQ Server on Solaris.
8. Move the queue manager to the new IBM WebSphere MQ installation.

   You need to carry out this step, only if you did not uninstall your current version of IBM WebSphere MQ.

   See “UNIX, Linux, and Windows: Side-by-side migration to a later version ” on page 65 or “UNIX, Linux, and Windows: Multi-stage migration to a later version” on page 72 for further information.
9. Start the queue manager.

   ```
   strmqm QmgrName
   ```

   When you first start a queue manager after migration:

   - Any new attributes for existing objects are set to their default values.
   - Any new default objects are created.
   - Queue manager data is migrated.

   Important: Do not use the -c option to start the queue manager, unless you explicitly want to reset or recreate the default system objects.

   You must start IBM WebSphere MQ before you start any listeners.

Backing up and restoring a queue manager

“The version naming scheme for IBM WebSphere MQ on UNIX, Linux, and Windows” on page 8
On UNIX, Linux, and Windows, IBM WebSphere MQ releases have a four-digit Version, Release, Modification, and Fix (VRMF) level code. The version and release parts of the code are significant; they identify the service life of a release. To run a queue manager at a different VR level, you must migrate the

queue manager, its applications, and the environment in which it runs. Depending on the migration path, the migration might require more or less effort.

Solaris: Reverting a queue manager to an earlier version

You can restore a queue manager to Version 7.1 from Version 7.5, if you have made a backup of the system or queue manager. If you have started the queue manager and processed any messages, or changed the configuration, the task cannot give you any guidance on restoring the current state of the queue manager.

Before you begin

1. You must have made a backup of the system or queue manager before you upgraded to Version 7.5. For more information see Backing up and restoring IBM WebSphere MQ queue manager data
2. If any messages were processed after starting the queue manager, you cannot easily undo the effects of processing the messages. You cannot restore the queue manager to Version 7.1 in its current state. The task cannot give you any guidance how to deal with subsequent changes that have occurred. For example, messages that were indoubt in a channel, or in a transmission queue on another queue manager, might have been processed. If the queue manager is part of a cluster, then configuration messages and application messages might have been exchanged.
3. If you are running on a server with multiple IBM WebSphere MQ installations, that is, IBM WebSphere MQ Version 7.0.1, Fix Pack 6 (or later), you must identify the installation. Make sure that the commands you enter run against the correct installation; see setmqenv.

About this task

When you restore a previous version of a queue manager, you restore the queue manager to its earlier code level. Queue manager data is restored to the state it was in when the queue manager was backed up.

Procedure

1. Log in as a user in `group mqm`.
2. Stop all applications using the IBM WebSphere MQ installation.

   If you use the IBM WebSphere MQ Managed File Transfer (MFT) component, ensure that any MFT agents have finished all of the file transfers that they were engaged in. There should be no incomplete transfers associated with the agents, and their SYSTEM.FTE.STATE queues should contain no messages.
3. End all the activity of queue managers associated with the IBM WebSphere MQ installation.
   a) Run the **dspmq** command to list the state of all the queue managers on the system.

      Run either of the following commands from the installation that you are updating:

      ```
      dspmq -o installation -o status
      dspmq -a
      ```

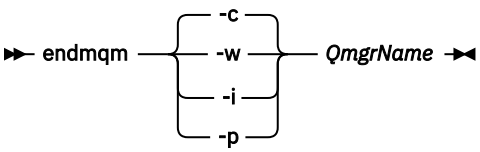
      dspmq -o installation -o status displays the installation name and status of queue managers associated with all installations of IBM WebSphere MQ.

      dspmq -a displays the status of active queue managers associated with the installation from which the command is run.

   b) Run the **MQSC** command, `DISPLAY LSSTATUS(*) STATUS` to list the status of listeners associated with a queue manager.

      ```
      echo "DISPLAY LSSTATUS(*) STATUS" | runmqsc QmgrName
      ```

   c) Run the **endmqm** command to stop each running queue manager associated with this installation.



The **endmqm** command informs an application that the queue manager it is connected to is stopping; see Stopping a queue manager.

For the maintenance to proceed, applications must respond to an **endmqm** command by disconnecting from the queue manager and releasing any IBM WebSphere MQ libraries they have loaded. If they do not, you must find another way to force applications to release IBM WebSphere MQ resources, such as by stopping the applications.

You must also stop applications that are using the client libraries that are part of the installation. Client applications might be connected to a different queue manager, running a different installation of IBM WebSphere MQ. The application is not informed about queue managers in the current installation being shut down.

Any applications that continue to have IBM WebSphere MQ shared libraries from the installation loaded prevent you applying IBM WebSphere MQ maintenance. An application might disconnect from a queue manager, or be forcibly disconnected, but keep a IBM WebSphere MQ shared library loaded.

d) Stop any listeners associated with the queue managers, using the command:

```
endmqlsr -m QMgrName
```

4. Restore the system, or IBM WebSphere MQ and the queue manager.

   If your backup procedure was to save the queue manager data, you must reinstall IBM WebSphere MQ.

   a) Uninstall the Version 7.5 installation.

   b) Reinstall IBM WebSphere MQ from a manufacturing refresh.

   c) Apply the fix pack and interim fixes that restore IBM WebSphere MQ to its previous level.

   d) Restore the queue manager data from the backup taken before installing Version 7.5.

5. Restart the Version 7.1 queue manager.

What to do next

You might be restoring Version 7.1 on a server with multiple IBM WebSphere MQ installations. If one of the installations is primary, after restoring Version 7.1, the Version 7.1 installation will be by default, the primary. You must review how applications connect to an installation. After restoring Version 7.1 some applications might connect to the wrong installation.

Related tasks

Backing up and restoring a queue manager

Windows: Migrating a queue manager to a later version

Follow these instructions to migrate a queue manager from Version 7.1 to Version 7.5.

Before you begin

1. Create a migration plan. Use the planning task, “Windows: Planning to migrate to a later version” on page 58 as a guide.
2. The migration paths to IBM WebSphere MQ Version 7.5 on Windows Vista are listed in Table 6 on page 100

Table 6. IBM WebSphere MQ Version 7.5 on Windows Vista or Windows Server 2008: supported migration paths	
From:	**To:**
IBM WebSphere MQ Version 6.0 plus fix pack 6.0.2.1 on Windows Vista or Windows Server 2008	IBM WebSphere MQ Version 7.5 on Windows Vista or Windows Server 2008
IBM WebSphere MQ Version 7.0.1 on Windows XP Professional SP2	IBM WebSphere MQ Version 7.5 on Windows Vista or Windows Server 2008

3. Review the IBM WebSphere MQ system requirements for Version 7.5; see System Requirements for IBM WebSphere MQ.
4. Back up your system before you install IBM WebSphere MQ Version 7.5 over Version 7.1. Once you have started a queue manager you cannot revert to the previous release. If you must restore the system, you cannot recover any work, such as changes to messages and objects, performed by IBM WebSphere MQ Version 7.5. For more information about backing up your system, see Backing up and restoring IBM WebSphere MQ queue manager data .
5. Review any other installed SupportPacs for their applicability to IBM WebSphere MQ Version 7.5.
6. If you are running on a server with multiple IBM WebSphere MQ installations, that is, IBM WebSphere MQ Version 7.0.1, Fix Pack 6 (or later), you must identify the installation. Make sure that the commands you enter run against the correct installation; see setmqenv.

About this task

Complete this task to migrate a queue manager to IBM WebSphere MQ Version 7.5 from Version 7.1.

All the objects that you previously created are 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. If you clear any of these components, the installation process uninstalls them. By default, a typical migration installs only the same features that were installed in the previous version installation.

For example, if IBM WebSphere MQ Explorer was not installed in the Version 7.1 installation, it is not installed in a Version 7.5 installation. If you want IBM WebSphere MQ Explorer, select a custom installation, and select the IBM WebSphere MQ Explorer feature on the **Features** panel. If you do not want IBM WebSphere MQ Explorer, uninstall the IBM WebSphere MQ Explorer feature by selecting a custom installation. Then clear the IBM WebSphere MQ Explorer feature on the **Features** panel. For more information about how to uninstall features, see Modifying the installation using WebSphere MQ Installation Launchpad.

You can also migrate a Version 7.1 queue manager to IBM WebSphere MQ Version 7.5 on a system where a previous version has been uninstalled. The queue manager data must have been retained, or restored from a backup.

Procedure

1. Log in as a user in `group mqm`.
2. Stop all applications using the IBM WebSphere MQ installation.

   If you use the IBM WebSphere MQ Managed File Transfer (MFT) component, ensure that any MFT agents have finished all of the file transfers that they were engaged in. There should be no incomplete transfers associated with the agents, and their SYSTEM.FTE.STATE queues should contain no messages.
3. End all the activity of queue managers associated with the IBM WebSphere MQ installation.
   a) Run the **dspmq** command to list the state of all the queue managers on the system.

Run either of the following commands from the installation that you are updating:

```
dspmq -o installation -o status
dspmq -a
```

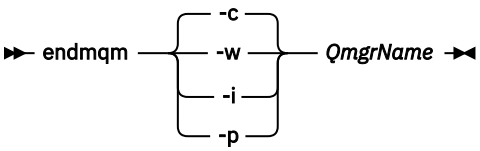
dspmq -o installation -o status displays the installation name and status of queue managers associated with all installations of IBM WebSphere MQ.

dspmq -a displays the status of active queue managers associated with the installation from which the command is run.

b) Run the **MQSC** command, `DISPLAY LSSTATUS(*) STATUS` to list the status of listeners associated with a queue manager.

```
echo "DISPLAY LSSTATUS(*) STATUS" | runmqsc QmgrName
```

c) Run the **endmqm** command to stop each running queue manager associated with this installation.



The **endmqm** command informs an application that the queue manager it is connected to is stopping; see Stopping a queue manager.

For the maintenance to proceed, applications must respond to an **endmqm** command by disconnecting from the queue manager and releasing any IBM WebSphere MQ libraries they have loaded. If they do not, you must find another way to force applications to release IBM WebSphere MQ resources, such as by stopping the applications.

You must also stop applications that are using the client libraries that are part of the installation. Client applications might be connected to a different queue manager, running a different installation of IBM WebSphere MQ. The application is not informed about queue managers in the current installation being shut down.

Any applications that continue to have IBM WebSphere MQ shared libraries from the installation loaded prevent you applying IBM WebSphere MQ maintenance. An application might disconnect from a queue manager, or be forcibly disconnected, but keep a IBM WebSphere MQ shared library loaded.

d) Stop any listeners associated with the queue managers, using the command:

```
endmqlsr -m QMgrName
```

4. Back up the queue manager.
5. Stop the MQSeries® Service and exit the Service taskbar icon application.
6. Upgrade the IBM WebSphere MQ installation to Version 7.5 using one of these procedures:
   - Modifying the installation using IBM WebSphere MQ Installation Launchpad
   - Silently modifying an IBM WebSphere MQ server installation using **msiexec**
7. Reenter domain, user ID, and password information

   When the installation of IBM WebSphere MQ Version 7.5 completes, the **Prepare WebSphere MQ Wizard** starts automatically.

   On Windows Vista or Windows Server 2008 where UAC is enabled: If you rerun the **Prepare WebSphere MQ Wizard**, ensure that the wizard is run with Administrator privilege, otherwise the wizard might fail.
8. Optionally, uninstall IBM WebSphere MQ Version 7.1.

   Note, that you carry out this step only if you are doing a single stage migration; see “UNIX, Linux, and Windows: Single-stage migration to a later version” on page 60

What to do next

You might be restoring Version 7.1 on a server with multiple IBM WebSphere MQ installations. If one of the installations is primary, after restoring Version 7.1, the Version 7.1 installation will be by default, the primary. You must review how applications connect to an installation. After restoring Version 7.1 some applications might connect to the wrong installation.

Backing up and restoring a queue manager

"The version naming scheme for IBM WebSphere MQ on UNIX, Linux, and Windows" on page 8
On UNIX, Linux, and Windows, IBM WebSphere MQ releases have a four-digit Version, Release, Modification, and Fix (VRMF) level code. The version and release parts of the code are significant; they identify the service life of a release. To run a queue manager at a different VR level, you must migrate the queue manager, its applications, and the environment in which it runs. Depending on the migration path, the migration might require more or less effort.

Windows: Reverting a queue manager to an earlier version

You can restore a queue manager to Version 7.1 from Version 7.5, if you have made a backup of the system or queue manager. If you have started the queue manager and processed any messages, or changed the configuration, the task cannot give you any guidance on restoring the current state of the queue manager.

Before you begin

1. You must have made a backup of the system or queue manager before you upgraded to Version 7.5. For more information see Backing up and restoring IBM WebSphere MQ queue manager data
2. If any messages were processed after starting the queue manager, you cannot easily undo the effects of processing the messages. You cannot restore the queue manager to Version 7.1 in its current state. The task cannot give you any guidance how to deal with subsequent changes that have occurred. For example, messages that were indoubt in a channel, or in a transmission queue on another queue manager, might have been processed. If the queue manager is part of a cluster, then configuration messages and application messages might have been exchanged.
3. If you are running on a server with multiple IBM WebSphere MQ installations, that is, IBM WebSphere MQ Version 7.0.1, Fix Pack 6 (or later), you must identify the installation. Make sure that the commands you enter run against the correct installation; see setmqenv.

About this task

When you restore a previous version of a queue manager, you restore the queue manager to its earlier code level. Queue manager data is restored to the state it was in when the queue manager was backed up.

Procedure

1. Log in as a user in `group mqm`.
2. Stop all applications using the IBM WebSphere MQ installation.

   If you use the IBM WebSphere MQ Managed File Transfer (MFT) component, ensure that any MFT agents have finished all of the file transfers that they were engaged in. There should be no incomplete transfers associated with the agents, and their SYSTEM.FTE.STATE queues should contain no messages.
3. End all the activity of queue managers associated with the IBM WebSphere MQ installation.
   a) Run the **dspmq** command to list the state of all the queue managers on the system.

      Run either of the following commands from the installation that you are updating:

```
dspmq -o installation -o status
dspmq -a
```

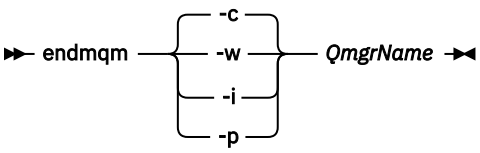
dspmq -o installation -o status displays the installation name and status of queue managers associated with all installations of IBM WebSphere MQ.

dspmq -a displays the status of active queue managers associated with the installation from which the command is run.

b) Run the **MQSC** command, `DISPLAY LSSTATUS(*) STATUS` to list the status of listeners associated with a queue manager.

```
echo "DISPLAY LSSTATUS(*) STATUS" | runmqsc QmgrName
```

c) Run the **endmqm** command to stop each running queue manager associated with this installation.



The **endmqm** command informs an application that the queue manager it is connected to is stopping; see Stopping a queue manager.

For the maintenance to proceed, applications must respond to an **endmqm** command by disconnecting from the queue manager and releasing any IBM WebSphere MQ libraries they have loaded. If they do not, you must find another way to force applications to release IBM WebSphere MQ resources, such as by stopping the applications.

You must also stop applications that are using the client libraries that are part of the installation. Client applications might be connected to a different queue manager, running a different installation of IBM WebSphere MQ. The application is not informed about queue managers in the current installation being shut down.

Any applications that continue to have IBM WebSphere MQ shared libraries from the installation loaded prevent you applying IBM WebSphere MQ maintenance. An application might disconnect from a queue manager, or be forcibly disconnected, but keep a IBM WebSphere MQ shared library loaded.

d) Stop any listeners associated with the queue managers, using the command:

```
endmqlsr -m QMgrName
```

4. Restore the system, or IBM WebSphere MQ and the queue manager.

   If your backup procedure was to save the queue manager data, you must reinstall IBM WebSphere MQ.

   a) Uninstall the Version 7.5 installation.

   b) Reinstall IBM WebSphere MQ from a manufacturing refresh.

   c) Apply the fix pack and interim fixes that restore IBM WebSphere MQ to its previous level.

   d) Restore the queue manager data from the backup taken before installing Version 7.5.

5. Restart the Version 7.1 queue manager.

What to do next

You might be restoring Version 7.1 on a server with multiple IBM WebSphere MQ installations. If one of the installations is primary, after restoring Version 7.1, the Version 7.1 installation will be by default, the primary. You must review how applications connect to an installation. After restoring Version 7.1 some applications might connect to the wrong installation.

Related tasks

Backing up and restoring a queue manager

Migrating an IBM WebSphere MQ MQI client to a later version UNIX, Linux, and Windows

Migrate a IBM WebSphere MQ MQI client to a later version of the product by completing the tasks in the following topics. Check IBM WebSphere MQ MQI client applications with Version 7.5 before migration. You must stop all IBM WebSphere MQ activity on the workstation, before upgrading the IBM WebSphere MQ MQI client. After upgrading the IBM WebSphere MQ MQI client, you must check the client channel configuration.

AIX: Migrating an IBM WebSphere MQ MQI client to a later version

Before migrating a IBM WebSphere MQ MQI client, create a migration plan. Stop all IBM WebSphere MQ activity on the client workstation. Upgrade the IBM WebSphere MQ MQI client installation. Make any essential configuration and application changes.

Before you begin

1. Create a migration plan. Use the planning task, “AIX: Planning to migrate to a later version” on page 52, as a guide.

Procedure

1. Review the WebSphere MQ system requirements for Version 7.5.

   See System Requirements for IBM WebSphere MQ.
2. Review all the changes in WebSphere MQ that affect you.

   See changes from “Changes in IBM WebSphere MQ Version 7.5” on page 138.
3. End all IBM WebSphere MQ activity on the workstation.
4. Upgrade the client.
   - To upgrade a IBM WebSphere MQ MQI client for AIX installation on a workstation; see Client installation procedure on an AIX workstation.
   - To upgrade a IBM WebSphere MQ MQI client for AIX installation on an AIX IBM WebSphere MQ server; see Installing a IBM WebSphere MQ MQI client on the same computer as the server.

What to do next

Complete the tasks in your migration plan, such as verifying IBM WebSphere MQ MQI client applications work correctly with Version 7.5.

Related tasks

Client installation procedure on an AIX workstation

“AIX: Planning to migrate to a later version” on page 52

Create a migration plan to migrate from your current version of IBM WebSphere MQ to a later version. You must back up your queue manager data before you migrate to the later version. You cannot revert a queue manager to the earlier version of the product once you start it on the later version.

Installing WebSphere MQ MQI clients on the same machine as the server

HP-UX: Migrating an IBM WebSphere MQ MQI client to a later version

Before migrating a IBM WebSphere MQ MQI client, create a migration plan. Stop all IBM WebSphere MQ activity on the client workstation. Upgrade the IBM WebSphere MQ MQI client installation. Make any essential configuration and application changes.

Before you begin

1. Create a migration plan. Use the planning task, "HP-UX: Planning to migrate to a later version" on page 53, as a guide.

Procedure

1. Review the WebSphere MQ system requirements for Version 7.5.

   See System Requirements for IBM WebSphere MQ.
2. Review all the changes in WebSphere MQ that affect you.

   See changes from "Changes in IBM WebSphere MQ Version 7.5" on page 138.
3. End all IBM WebSphere MQ activity on the workstation.
4. Upgrade the client.
   - To upgrade a IBM WebSphere MQ MQI client for HP-UX installation on a workstation; see Client installation procedure on an HP-UX workstation.
   - To upgrade a IBM WebSphere MQ MQI client for HP-UX installation on an HP-UX IBM WebSphere MQ server; see Installing an IBM WebSphere MQ MQI client on the same computer as the server.

What to do next

Complete the tasks in your migration plan, such as verifying IBM WebSphere MQ MQI client applications work correctly with Version 7.5.

Related tasks

Client installation procedure on an HP-UX workstation

"HP-UX: Planning to migrate to a later version" on page 53

Create a migration plan to migrate from your current version of IBM WebSphere MQ to a later version. You must back up your queue manager data before you migrate to the later version. You cannot revert a queue manager to the earlier version of the product once you start it on the later version.

Linux: Migrating an IBM WebSphere MQ MQI client to a later version

Before migrating a IBM WebSphere MQ MQI client, create a migration plan. Stop all IBM WebSphere MQ activity on the client workstation. Upgrade the IBM WebSphere MQ MQI client installation. Make any essential configuration and application changes.

Before you begin

1. Create a migration plan. Use the planning task, "Linux: Planning to migrate to a later version" on page 55, as a guide.

Procedure

1. Review the WebSphere MQ system requirements for Version 7.5.

   See System Requirements for IBM WebSphere MQ.
2. Review all the changes in WebSphere MQ that affect you.

   See changes from "Changes in IBM WebSphere MQ Version 7.5" on page 138.
3. End all IBM WebSphere MQ activity on the workstation.

4. Upgrade the client.
   - To upgrade a IBM WebSphere MQ MQI client for Linux installation on a workstation; see Client installation procedure on an Linux workstation.
   - To upgrade a IBM WebSphere MQ MQI client for Linux installation on an Linux IBM WebSphere MQ server; see Installing an IBM WebSphere MQ MQI client on the same computer as the server.

What to do next

Complete the tasks in your migration plan, such as verifying IBM WebSphere MQ MQI client applications work correctly with Version 7.5.

Related tasks

Client installation procedure on a Linux workstation

"Linux: Planning to migrate to a later version" on page 55

Create a migration plan to migrate from your current version of IBM WebSphere MQ to a later version. You must back up your queue manager data before you migrate to the later version. You cannot revert a queue manager to the earlier version of the product once you start it on the later version.

Solaris: Migrating an IBM WebSphere MQ MQI client to a later version

Before migrating a IBM WebSphere MQ MQI client, create a migration plan. Stop all IBM WebSphere MQ activity on the client workstation. Upgrade the IBM WebSphere MQ MQI client installation. Make any essential configuration and application changes.

Before you begin

1. Create a migration plan. Use the planning task, "Solaris: Planning to migrate to a later version" on page 56, as a guide.

Procedure

1. Review the WebSphere MQ system requirements for Version 7.5.

   See System Requirements for IBM WebSphere MQ.
2. Review all the changes in WebSphere MQ that affect you.

   See changes from "Changes in IBM WebSphere MQ Version 7.5" on page 138.
3. End all IBM WebSphere MQ activity on the workstation.
4. Upgrade the client.
   - To upgrade a IBM WebSphere MQ MQI client for Solaris installation on a workstation; see Client installation procedure on an Solaris workstation.
   - To upgrade a IBM WebSphere MQ MQI client for Solaris installation on an Solaris IBM WebSphere MQ server; see Installing an IBM WebSphere MQ MQI client on the same computer as the server.

What to do next

Complete the tasks in your migration plan, such as verifying IBM WebSphere MQ MQI client applications work correctly with Version 7.5.

Related tasks

Client installation procedure on a Solaris workstation

"Solaris: Planning to migrate to a later version" on page 56

Create a migration plan to migrate from your current version of IBM WebSphere MQ to a later version. You must back up your queue manager data before you migrate to the later version. You cannot revert a queue manager to the earlier version of the product once you start it on the later version.

Windows: Migrating an IBM WebSphere MQ MQI client to a later version

Before migrating a IBM WebSphere MQ MQI client, create a migration plan. Stop all IBM WebSphere MQ activity on the client workstation. Upgrade the IBM WebSphere MQ MQI client installation. Make any essential configuration and application changes.

Before you begin

1. Create a migration plan. Use the planning task, "Windows: Planning to migrate to a later version" on page 58, as a guide.

Procedure

1. Review the WebSphere MQ system requirements for Version 7.5.

   See System Requirements for IBM WebSphere MQ.
2. Review all the changes in WebSphere MQ that affect you.

   See changes from "Changes in IBM WebSphere MQ Version 7.5" on page 138.
3. End all IBM WebSphere MQ activity on the workstation.
4. Upgrade the client.
   - To upgrade a IBM WebSphere MQ MQI client for Windows installation on a workstation; see Client installation procedure on an Windows workstation.
   - To upgrade a IBM WebSphere MQ MQI client for Windows installation on an Windows IBM WebSphere MQ server; see Installing an IBM WebSphere MQ MQI client on the same computer as the server.

What to do next

Complete the tasks in your migration plan, such as verifying IBM WebSphere MQ MQI client applications work correctly with Version 7.5.

Related tasks

Client installation procedure on a Windcws workstation

"Windows: Planning to migrate to a later version" on page 58

Create a migration plan to migrate from your current version of IBM WebSphere MQ to a later version. You must back up your queue manager data before you migrate to the later version. You cannot revert a queue manager to the earlier version of the product once you start it on the later version.

Reverting an IBM WebSphere MQ MQI client and client connection to an earlier version

If you restore a Version 7.5 IBM WebSphere MQ MQI client to the Version 7.1 code level, you must undo the configuration changes manually.

About this task

It is unusual to restore earlier IBM WebSphere MQ MQI client libraries to a workstation. The principal tasks are listed in the following steps.

Procedure

1. End all IBM WebSphere MQ activity on the workstation.
2. Uninstall the Version 7.5 IBM WebSphere MQ MQI client code.

3. Follow the client installation procedure for the platform to install the Version 7.1IBM WebSphere MQ MQI client.
4. If you configured a Client Connection Definition Table (CCDT) from a Version 7.5 queue manager, revert to using a table created by a Version 7.1 queue manager.

   The CCDT must always be created by a queue manager on the same, or earlier, release to the client.

Migrating applications to a later version

IBM WebSphere MQ applications do not require migration between Version 7.1 to Version 7.5.

Migrating IBM WebSphere MQ library loading from Version 7.0.1 to Version 7.5

No change in the way IBM WebSphere MQ libraries are loaded is normally required if you upgrade from Version 7.0.1 to Version 7.5. You must have followed the instructions on building IBM WebSphere MQ applications in Version 7.0.1 and you must replace IBM WebSphere MQ Version 7.0.1 with IBM WebSphere MQ Version 7.5. If you choose to take advantage of multi-installation in IBM WebSphere MQ Version 7.5, based on the side-by-side or multi-stage migration scenarios, you must modify the environment for the operating system to resolve IBM WebSphere MQ dependencies for an application. Typically, you can modify the runtime environment, rather than relink the application.

Before you begin

To migrate applications from Version 7.0.1 to Version 7.5, you must know how the operating system loads an IBM WebSphere MQ library for an application. Is the load path fixed by the application, and can you set the path in an environment variable? It is not essential to know the name of the IBM WebSphere MQ library that the application loads. The library name does not change from Version 7.0.1 to Version 7.5, although the contents of the library do.

About this task

To migrate an application from Version 7.0.1 to Version 7.5, you do not have to recompile or relink the application, because the IBM WebSphere MQ libraries are compatible with later versions; see “Application compatibility and interoperability with later versions of IBM WebSphere MQ” on page 37. You might have to configure the runtime environment differently, for the operating system to load the IBM WebSphere MQ Version 7.5 library. If you replaced Version 7.0.1 with Version 7.5, following the "Single-stage" approach; you do not need to do anything at all; see “UNIX, Linux, and Windows: Single-stage migration to a later version” on page 60[9].

IBM WebSphere MQ Version 7.5 provides two commands to assist you in configuring the runtime environment, **setmqinst** and **setmqenv**. **setmqinst** sets the primary installation; see setmqinst. **setmqenv** initializes the command environment by setting environment variables; see setmqenv.

Procedure

How you configure the runtime environment depends on a number of factors, some of which apply to your configuration. Consider which of the following questions apply to your configuration.

1. Did you follow the build procedure documented in the IBM WebSphere MQ Version 7.0.1 product documentation?

   You might be following a different build procedure tailored to your development environment, or adapted from a development tool such as Microsoft Visual Studio.
2. How did you specify the load path Version 7.0.1?
3. Is the application is loaded by another environment, such as Eclipse, or an application server?

[9] If you changed the location of the IBM WebSphere MQ Version 7.0.1 libraries, or created symbolic links to the libraries, this statement might not hold true.

You must modify the parameters that govern how the parent environment loads applications, not the way the parent environment is loaded.

4. Is the configuration for Windows, or UNIX and Linux?

   On Windows, the functions performed by an application might require that the queue manager it connects to is associated with the primary installation.

5. What constraints and requirements do you have on how the load path is specified in Version 7.5?

   Security rules might restrict the use of `LD_LIBRARY_PATH`.

6. Is Version 7.5 installed alongside Version 7.0.1?

   If Version 7.0.1 is installed:

   - You cannot make a Version 7.5 installation primary.
   - You cannot install Version 7.5 in the default installation path, that was referenced by applications in Version 7.0.1.

What to do next

Environment configuration on the Windows, and UNIX platforms is a little different. If you have followed the documented build procedure in Version 7.0.1, look at "Windows: Migrating IBM WebSphere MQ library loading from Version 7.0.1 to Version 7.5" on page 110 or "UNIX and Linux: Migrating IBM WebSphere MQ library loading from Version 7.0.1 to Version 7.5" on page 112. These topics show the effects of using setmqinst and setmqenv commands to configure the operating system environment for the three migration scenarios listed in Related information.

Related tasks

Changing the primary installation

Connecting applications in a multiple installation environment

"UNIX, Linux, and Windows: Single-stage migration to a later version" on page 60
Single-stage migration is the term used to describe replacing the only installation of IBM WebSphere MQ on a server, with a later release. Single stage migration is also known as `upgrading in place` or `in place upgrade`. Until Version 7.0.1.6, single-stage was the only migration scenario. Single-stage migration preserves existing scripts and procedures for running IBM WebSphere MQ the most. With other migration scenarios you might change some scripts and procedures, but you can reduce the effect queue manager migration has on users.

"UNIX, Linux, and Windows: Side-by-side migration to a later version " on page 65

"UNIX, Linux, and Windows: Multi-stage migration to a later version" on page 72

"UNIX and Linux: Migrating IBM WebSphere MQ library loading from Version 7.0.1 to Version 7.5" on page 112
Investigate whether applications connecting to IBM WebSphere MQ for Windows Version 7.5 are linked to, and load libraries from, the correct installation.

"Windows: Migrating IBM WebSphere MQ library loading from Version 7.0.1 to Version 7.5" on page 110
Investigate whether applications connecting to IBM WebSphere MQ for Windows Version 7.5 are linked to, and load libraries from, the correct installation.

Related reference

"Coexistence" on page 137
Queue managers, with different names, can coexist on any server as long as they use the same IBM WebSphere MQ installation. On UNIX, Linux, and Windows, different queue managers can coexist on the same server and be associated with different installations. In addition to queue managers coexisting on a server, objects, and commands must work correctly with different queue managers running at different command levels.

setmqenv

setmqinst

setmqm

Windows: Migrating IBM WebSphere MQ library loading from Version 7.0.1 to Version 7.5

Investigate whether applications connecting to IBM WebSphere MQ for Windows Version 7.5 are linked to, and load libraries from, the correct installation.

Before you begin

Read “Multi-installation queue manager coexistence on UNIX, Linux, and Windows” on page 31 and “Migrating IBM WebSphere MQ library loading from Version 7.0.1 to Version 7.5” on page 108 before starting this task.

Plan and install IBM WebSphere MQ for Windows Version 7.5, and remember the installation name and whether the installation was set to primary.

About this task

Windows searches numerous directories for load libraries, called DLLs; see Dynamic-Link Library Search Order.

The build procedure documented for IBM WebSphere MQ Version 7.0.1 applications is to place the IBM WebSphere MQ libraries to load before any other product libraries in the **cl** command. The IBM WebSphere MQ `.lib` libraries must be in the PATH environment variable you have specified at build time, and the DLL libraries at run time. The PATH variable is used by the application process to find the libraries it must load. If you have followed this build procedure, then the effect of installing Version 7.5 on the libraries that are loaded depends on the migration scenario; see Table 7 on page 110.

Table 7. Windows configurations

Action	Scenario	Version 7.5 replaces Version 7.0.1 in the same location "Single-stage"	Version 7.5 replaces Version 7.0.1 in a different location "Side-by-side"	Version 7.5 alongside Version 7.0.1 "Multi-stage"
setmqinst		**setmqinst** makes the Version 7.5 installation primary. The global PATH is changed to point to the Version 7.5 library and all Windows features work with Version 7.5[See note].		No Version 7.5 installation can be primary, because Version 7.0.1 is installed.
No other configuration actions		Library loading works correctly. The global PATH contains the location of the Version 7.5 libraries. Even if the Version 7.5 installation is not primary, library loading works correctly. The Version 7.5 libraries are in the same location as the Version 7.0.1 libraries were.	Library loading probably works correctly. The library loading might not work, if the application process locally modified the PATH to reference the location of the Version 7.0.1 libraries. A local setting of PATH might override the global PATH that is set by **setmqinst**.	The library loading continues to work with Version 7.0.1 correctly, nothing works with Version 7.5.

Table 7. Windows configurations (continued)				
Action	**Scenario**	**Version 7.5 replaces Version 7.0.1 in the same location** **"Single-stage"**	**Version 7.5 replaces Version 7.0.1 in a different location** **"Side-by-side"**	**Version 7.5 alongside Version 7.0.1** **"Multi-stage"**
setmqenv		Library loading works correctly. **setmqenv** sets the local PATH correctly.		Library loading works correctly, both for Version 7.0.1 and Version 7.5. **setmqenv** sets the local PATH correctly for Version 7.5. But the Windows features that depend on the global path do not work correctly with Version 7.5 See note. The correct Version 7.0.1 is loaded, because the Version 7.5 library loads the Version 7.0.1 library for queue managers that have not been migrated from Version 7.0.1.

Procedure

Identify which Version 7.5 installation the operating system is going to load IBM WebSphere MQ libraries from:

- If you have a multiple Version 7.5 installations to load from on a server, IBM WebSphere MQ checks that the installation the library was loaded from is the installation that is associated with any queue manager the application calls. IBM WebSphere MQ loads the correct library if the wrong library is loaded. It is necessary to configure only one runtime environment for all IBM WebSphere MQapplications.
- A typical choice is set the primary installation. Setting an installation to be primary places its library path in the global PATH variable.
- If you upgraded a Version 7.0.1 installation to Version 7.5, a link path to the Version 7.0.1 installation now points to a Version 7.5 installation. Applications that have a fixed linkage path to the Version 7.0.1 installation now load the Version 7.5 libraries. They are then switched to the installation that is associated with any queue manager they connect to.
- If you rebuild an application, it must link to a Version 7.5 installation.
- If an application uses COM or ActiveX it can connect to any queue manager as long as there is a primary installation and it is Version 7.5 or later.

  Note: If Version 7.0.1 is installed, COM or ActiveX server applications connect to queue managers associated only with the Version 7.0.1 installation. COM or ActiveX client applications are not affected by the limitation.
- If you are running the IBM WebSphere MQ .NET monitor in transactional mode, the queue manager it connects to must be the primary installation.

What to do next

If you add further Version 7.5 installations, you must decide which installation to make primary, if you have chosen to make any primary. As long as applications load IBM WebSphere MQ libraries from one of the Version 7.5 installations, such as the primary installation, they can connect to queue managers associated with any other Version 7.5 installation.

On Windows, you might build applications with different development tools. You must identify the property of the development tool that sets the PATH of the application that is being built, and not the properties of the tool itself. For example, if you are debugging with Microsoft Visual Studio, you can insert a call to **setmqenv** in the **Environment** property of the debugging section of the **Configuration** properties of a project.

A Windows application might call `LoadLibrary` and specify an explicit load path. You might build a side-by-side assembly and configure an explicit load path. If an application uses either of these mechanisms, and the Version 7.5 IBM WebSphere MQ library is not on the same path as the earlier release, you must recompile, or configure and relink your application to load the Version 7.5 libraries.

Related tasks

"UNIX, Linux, and Windows: Single-stage migration to a later version" on page 60
Single-stage migration is the term used to describe replacing the only installation of IBM WebSphere MQ on a server, with a later release. Single stage migration is also known as `upgrading in place` or `in place upgrade`. Until Version 7.0.1.6, single-stage was the only migration scenario. Single-stage migration preserves existing scripts and procedures for running IBM WebSphere MQ the most. With other migration scenarios you might change some scripts and procedures, but you can reduce the effect queue manager migration has on users.

"UNIX, Linux, and Windows: Side-by-side migration to a later version " on page 65

"UNIX, Linux, and Windows: Multi-stage migration to a later version" on page 72

Related reference

"Coexistence" on page 137
Queue managers, with different names, can coexist on any server as long as they use the same IBM WebSphere MQ installation. On UNIX, Linux, and Windows, different queue managers can coexist on the same server and be associated with different installations. In addition to queue managers coexisting on a server, objects, and commands must work correctly with different queue managers running at different command levels.

setmqenv

setmqinst

setmqm

Related information

Changing the primary installation

Connecting applications in a multiple installation environment

Features that can be used only with the primary installation on Windows

UNIX and Linux: Migrating IBM WebSphere MQ library loading from Version 7.0.1 to Version 7.5

Investigate whether applications connecting to IBM WebSphere MQ for Windows Version 7.5 are linked to, and load libraries from, the correct installation.

Before you begin

Read "Multi-installation queue manager coexistence on UNIX, Linux, and Windows" on page 31 and "Migrating IBM WebSphere MQ library loading from Version 7.0.1 to Version 7.5" on page 108 before starting this task.

Plan and install IBM WebSphere MQ for Windows Version 7.5, and remember the installation name and whether the installation was set to primary.

About this task

In Version 6.0 and Version 7.0.1, the documented build procedure for IBM WebSphere MQ applications is to include an explicit library path to the location of the IBM WebSphere MQ libraries, and to `/usr/lib`, in the link step of the compiler; see Figure 24 on page 113. The same build procedure is documented

for Version 7.5. Previous versions of IBM WebSphere MQ, before Version 6.0, did not document a specific location to load run time libraries from.

```
gcc -m32 -o amqsput_32_r amqsput0.c -I/opt/mqm/inc -L/opt/mqm/lib
-Wl,-rpath=/opt/mqm/lib -Wl,-rpath=/usr/lib -lmqm_r -lpthread
```

Figure 24. Linux C server application, 32 bit, threaded compile and link Version 7.0.1

The documented build step for other UNIX platforms is similar. The examples in Table 8 on page 113 are all based on Linux.

If you have followed this build procedure, then the effect of installing Version 7.5 on library loading depends on the migration scenario; see Table 8 on page 113:

Table 8. UNIX and Linux configurations

Action	Scenario	Version 7.5 replaces Version 7.0.1 in the same location "Single-stage"	Version 7.5 replaces Version 7.0.1 in a different location "Side-by-side"	Version 7.5 alongside Version 7.0.1 "Multi-stage"
setmqinst		**setmqinst** makes the Version 7.5 installation primary. Symbolic links to the IBM WebSphere MQ link libraries are inserted into `/usr/lib`.		No Version 7.5 installation can be primary, because Version 7.0.1 is installed.
No other configuration actions		Library loading works correctly. Library loading works, even without the Version 7.5 installation being made primary, because the libraries are installed in `/opt/mqm/lib` and the application was built with the link option, `-rpath=/opt/mqm/lib`	Library loading works correctly. Library loading works, because the installation is primary, and the application was built with the link option, `-rpath=/usr/lib`.	The library loading continues to work with Version 7.0.1 correctly, nothing works with Version 7.5.
setmqenv, without setting the -k or -l options.		Library loading works correctly. **setmqenv** is unnecessary. Library loading works, because the libraries are installed in `/opt/mqm/lib` and the application was built with the link option, `-rpath=/opt/mqm/lib`.	Library loading works correctly. **setmqenv** is unnecessary. Library loading works, because the installation is primary, and the application was built with the link option, `-rpath=/usr/lib`.	The library loading continues to work with Version 7.0.1 correctly, nothing works with Version 7.5.

Table 8. UNIX and Linux configurations (continued)				
Action	**Scenario**	**Version 7.5 replaces Version 7.0.1 in the same location** **"Single-stage"**	**Version 7.5 replaces Version 7.0.1 in a different location** **"Side-by-side"**	**Version 7.5 alongside Version 7.0.1** **"Multi-stage"**
setmqenv, with the -k or -l options set.		Library loading works correctly.		Library loading works correctly, both for Version 7.0.1 and Version 7.5. The correct Version 7.0.1 is loaded, because the Version 7.5 library loads the Version 7.0.1 library for queue managers that have not been migrated from Version 7.0.1.
		The operating system finds the IBM WebSphere MQ library location set by **setmqenv**. **setmqenv** adds the location to LD_LIBRARY_PATH[10]. LD_LIBRARY_PATH is searched before paths set in the application or paths in the default search path. Not all applications can load a library using LD_LIBRARY_PATH. In which case the application works only if the library location is /opt/mqm/lib or /usr/lib		

Procedure

Identify which Version 7.5 installation the operating system is going to load IBM WebSphere MQ libraries from:

- If you have a multiple Version 7.5 installations to load from on a server, IBM WebSphere MQ checks that the installation the library was loaded from is the installation that is associated with any queue manager the application calls. IBM WebSphere MQ loads the correct library if the wrong library is loaded. It is necessary to configure only one runtime environment for all IBM WebSphere MQapplications.
- A typical choice is to set the primary installation. Setting an installation to be primary places symbolic links to the IBM WebSphere MQ libraries in /usr/lib. Applications built following the Version 6.0 or Version 7.0 instructions have an explicit link to /usr/lib. /usr/lib is also normally in the default library search path.
- If you upgraded a Version 7.0.1 installation to Version 7.5, a link path to the Version 7.0.1 installation now points to a Version 7.5 installation. Applications that have a fixed linkage path to the Version 7.0.1 installation now load the Version 7.5 libraries. They are then switched to the installation that is associated with any queue manager they connect to.
- If you rebuild an application, it must link to a Version 7.5 installation.
- If you set LD_LIBRARY_PATH, or LIBPATH on AIX, you must check that the application is able to use LD_LIBRARY_PATH. setuid or setgid, applications, or applications built in other ways, might ignore LD_LIBRARY_PATH for security reasons.

What to do next

If you add further Version 7.5 installations, you must decide which installation to make primary, if you have chosen to make any primary. As long as applications load IBM WebSphere MQ libraries from one of the Version 7.5 installations, such as the primary installation, they can connect to queue managers associated with any other Version 7.5 installation.

[10] LIBPATH on AIX. On HP-UX LD_LIBRARY_PATH is set, not SHLIB_PATH.

Related tasks

"UNIX, Linux, and Windows: Single-stage migration to a later version" on page 60
Single-stage migration is the term used to describe replacing the only installation of IBM WebSphere MQ on a server, with a later release. Single stage migration is also known as `upgrading in place` or `in place upgrade`. Until Version 7.0.1.6, single-stage was the only migration scenario. Single-stage migration preserves existing scripts and procedures for running IBM WebSphere MQ the most. With other migration scenarios you might change some scripts and procedures, but you can reduce the effect queue manager migration has on users.

"UNIX, Linux, and Windows: Side-by-side migration to a later version " on page 65

"UNIX, Linux, and Windows: Multi-stage migration to a later version" on page 72

Related reference

"Coexistence" on page 137
Queue managers, with different names, can coexist on any server as long as they use the same IBM WebSphere MQ installation. On UNIX, Linux, and Windows, different queue managers can coexist on the same server and be associated with different installations. In addition to queue managers coexisting on a server, objects, and commands must work correctly with different queue managers running at different command levels.

setmqenv

setmqinst

setmqm

Related information

External library and control command links to primary installation on UNIX and Linux

Connecting applications in a multiple installation environment

Changing the primary installation

Loading WebSphere MQ Version 7.1 libraries

Linux: Rebuilding a C++ application

C++ IBM WebSphere MQ MQI client and server applications on Linux must be recompiled using GNU Compiler Collection (GCC) 4.1.2, or later. Compilers older than GCC 4.1.2 are no longer supported. The C++ GCC 4.1.2 run time libraries, or later, must be installed in `/usr/lib` or `/usr/lib64`

If you are using one of the supported Linux distributions, the libraries are correctly installed; see System Requirements for IBM WebSphere MQ.

The GCC 4.1.2 libraries support SSL and TLS connections from a IBM WebSphere MQ MQI client. SSL and TLS use GSKit version 8, which depends on `libstdc++.so.6`. `libstdc++.so.6` is included in GCC 4.1.2.

Before you begin

1. Check the required level of GCC for your distribution of Linux; see System Requirements for IBM WebSphere MQ.
2. If you are using SSL or TLS, also check the required level of `libstdc++.so`.
3. Check whether the application requires rebuilding. Run the following command to display what version of `libstdc++.so` the application depends upon. If the result is less than `libstdc++.so.6`, you must rebuild your application.

```
ldd ApplicationPath
```

About this task

The task describes the steps required to rebuild a Linux C++ IBM WebSphere MQ application. For more detailed instructions about building Linux applications for IBM WebSphere MQ; see Building your application on Linux

Procedure

1. Check that the required GCC library is installed correctly.

   Run one of the following commands:

   - Check the 32 bit library on an x86 Linux system:

     ```
     ls -l /usr/lib/libstdc++.so.6
     ```

   - Check the 64 bit library on any other Linux system.

     ```
     ls -l /usr/lib64/libstdc++.so.6
     ```

2. Check that the GCC compiler is at least at version 4.1.2

   Run the following command to display the version of GCC.

   ```
   gcc -v
   ```

3. Rebuild the application

   The commands to compile and link Linux C++ applications are described in Building 32-bit applications and Building 64-bit applications

What to do next

When you deploy your Linux C++ application, ensure that the same GCC runtime library is correctly installed on the run time system.

Migrating a queue manager cluster

Migrate a queue manager cluster by migrating each of the queue managers in the cluster. No changes since Version 6.0 specifically effect the migration of queue manager clusters. But you must consider what the effect is of migrating some queue managers in cluster, before all the queue managers are migrated.

Before you begin

Check that no cluster-specific migration issues are identified for the migration you are intending to perform. No cluster-specific changes have been made that affect migration between Version 6.0 and Version 7.1, and between Version 7.0.1 and Version 7.1.

Procedure

Consider the following issues that 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 the systems architecture of your organization.

Steps

Some techniques for migrating a cluster with minimal planned and unplanned outages are described in the following topics:

Related concepts

“How mixed version cluster repositories are updated” on page 43
Repositories store records for an object in a cluster in the version of the record format that matches the version of the queue manager hosting the repository. Repository queue managers forward object records,

before they are stored, in the format that they are received in. The recipient ignores fields from a newer version, and uses default values for fields that are not present in the record.

"Queue manager cluster migration" on page 42
You can migrate queue managers in a cluster all at once, or one at a time, which is called a staged migration. Migrate full repository queue managers in a cluster before partial repository queue managers.

Related information

Migrating WebSphere MQ queue manager clusters from WebSphere MQ V6 to V7

Migrating a queue manager cluster: Create a plan

Before carrying out the migration of a queue manager cluster, plan what you are going to do. Identify the roles that different queue managers play in the cluster, and decide in what order to migrate the queue managers.

Procedure

- What queue manager and application migration issues must be dealt with between the old and new versions?
- What system architecture and change control procedures must you consider?
- Consider migration questions specific to clusters, such as migrating full repositories first, and migrating overlapping clusters.
- Are any of the queue managers in a queue-sharing group, or part of a high-availability solution?
- Is the cluster a publish/subscribe cluster? Which queue manager is a cluster topic host?
- Decide whether to carry out a staged migration, or migrate all queue managers at the same time.
- Do you have a test system to migrate, and a production system?
- Document and test the plan before migrating production queue managers.

Related concepts

"Application migration and interoperation" on page 41
IBM WebSphere MQ supports running applications compiled and linked against previous versions of IBM WebSphere MQ with later levels of IBM WebSphere MQ.

Key roles for publish/subscribe cluster queue managers

"Migrate a queue manager in a high availability configuration" on page 44
Follow standard procedures to migrate a queue manager that is part of a high availability configuration on platforms other than z/OS.

"How mixed version cluster repositories are updated" on page 43
Repositories store records for an object in a cluster in the version of the record format that matches the version of the queue manager hosting the repository. Repository queue managers forward object records, before they are stored, in the format that they are received in. The recipient ignores fields from a newer version, and uses default values for fields that are not present in the record.

"Queue manager cluster migration" on page 42
You can migrate queue managers in a cluster all at once, or one at a time, which is called a staged migration. Migrate full repository queue managers in a cluster before partial repository queue managers.

"Queue manager migration" on page 38

Migrating a queue manager cluster: Create a backout plan

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

Before you begin

What backout capabilities do the queue managers in the cluster support? If no new function at the new level has been used, queue managers running on z/OS can be restored to an earlier level. On other

platforms, the only backout option is to restore a queue manager to a previous state. In restoring a queue manager, you lose any persistent changes since the queue manager started running at the new level.

About this task

The backout plan must consider how to maintain the availability of the cluster. It must deal with any issues arising from migrating a queue manager in the cluster.

Procedure

The backout plan must describe the following points:

- What constitutes a successful migration.
- The conditions that trigger the backout procedure.
- Alternative backout actions, such as:
  a) Suspending a queue manager from the cluster.
  b) Backward migration
  c) Keeping a queue manager offline until an external issue is resolved.

Related concepts

"Reverting a queue manager to a previous version" on page 39
On platforms other than z/OS it is not possible to reverse a migration. You can remove an upgrade before you have started a queue manager. After a queue manager has been started, if you remove the upgrade, the queue manager will not work.

Migrating a queue manager cluster: Migrating one cluster queue manager

Follow these steps to migrate a single queue manager in a cluster. Base your cluster migration plan on applying these steps to each queue manager in the cluster.

Procedure

1. Suspend the queue manager that you want to migrate from the cluster:
   a) Issue the **MQSC** command:

   ```
   SUSPEND QMGR CLUSTER(cluster name)
   ```

   b) Check that no messages are sent to the queue manager.

   You must close any application that continues to send messages to this queue manager. The cluster workload algorithm might choose the suspended queue manager. If there are no other valid destinations, or if an application has an affinity with the queue manager, it might select the queue manager.
2. Save a record of all cluster objects known by this queue manager. This data is used after migration to check that objects have been migrated successfully.
   a) Issue the command to view cluster queue managers.

   ```
   DISPLAY CLUSQMGR(*)
   ```

   b) Issue the command to view cluster queues.

   ```
   DISPLAY QC(*)
   ```
3. Save a record from the full repository of its view of the cluster objects owned by this queue manager. The record is used after migration to check that objects have been migrated successfully.
   a) Issue the command on the full repositories to display this queue manager.

   ```
   DISPLAY CLUSQMGR(migrated queue manager name)
   ```

b) Issue the command on the full repositories to display the cluster queues for this queue manager

```
DISPLAY QC(*) WHERE(CLUSQMGR EQ migrated queue manager name)
```

4. Migrate the queue manager.

   Do one of the queue manager migration tasks, depending on the platform; see “Migrating IBM WebSphere MQ to a later version on UNIX, Linux, and Windows ” on page 83.

   The queue manager migration process is, in outline:

   a) Stop the queue manager.

   b) Take a backup of the queue manager.

   c) Install the new version of WebSphere MQ.

   d) Restart the queue manager.

5. Ensure that all cluster objects have been migrated successfully.

   a) Issue the command to view cluster queue managers, and check the output against the data saved before migration.

   ```
   DISPLAY CLUSQMGR(*)
   ```

   b) Issue the command to view cluster queues and check the output against the data saved before migration.

   ```
   DISPLAY QC(*)
   ```

6. Check that the queue manager is communicating with the full repositories correctly.
7. Check that cluster channels to full repositories can start.
8. Check that the full repositories still have information about the migrated cluster queue manager and its cluster queues.

   a) Issue the command on the full repositories and check the output against the data saved before migration.

   ```
   DISPLAY CLUSQMGR(migrated_queue_manager_name)
   ```

   b) Issue the command on the full repositories and check the output against the data saved before migration.

   ```
   DISPLAY QC(*) WHERE(CLUSQMGR EQ migrated_queue_manager_name)
   ```

9. Test that applications on other queue managers can put messages to queues owned by the migrated cluster queue manager.
10. Test that applications on the migrated queue manager can put messages to the queues owned by other cluster queue managers.
11. Resume the queue manager.

    a) Issue RESUME QMGR CLUSTER(*cluster name*)

12. Closely monitor the queue manager and applications in the cluster for a while.

Related concepts

“Queue manager migration” on page 38

Related reference

DISPLAY CLUSQMGR

DISPLAY QUEUE

RESUME QMGR

SUSPEND QMGR

Migrating a queue manager cluster: Migrating the test system

Migrate each queue manager in the test system.

About this task

For each queue manager in the test system, in the order defined in the migration plan you developed in "Migrating a queue manager cluster: Create a plan" on page 117, migrate, and test the queue manager.

Migrating a queue manager cluster: Migrating the production system

Migrate each queue manager in the production system.

About this task

For each queue manager in the production system, in the order defined in the migration plan you developed in "Migrating a queue manager cluster: Create a plan" on page 117, migrate, and test the queue manager.

Windows: Migrating an MSCS configuration

Migrate queue managers in MSCS configuration 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 must always upgrade an offline node with no online IBM WebSphere MQ resources. In an Active/Passive configuration, if the node is Passive, you must ensure it cannot be switched to Active during the upgrade process.

The example, "Migrating a four-node MSCS cluster from IBM WebSphere MQ Version 6.0 to IBM WebSphere MQ Version 7.5" on page 120, shows this procedure applied to a four-node cluster.

Procedure

1. Modify the possible owners of the IBM WebSphere MQ resource to encompass only the Active node or nodes. With no owners assigned to Passive nodes, the IBM WebSphere MQ resource that is being migrated cannot be activated.
2. Ensure that the group containing the IBM WebSphere MQ resource is currently on one of the nodes defined as a possible owner. The group must include any applications connecting to the queue manager resource.
3. Stop the cluster service on the node being migrated. The MSCS cache is cleared of any IBM WebSphere MQ DLLs that have been registered.
4. Migrate the selected node by following the standard instructions in "Windows: Migrating a queue manager to a later version" on page 99. Apply the required maintenance level.
5. Start the cluster service on the selected node.
6. On the next node to be migrated, ensure that the IBM WebSphere MQ resources are offline.
7. Remove this node from the list of possible owners. For clusters with more than two nodes, see the Additional considerations later in this topic.
8. Move the group containing the IBM WebSphere MQ resource to one of the possible owners and bring it online.
9. Repeat steps 3-8 as necessary for any remaining nodes.

Migrating a four-node MSCS cluster from IBM WebSphere MQ Version 6.0 to IBM WebSphere MQ Version 7.5

The example in Table 9 on page 122 illustrates the steps involved in migrating a four-node MSCS cluster.

In the example IBM WebSphere MQ resources include queue managers, applications, and dependant MSCS resources, such as an IP address defined an as MSCS resource. In each step, the changes are italicized.

Step 1

Select the node to migrate and prepare it for upgrading from IBM WebSphere MQ Version 6.0 to IBM WebSphere MQ Version 7.5.

1. Select node 1 to be migrated and convert it into a Passive node with no running IBM WebSphere MQ resources.
2. Modify the possible owners of the group containing the IBM WebSphere MQ resources, to encompass only the required online nodes. Failover does not attempt to switch IBM WebSphere MQ resources to the node that is not a possible owner. It is safe to migrate that node.
3. Move the group containing the IBM WebSphere MQ resource to one of the nodes that is a possible owner, and bring it online.
4. Stop the cluster service on the node being migrated. Stopping the service clears the MSCS cache of any IBM WebSphere MQ libraries that have been registered for MSCS. The node goes offline.

Step 2

Migrate IBM WebSphere MQ from Version 6.0 to Version 7.5

Step 3

Start the cluster service on the selected node. The node becomes online, but it is not a possible owner, so no work is switched to it.

Step 4

Repeat steps 1 - 3 for node 2. Nodes 1 and 2 are now online, and you have migrated them to IBM WebSphere MQ Version 7.5. They are still doing no work, as they are not possible owners of any of the IBM WebSphere MQ resource groups.

Step 5

Migrate the cluster from running IBM WebSphere MQ Version 6.0 to IBM WebSphere MQ Version 7.5. The number of migrated nodes is now greater or equal to the number of unmigrated nodes.

1. Change the set of possible owners from 3 , 4 to 1 , 2.
2. Move the IBM WebSphere MQ resource groups from nodes 3 and 4 to nodes 1 and 2 and bring online.
3. From this point onward, the list of possible owners must include migrated nodes only. The WebSphere MQ resource must never failover to a node running a back level version of the product.

Note: If you must revert IBM WebSphere MQ to an earlier version, the IBM WebSphere MQ resources must be removed from MSCS control, before performing an uninstallation of IBM WebSphere MQ

Step 6

Migrate node 3 to Version 7.5.

1. Follow steps 1 - 3 for node 3.
2. Add node 3 to the list of possible owners.
3. Move the QMC resource group back from node 1 to node 3 and bring online again.

Step 7

Repeat step 6 for node 4.

Table 9. Migrating a four-node MSCS cluster									
Steps		**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**
Node 1	State	Online	*Offline*	Offline	*Online*	Online	Online	Online	Online
	Version	Version 6.0	Version 6.0	*Version 7.5*	Version 7.5	Version 7.5	Version 7.5	Version 7.5	Version 7.5
	Groups	QMA					*QMC, QMA*	*QMA*	QMA
Node 2	State	Online	Online	Online	Online	Online	Online	Online	Online
	Version	Version 6.0	Version 6.0	Version 6.0	Version 6.0	*Version 7.5*	Version 7.5	Version 7.5	Version 7.5
	Groups	QMB	QMB	QMB	QMB		*QMD, QMB*	QMD, QMB	*QMB*
Node 3	State	Online	Online	Online	Online	Online	Online	Online	Online
	Version	Version 6.0	Version 6.0	Version 6.0	Version 6.0	Version 6.0	Version 6.0	*Version 7.5*	Version 7.5
	Groups	QMC	QMC, QMA	QMC, QMA	QMC, QMA	QMC, QMA		*QMC*	QMC
Node 4	State	Online	Online	Online	Online	Online	Online	Online	Online
	Version	Version 6.0	Version 6.0	Version 6.0	Version 6.0	Version 6.0	Version 6.0	Version 6.0	*Version 7.5*
	Groups	QMD	QMD	QMD	QMD	QMD, QMB			*QMD*
Possible Owners		1,2,3,4	*2,3,4*	2,3,4	2,3,4	*3,4*	*1,2*	*1,2,3*	*1,2,3,4*
Task			Update 1			Update 2	Transfer	Update 3	Update 4

What to do next

Additional considerations in an MSCS setup with more than 2 nodes: A cluster might contain enough nodes for you to form a group of migrated queue managers and a group of unmigrated nodes. Switch to the migrated group when it contains half the number of queue managers. Before you have reached the half way point, the unmigrated group are possible owners. When you reach the half way point, switch the possible owners to the migrated group.

Related concepts

“Migrate a queue manager in a high availability configuration” on page 44
Follow standard procedures to migrate a queue manager that is part of a high availability configuration on platforms other than z/OS.

Migrating from a single instance to a multi-instance queue manager

To migrate a single instance queue manager to a multi-instance queue manager, you must move the queue manager data to a shared directory, and reconfigure the queue manager on two other servers.

Before you begin

You must check the prerequisites for running a multi-instance queue manager as part of this task. Some environments have been tested with multi-instance queue managers, and are known to work. They are AIX, Red Hat Linux, SUSE Linux Enterprise Server, HP-UX with the file system on Linux Red Hat, IBM i, and Windows Server. See Testing and support statement for WebSphere MQ multi-instance queue managers

for the latest list of tested environments. The support statement has detailed version and prerequisite information for each environment it lists. Other environments might work; a test tool is provided with WebSphere MQ to assist you in qualifying other environments.

You must have three servers to run a multi-instance queue manager. One server has a shared file system to store the queue manager data and logs. The other servers run the active and standby instances of the queue manager.

About this task

You have a single-instance queue manager that you want to convert to a multi-instance queue manager. The queue manager conversion itself is straightforward, but you must do other tasks to create a fully automated production environment.

You must check the prerequisites for a multi-instance queue manager, set up the environment and check it. You must set up a monitoring and management system to detect if the multi-instance queue manager has failed and been automatically restarted. You can then find out what caused the restart, remedy it, and restart the standby. You must also modify applications, or the way applications are connected to the queue manager, so that they can resume processing after a queue manager restart.

Procedure

1. Check the operating system that you are going to run the queue manager on, and the file system on which the queue manager data and logs are stored on. Check that they can run a multi-instance queue manager.
   a) Consult Testing and support statement for WebSphere MQ multi-instance queue managers. See whether the combination of operating system and file system is tested and capable of running a multi-instance queue manager.

      A shared file system must provide lease-based locking to be adequate to run multi-instance queue managers. Lease-based locking is a recent feature of some shared file systems, and in some case fixes are required. The support statement provides you with the essential information.

   b) Run **amqmfsck** to verify that the file system is configured correctly.

      File systems are sometimes configured with performance at a premium over data integrity. It is important to check the file system configuration. A negative report from the **amqmfsck** tool tells you the settings are not adequate. A positive result is an indication that the file system is adequate, but the result is not a definitive statement that the file system is adequate. It is a good indication.

   c) Run the integrity checking application provided in the technote, Testing a shared file system for compatibility with WebSphere MQ Multi-instance Queue Managers.

      The checking application tests that the queue manager is restarting correctly.
2. Configure a user and group to be able to access a share on the networked file system from each server that is running a queue manager instance.

   On Windows, the security IDs (SIDs) of the `mqm` group can be different; see Windows domains and multi-instance queue managers .
   On UNIX and Linux, the `uid` and `gid` for `mqm` in `/etc/password` must be the same on each system; see Create a multi-instance queue manager on Linux.
3. Set up a directory for the share on the networked file system with the correct access permissions.

   A typical configuration is to set up a single shared directory that contains all data and log directories for all queue managers that use the shared disk; see Share named qmgrs and log directories (version 7.0.1 onwards) in Example directory configurations on UNIX systems.

   For example, create a root directory on the share called `MQHA` that has subdirectories `data` and `logs`. Each queue manager creates its own data and log directories under `data` and `logs`. Create `MQHA` with the following properties:

On Windows, create *drive*\MQHA on the shared drive. The owner is a member of mqm. mqm must have full-control authority. Create a share for *drive*\MQHA.

On UNIX, create /MQHA on the shared drive. /MQHA is owned by the user and group mqm and has the access permissions rwx.

If you are using an NFS v4 file server, add the line /MQHA * rw,sync,no_wdelay,fsid=0) to etc/exports, and then start the NFS daemon: /etc/init.d/nfs start.

4. Copy the queue manager data and the logs to the share.

   You can choose to copy files manually, by following the procedure to back up the queue manager. On Windows, you can run the **hamvmqm** command to move the queue manager data to the share. The **hamvmqm** command works for queue managers created before version 7.0.1, and not reconfigured with a datapath, or for queue managers that do not have a **DataPath** configuration attribute. Choose one of these methods:

   - Follow the instructions in Backing up queue manager data, copying the queue manager data to the share. You must use this method if the **DataPath** configuration attribute is specified for this queue manager.
   - Stop the queue manager, and then type the command,

     ```
     hamvmqm /m /dd share\data /dd share\logs
     ```

     Where *share* is to be the location of the data and logs that you created in step "3" on page 123.

5. Update the queue manager configuration information stored on the current queue manager server.

   If you moved the queue manager data and logs by running the **hamvmqm** command, the command has already modified the configuration information correctly for you.

   If you moved the queue manager data and logs manually, you must complete the following steps.

   - On Windows:

     a. Modify the log registry key:

     ```
     HKEY_LOCAL_MACHINE\SOFTWARE\IBM\MQSeries\CurrentVersion\Configuration\QueueManager\QMgrName\Log
     "LogPath"="share\\logs\\QMgrName\\"
     ```

     b. Modify the Prefix registry key:

     ```
     HKEY_LOCAL_MACHINE\SOFTWARE\IBM\MQSeries\CurrentVersion\Configuration\QueueManager\QMgrName
     "Prefix"="share\\data"
     ```

   - On UNIX, and Linux,

     a. Modify the Log: stanza in the queue manager qm.ini file, which is on the *share*:

     ```
     LogPath=share/logs/QMgrName
     ```

     b. Modify the QueueManager: stanza in the WebSphere MQ mqs.ini file, which is typically in the /var/mqm directory on UNIX and Linux:

     ```
     DataPath=share/data/QMgrName
     ```

   Where, *QMgrName* is the representation of the queue manager name in the existing registry key on Windows. *QMgrName* is the Directory name in the QueueManager: stanza in the mqs.ini file on UNIX, and Linux. *share* is share where the data and logs are moved to.

6. Add the queue manager configuration information to the new queue manager server.

   a) Run the **dspmqinf** command to display the queue manager information

      Run the command on the server that ran the queue manager in version 6.0.

   ```
   dspmqinf -o command QMgrName
   ```

The command output is formatted ready to create a queue manager configuration.

```
addmqinf -s QueueManager -v Name=QMgrName -v Directory=QMgrName -v
Prefix=d:\var\mqm Datapath=\share\data\QMgrName
```

b) Create a queue manager configuration on the other server.

Run the **addmqinf** command copied from the previous output

7. Add the network address of the new server to the connection name in client and channel definitions.

a) Find all the client, sender, and requester TCPIP settings that refer to the server.

Client settings might be in Client Definition Tables (CCDT), in environment variables, in Java properties files, or in client code.
Cluster channels automatically discover the connection name of a queue manager from its cluster receiver channel. As long as the cluster receiver channel name is blank or omitted, TCPIP discovers the IP address of the server hosting the queue manager.

b) Modify the connection name for each of these connections to include the TCPIP addresses of both servers that are hosting the multi-instance queue manager.

For example, change:

```
echo DISPLAY CHANNEL(ENGLAND) CONNAME | runmqsc QM1
```

```
5724-H72 (C) Copyright IBM Corp. 1994, 2025. ALL RIGHTS RESERVED.
Starting MQSC for queue manager QM1.
1: DISPLAY CHANNEL(ENGLAND) CONNAME
AMQ8414: Display Channel details.
CHANNEL(ENGLAND) CHLTYPE(SDR)
CONNAME(LONDON)
```

Into:

```
echo ALTER CHANNEL(ENGLAND) CHLTYPE(SDR) CONNAME('LONDON, BRISTOL') | runmqsc QM1
```

8. Update your monitoring and management procedures to detect the queue manager restarting.
9. Update client applications to be automatically reconnectable, if appropriate.
10. Update the start procedure for your WebSphere MQ applications to be started as queue manager services.
11. Start each instance of the queue manager, permitting them to be highly available.

The first instance of the queue manager that is started becomes the active instance.
Issue the command twice, once on each server.

```
strmqm -x QMgrName
```

What to do next

To get the highest availability out of multi-instance queue managers, you must design client applications to be reconnectable and server applications to be restartable; see Application recovery.

Related concepts

Application recovery
Automatic client reconnection
Channel and client reconnection
Multi-instance queue managers
Queue manager configuration files, qm.ini
Shared file system
The WebSphere MQ configuration file, mqs.ini
Windows domains and multi-instance queue managers
Working with services

Related tasks

Backing up queue manager data

Changing configuration information on Windows, UNIX and Linux systems

Create a multi-instance queue manager on Linux

Moving a queue manager to MSCS storage

Testing a shared file system for compatibility with WebSphere MQ Multi-instance Queue Managers

Testing and support statement for WebSphere MQ multi-instance queue managers

Verifying shared file system locking

Related reference

amqmfsck (file system check)

Reverting to a single-instance queue manager

Revert a multi-instance queue manager to a single instance queue manager by stopping the standby instance. Then restart the active instance and do not set the flag that permits standby instances.

Before you begin

You have at least three servers configured to run a queue manager as a multi-instance queue manager. The queue manager is currently running as a multi-instance queue manager, with one standby instance active.

About this task

The task involves deactivating the active standby so that only the running multi-instance queue manager remains active. To prevent a standby instance being started in the future, you must stop the active instance and restart it. When you restart it, you start it as a single instance queue manager that prevents standby instances being started. The standby instance is stopped as a separate step, to give you the option of restarting the active instance at a later date. You can stop both instances by running the standard `endmqm` *QMgrName* command on the server running the active queue manager.

Procedure

1. Stop the standby queue manager instance.

   On the server running the standby instance:

   - Windows, UNIX, and Linux

     ```
     endmqm -w QMgrName
     ```

2. Stop the active queue manager instance.

   On the server running the active instance:

   - Windows, UNIX, and Linux

     ```
     endmqm -w (QMgrName)
     ```

3. Restart the queue manager, preventing standbys.

   On the server that is going to run the queue manager:

   - Windows, UNIX, and Linux

     ```
     strmqm QMgrName
     ```

What to do next

You might want to run the queue manager as a single instance on the same server as the queue manager data.

When the queue manager is stopped move the queue manager data back to the server that is running the queue manager. Alternatively install WebSphere MQ, and then move the queue manager configuration definition onto the server with the queue manager data. Both tasks are variations of steps in "Migrating from a single instance to a multi-instance queue manager" on page 122 to create a multi-instance queue manager.

Converting a publish/subscribe hierarchy to a publish/subscribe cluster

Conversion to a cluster requires manual steps and the whole hierarchy needs to be converted at the same time.

Before you begin

1. Migrate all the queue managers to the latest version, migrating the publish/subscribe hierarchy.
2. Advise users that the publish/subscribe system is not available while conversion takes place. The conversion takes time. All the queue managers in the hierarchy have to be converted together. Publications that are being processed are not lost, but all running publish/subscribe applications that use the hierarchy must be stopped. There are manual configuration tasks to be performed, some of which are not automatic transformations, before applications can be restarted.
3. To minimize disruption, plan, prepare, and test scripts to do the manual steps. Also review which topic objects are to be clustered with the publish/subscribe application owners.

About this task

The task is organized to minimize disruption. Migrate all the queue managers involved before starting the task. Step 1 (Create a cluster and add all the queue managers in the hierarchy to the cluster) can be done in advance, without disrupting applications. Stop applications between step 2 (Stop publish/subscribe applications) and step 5 (Set the **CLUSTER** attribute on the topic objects that you have decided to make clustered) inclusive.

In step 5 (Set the **CLUSTER** attribute on the topic objects that you have decided to make clustered), you need to define cluster topic objects on the cluster topic host. There are administrative advantages of using cluster topic objects, but that is not the reason for defining cluster topic objects during the conversion task. The principal reason for defining cluster topic objects as part of the conversion task to restore existing publish/subscribe applications to a fully working state.

Existing publish/subscribe applications that use a hierarchy do not work in a cluster, unless you identify and define the correct cluster topics. In hierarchical publish/subscribe, subscriptions are propagated to every queue manager in the hierarchy, as long as the subscription is registered with **SUBSCOPE**(ALL). In clustered publish/subscribe no subscriptions are propagated to the cluster, unless the subscription resolves to a topic with the attribute **CLUSTER**(*clusterName*) set, and **SUBSCOPE**(ALL).

Part of the migration procedure is to identify or create topic objects that are to be given the cluster attribute. Application knowledge is required to choose the correct topics to associate with cluster topic objects. Here is some guidance to help you in choosing which topics to cluster.

- If your Version 6.0 publish/subscribe applications used streams, part of the migration process is to create topic objects corresponding to the streams. These topic objects are prime candidates to be made into cluster topic objects.

  The migration process creates one topic object for each stream that is defined on a Version 6.0 queue manager. Thus, depending on what streams are defined on what queue managers, you might find different topic objects created on different queue managers.

  Unlike any other clustered topic object, if you are doing queued publish/subscribe you must define the topic objects that correspond to streams on every queue manager in the cluster. Do *not* follow the typical procedure, and define the topic object only on the cluster topic host. If you are doing integrated publish/subscribe, you only need to define the cluster topic once. The queued publish/subscribe daemon requires a locally defined topic object corresponding to the name of the stream in SYSTEM.QPUBSUB.QUEUE.NAMELIST. You must set the **CLUSTER** attribute on each of the

automatically defined local topic objects. The cluster attribute causes subscriptions to topics that resolve to a stream topic object to be propagated to other members of the cluster. The attributes of the topic object are resolved to the cluster topic object that is locally defined, and not to the latest cluster topic object to be defined.

- Identify or create as few topic objects to be clustered as the application design requires. Allow other topics to inherit from these topic objects. To this end, look for topics near the root of the topic trees of your publish/subscribe applications, and make them clustered.
- Identify any Version 6.0 publish/subscribe applications that are going to be hard to migrate from a hierarchy to a cluster.

  Applications that use the default stream and have a flat topic space, or do not have topics you can clearly identify as the root of their topic trees, are hard to migrate. It might be difficult to know how to define cluster topic objects associated with the topics an application uses. Do *not*, except as a last resort, set the cluster attribute on `SYSTEM.BASE.TOPIC` as a way of causing all topics to inherit the cluster attribute. It might be worthwhile to convert your existing Version 6.0 publish/subscribe applications that use the default stream, to use named streams. Then each stream converts to a defined topic object that you can cluster.

  Note: If you set the cluster attribute on `SYSTEM.BASE.TOPIC`, you do not need to set it on any other topics that inherit from `SYSTEM.BASE.TOPIC`.

Procedure

1. Create a cluster and add all the queue managers in the hierarchy to the cluster.
   a) Create a cluster or nominate an existing cluster.

      The cluster does not need to be an existing publish/subscribe cluster.
      Use IBM WebSphere MQ script commands (MQSC), or any other type of administration command or utility that is available on your platform, such as the IBM WebSphere MQ Explorer.

   b) Check that each queue manager is in the cluster.

      Run the MQSC command:

      ```
      DISPLAY CLUSQMGR(*)
      ```

      If a queue manager is not in the cluster, add it.
2. Stop publish/subscribe applications.

   Allow current work to complete.

   a) Stop all publish/subscribe publishers.

      No more work must be put into the system.
      Do not stop new work by disabling the input stream queues - the input streams are needed to process publications that remain in the hierarchy. You must stop the applications themselves.

   b) Switch off queued publish/subscribe on all the queue managers in the hierarchy.

      Leave version 7.0.1 integrated publish/subscribe running.

      Run the following MQSC command on all the queue managers in the hierarchy.

      ```
      ALTER QMGR PSMODE(COMPAT)
      ```

   c) Wait.

      Transmission queues and channels used to connect queue managers in the hierarchy must complete processing publications that are already in the system.
      When no more publications are left in transmission queues and channels, all the publications have reached their destination queue manager. The queue manager delivers the publications to subscribers when queued publish/subscribe is re-enabled.
3. Delete all the channels and transmission queues used to connect queue managers in the hierarchy.

a) Stop all the channels used to connect the queue managers in the hierarchy.

   Run the following MQSC command on all the queue managers in the hierarchy.

```
STOP CHANNEL(SenderChanName) MODE(QUIESCE)
```

b) Delete all the channels used to connect the queue managers in the hierarchy.

   Run the following MQSC command on all the queue managers in the hierarchy.

```
DELETE CHANNEL(SenderChanName)
DELETE CHANNEL(ReceiverChanName)
```

c) Delete the transmission queues associated with the channels that were deleted.

   Run the following MQSC command on all the queue managers in the hierarchy.

```
DELETE QLOCAL(xmitQName)
```

4. Delete the queue manager hierarchy.

a) Enable queued publish/subscribe on each queue manager in the hierarchy using the MQSC command:

```
ALTER QMGR PSMODE(ENABLE)
```

b) Run the following MQSC command on each queue manager in the hierarchy, except the uppermost parent in the hierarchy which has no parent.

```
ALTER QMGR PARENT(' ')
```

   Alternatively, on IBM i run the following commands to remove queue managers from the hierarchy.

   i) Run **WRKMQMPS PUBSUBNAME**(*parentQmgrName*) to display the hierarchy.

   ii) Use **option 4=Remove** to remove the parent from the hierarchy.

   iii) Use **option 34=Work with Pub/Sub** to move down the subhierarchy

   iv) Repeat options **4** and **34** until there are no child queue managers displayed.

c) Before proceeding to the next step confirm that all the hierarchical relationships have been canceled.

   Run the following MQSC command on each queue manager.

```
DISPLAY PUBSUB TYPE(ALL)
```

5. Set the **CLUSTER** attribute on the topic objects that you have decided to make clustered.

a) If you need to create cluster topic objects, define them on the cluster topic host.

   For integrated publish/subscribe, define cluster topic objects on only one queue manager.

b) If you are setting the cluster attribute on existing topic objects, created by **strmqbrk**, set the cluster attribute on the topic object defined on the cluster topic host. Delete the topic object from the other queue managers.

   Multiple definitions of a cluster topic object on multiple queue managers in a cluster can lead to problems.

c) Review whether to delete any topic objects that were created by **strmqbrk** that are not clustered.

   Subscriptions that inherit from these topic objects do not get propagated to other queue managers in the cluster.

6. Restart publish/subscribe applications.

Related reference

DISPLAY CLUSQMGR

DISPLAY CLUSQMGR

WebSphere MQ cluster commands

WebSphere Event Broker and WebSphere Message Broker migration tasks

Migrate the publish/subscribe broker and collectives from WebSphere Event Broker or WebSphere Message Broker version 6.0 or 6.1 to Version 7.5. Migrate SCADA applications that use IBM WebSphere MQ Telemetry.

Before you begin

IBM WebSphere MQ does not support the content-based filtering provided in WebSphere Event Broker version 6.0. If you use this function and intend to continue to do so, you must migrate to WebSphere Message Broker version 7.0. Content-based filtering is supported in IBM WebSphere MQ Version 7.5, and used by WebSphere Message Broker; see Enabling content-based filtering with publish/subscribe.

About this task

Users can migrate applications from WebSphere Event Broker version 6.0 to use the publish/subscribe broker in IBM WebSphere MQ Version 7.5.

If you are upgrading from WebSphere Message Broker version 6.0 or 6.1, you must migrate any applications that use publish/subscribe. You must also migrate message flow applications that use `SCADAInput` or `SCADAOutput` nodes.

Procedure

If WebSphere Event Broker or WebSphere Message Broker are connected into a collective you must do the task “Migrating a WebSphere Message or Event Broker publish/subscribe collective to a WebSphere MQ publish/subscribe cluster” on page 133 before “Migrating publish/subscribe configuration data from WebSphere Event Broker or WebSphere Message Broker version 6.0 or 6.1” on page 131.

- Migrate publish/subscribe collectives to publish/subscribe clusters.

  “Migrating a WebSphere Message or Event Broker publish/subscribe collective to a WebSphere MQ publish/subscribe cluster” on page 133
- Migrate the publish/subscribe broker from WebSphere Event Broker or WebSphere Message Broker to IBM WebSphere MQ.

  “Migrating publish/subscribe configuration data from WebSphere Event Broker or WebSphere Message Broker version 6.0 or 6.1” on page 131
- Migrate SCADA applications to use IBM WebSphere MQ Telemetry.

  “Migration of telemetry applications from using WebSphere Message Broker version 6 to use IBM WebSphere MQ Telemetry and WebSphere Message Broker version 7.0” on page 135

What to do next

The WebSphere Event Broker or WebSphere Message Broker publish/subscribe information is not deleted by the migration process. This information is stored in the subscription database tables, and is available to use until you explicitly delete it.

Related concepts

“Migration of the publish/subscribe broker in WebSphere Event Broker and WebSphere Message Broker” on page 48

“Migration phases during the migration of WebSphere Message Broker or WebSphere Event Broker” on page 49

The migration of the publish/subscribe broker from WebSphere Message Broker or WebSphere Event Broker to WebSphere MQ has three phases. The rehearsal phase, the initial phase, and the completion phase.

Related tasks

Selecting on the content of a message

Migrating publish/subscribe configuration data from WebSphere Event Broker or WebSphere Message Broker version 6.0 or 6.1

Complete these tasks to migrate publish/subscribe configuration data from WebSphere Event Broker Version 6.0 to WebSphere MQ Version 7.0.1 and later versions.

Before you begin

Familiarize yourself with the information in "Migration of the publish/subscribe broker in WebSphere Event Broker and WebSphere Message Broker" on page 48.

Install WebSphere MQ Version 7.0.1 or a later version.

On distributed systems, set up and initialize a command environment for the WebSphere Event Broker from which you are migrating. You are going to run the WebSphere MQ commands and WebSphere Event Broker using this environment.

Set the queue manager **PSMODE** attribute to `COMPAT`, using the following command: `ALTER QMGR PSMODE(COMPAT)`. In `COMPAT` mode the queue manager processes publish/subscribe calls. It does not process any publish/subscribe command messages.

Procedure

1. Optional: Run the rehearsal phase of the migration.

   For example, on supported platforms other than z/OS, use the following command to rehearse the migration from a broker named BRK1:

   ```
   migmbbrk -r -b BRK1
   ```

   On z/OS, run the equivalent **CSQUMGMB** utility.
2. Review the contents of the log file and the security commands file to check what would happen in a real migration.
3. Run the initial phase of the migration.

   For example, on supported platforms other than z/OS, use the following command to run the initial phase from a broker named BRK1:

   ```
   migmbbrk -t -b BRK1
   ```

   On z/OS, run the equivalent **CSQUMGMB** utility.
4. Review the commands in the security commands file `amqmigrateacl.txt`. Ensure that they create a security environment that is equivalent to your broker security environment. The migration rehearsal might find an Access Control List (ACL) entry that denies access. If so, it reports it in the security command file and advises that the ACL migration must be performed manually. There are two alternative ways to migrate the access control information:

   - Modify the security commands, then go to step "5" on page 132

     Modify the generated commands so that they set up a security environment in the queue manager that is equivalent to the security environment that existed in the broker.
   - Revise the security definitions in the broker to generate security commands that require no modification.

     Modify the security environment in the broker to be similar to WebSphere MQ by following these steps:

a. Set the ACL entry for the root of the topic tree to **Deny** for both publish and subscribe.

   The **Deny** setting matches the setting for the root of the WebSphere MQ topic tree. **Deny** is the exact opposite of the broker default. It can result in many publishers and subscribers no longer having permission to perform operations that they have been doing successfully.

   The next two steps restore the successful operation of the denied publishers and subscribers:

   i) Remove all ACL entries that deny access, apart from the entry in step 4a.

      Many of these entries are likely to be redundant following denying access at the root of the topic tree. Other entries might require more extensive changes.

   ii) Add ACL entries, where required, to grant access and to restore a correct security environment.

b. Run the initial phase of the migration again and review the security command file.

   If your changes are correct, it creates a security environment in the queue manager that is equivalent to the security environment that existed in the broker.

5. Run the security commands to set up the security environment before you run the completion phase of the migration.

   If you do not run the security commands again, the migration fails.

6. Run the completion phase of the migration.

   - On supported platforms other than z/OS, run the following command to migrate the publish/subscribe configuration data from broker BRK1

     ```
     migmbbrk -c -o -b BRK1
     ```

     The command overwrites subscriptions or retained publications that exist in the queue manager and have the same name as a migrating subscription or retained publication.
   - On z/OS, use the equivalent **CSQUMGMB** utility.

   The completion phase migrates the publish/subscribe configuration data to the queue manager and shuts down the broker. It creates a log file and a security commands file.

   Note: It is possible that the broker state has changed since the initial phase was run and that additional topic objects are now required. If so, the completion phase creates these topic objects as necessary. The completion phase does not delete topic objects that have become unnecessary. You must delete topic objects that you do not require.

7. Add the name of every queue that is named in an WebSphere Event Broker or WebSphere Message Broker `MQInput` node to the namelist `SYSTEM.QPUBSUB.QUEUE.NAMELIST`.

   To edit a namelist, use either the WebSphere MQ Explorer or the following MQSC command:

   ```
   ALTER NAMELIST
   ```

   Note: Each queue name referenced in the `SYSTEM.QPUBSUB.QUEUE.NAMELIST` namelist also has an associated Topic object. You must define the Topic objects before adding the associated queue name to the `SYSTEM.QPUBSUB.QUEUE.NAMELIST` namelist. For more information about setting up the `SYSTEM.QPUBSUB.QUEUE.NAMELIST` namelist, see Mapping between streams and topics.

   WebSphere Event Broker and WebSphere Message Broker put published messages to, and read published messages from, queues using an `MQInput` node. By listing these queues in `SYSTEM.QPUBSUB.QUEUE.NAMELIST`, the queue manager monitors these queues for published messages.

   When the migration shuts down the broker, published messages are not read from the `MQInput` node in this way any more. Instead, the queue manager monitors the relevant queues.

8. Check the broker domain Event Log to confirm that the broker has been shut down.
9. Use the following command to set the **PSMODE** queue manager attribute to `ENABLED`.

```
ALTER QMGR PSMODE(ENABLED)
```

PSMODE(ENABLED) starts the queued publish/subscribe command message interface, and the integrated publish/subscribe call interface. The queue manager does all publish/subscribe processing.

What to do next

The migration process uses a queue called SYSTEM.TEMP.MIGMBBRK.REPLY.QUEUE to receive messages from the broker. When the process starts it checks for the existence of the queue and creates or clears it as necessary. When it has finished, it attempts to delete the queue. However, because the broker also uses the queue to send replies, te broker might still have the queue open. If so, the migration process is unable to delete the queue. When you have completed the migration, check if SYSTEM.TEMP.MIGMBBRK.REPLY.QUEUE exists, and if it does, delete it.

Related concepts

"Publish/Subscribe Broker: Access Control List (ACL) migration" on page 50
An automated process migrates the publish/subscribe broker from WebSphere Event Broker and WebSphere Message Broker to IBM WebSphere MQ. The process does not migrate the ACLs automatically. The migration task provides two methods to migrate the ACL to WebSphere MQ.

Related information

IBM Redbook: Connecting Your Business Using IBM WebSphere Message Broker V7 as an ESB

Migrating a WebSphere Message or Event Broker publish/subscribe collective to a WebSphere MQ publish/subscribe cluster

Complete this task to migrate a publish/subscribe collective. Migrate the collective from WebSphere Event Broker Version 6.0 or WebSphere Message Broker Version 6.0 or 6.1. Migrate it to a WebSphere MQ publish/subscribe cluster.

Before you begin

Install WebSphere MQ version 7.0.1 or a later version.

Make sure that you are familiar with the information about migration from WebSphere Message Broker and WebSphere Event Broker. See "Publish/Subscribe Broker: Access Control List (ACL) migration" on page 50.

Make sure that you are familiar with the first tasks to set up a new cluster and add a queue manager to it. See Configuring a queue manager cluster

On distributed systems, set up and initialize a command environment in which WebSphere MQ commands and WebSphere Event Broker commands can run.

Set the queue manager **PSMODE** attribute to COMPAT, using the following command: ALTER QMGR PSMODE(COMPAT)

Procedure

1. Stop all the publish/subscribe applications in the topology so that all in-flight messages are processed.
2. Use the **MQSC** command DISPLAY QUEUE(*) to check that the current depth on the following queues is zero:
   - SYSTEM.BROKER.INTERBROKER.QUEUE.1A
   - SYSTEM.BROKER.INTERBROKER.QUEUE.1N
   - SYSTEM.BROKER.INTERBROKER.QUEUE.1T
   - SYSTEM.BROKER.CONTROL.QUEUE
   - Transmit queues

- Message flow input queues

3. Upgrade the underlying queue manager of each broker in the collective to WebSphere MQ version 7.0.1 or a later version.
4. If any queue manager is currently connected using a manually defined channel and transmit queue, delete the manually defined channel now.

   During the removal of the transmit queues it is possible that messages from the WebSphere Message Broker configuration manager are left stranded on transmit queues. The configuration messages have not been processed off the queue, because all applications are stopped. The stranded messages are not a problem. If the configuration manager does not receive a timely response it automatically attempts to reestablish communication. When migration is complete, the configuration manager is no longer used.

   Note: If you roll back the publish/subscribe configuration migration, you must recreate these manually defined channels and transmit queues. If you do not, the cluster channels are used for communication between the brokers.
5. Use **MQSC** commands to set up a queue manager cluster containing all the queue managers that are associated with the brokers:
   a) Decide on a cluster name
   b) Nominate two queue managers as full repository queue managers:

   ```
   ALTER QMGR REPOS('clusname')
   ```

   c) Define the cluster receiver channel on each queue manager:

   ```
   DEFINE CHANNEL('to.qmgr_name') CHLTYPE(CLUSRCVR) TRPTYPE(TCP)
   CONNAME('(hostname(fr_listener_port)') CLUSTER('clusname')
   ```

   d) Start the channel:

   ```
   START CHANNEL('to.qmgr_name')
   ```

   e) Define the cluster to send to the full repository on each queue manager:

   ```
   DEFINE CHANNEL('to.fr_qmgr_name') CHLTYPE(CLUSSDR) TRPTYPE(TCP) CONNAME('fr
   hostname(fr_listener_port)') CLUSTER('clusname')
   ```

   f) Start the channel:

   ```
   START CHANNEL('to.fr_qmgr_name')
   ```
6. Create the publish/subscribe cluster. On each queue manager, alter the topic SYSTEM.BASE.TOPIC to add it to the cluster:

   ```
   ALTER TOPIC(SYSTEM.BASE.TOPIC) CLUSTER('clusname')
   ```
7. Wait for the cluster topics to propagate and then check that all queue managers are correctly participating in the cluster:

   ```
   DISPLAY CLUSQMGR(*)
   ```
8. For each queue manager, migrate the publish/subscribe configuration data to WebSphere MQ. See “Migrating publish/subscribe configuration data from WebSphere Event Broker or WebSphere Message Broker version 6.0 or 6.1” on page 131.
9. After migrating the publish/subscribe configuration data for each broker to each queue manager, resynchronize proxy subscriptions. The proxy subscriptions are resynchronized with all the other queue managers in the publish/subscribe cluster. Run this MQSC command to trigger the resynchronization:

   ```
   REFRESH QMGR TYPE (PROXYSUB)
   ```

Note: If you roll back and rerun the broker publish/subscribe migration, you must run this resynchronization step and all the steps that follow it.

10. Wait for all proxy subscriptions and retained publications to be propagated. Check that the status of subscriptions and publications is what you expect to see:
    a) Check the proxy subscriptions:

    ```
    DISPLAY SUB(*) SUBTYPE(PROXY)
    ```

    b) Check retained publications:

    ```
    DISPLAY TPSTATUS('#') RETAINED
    ```

    c) Use this MQSC command to check that the current depth of the following queues is zero:

    ```
    DISPLAY QLOCAL(*)

    SYSTEM.INTER.QMGR.CONTROL
    SYSTEM.INTER.QMGR.PUBS
    SYSTEM.INTER.QMGR.FANREQ
    SYSTEM.CLUSTER.TRANSMIT.QUEUE
    ```

11. Restart your publish/subscribe applications.

What to do next

The migration process uses a queue called SYSTEM.TEMP.MIGMBBRK.REPLY.QUEUE to receive messages from the broker. When the process starts it checks for the existence of the queue and creates or clears it as necessary. When it has finished, it attempts to delete the queue. However, because the broker also uses the queue to send replies, te broker might still have the queue open. If so, the migration process is unable to delete the queue. When you have completed the migration, check if SYSTEM.TEMP.MIGMBBRK.REPLY.QUEUE exists, and if it does, delete it.

Related concepts

“Publish/Subscribe Broker: Access Control List (ACL) migration” on page 50
An automated process migrates the publish/subscribe broker from WebSphere Event Broker and WebSphere Message Broker to IBM WebSphere MQ. The process does not migrate the ACLs automatically. The migration task provides two methods to migrate the ACL to WebSphere MQ.

Related tasks

“Migrating publish/subscribe configuration data from WebSphere Event Broker or WebSphere Message Broker version 6.0 or 6.1” on page 131
Complete these tasks to migrate publish/subscribe configuration data from WebSphere Event Broker Version 6.0 to WebSphere MQ Version 7.0.1 and later versions.

Configuring a queue manager cluster

Migration of telemetry applications from using WebSphere Message Broker version 6 to use IBM WebSphere MQ Telemetry and WebSphere Message Broker version 7.0

Migrating telemetry applications from using WebSphere Message Broker v6 to use IBM WebSphere MQ Telemetry and WebSphere Message Broker v7 requires a number of steps. Only the last step involves modifying the telemetry application. The first steps are migrating IBM WebSphere MQ and WebSphere Message Broker to v7.

You can run WebSphere Message Broker v6 with IBM WebSphere MQ Telemetry, if you convert all your SCADA flows to use IBM WebSphere MQ Telemetry. You must migrate IBM WebSphere MQ to Version 7.1 to use IBM WebSphere MQ Telemetry.

Before you begin

Your target environment is going to run the IBM WebSphere MQ Telemetry feature on IBM WebSphere MQ Version 7.1, with WebSphere Message Broker v7. IBM WebSphere MQ Telemetry runs on Windows, AIX, and Linux, see https://www.ibm.com/software/products/en/wmq-telemetry for detailed information.

Consider setting up parallel development and test environments before starting the task of upgrading and migrating all the components. In a large production environment, migrating the telemetry applications is the last step after upgrading IBM WebSphere MQ and WebSphere Message Broker to v7.

An alternative migration path is to migrate telemetry applications before migrating WebSphere Message Broker to v7. It is likely to be more effort to follow this path. New functionality in WebSphere Message Broker v7, namely republishing to the same topic, makes it easier to migrate some flows from SCADA to IBM WebSphere MQ Telemetry. By migrating the telemetry applications last you also avoid testing telemetry twice. First, you would test telemetry with WebSphere Message Broker v6, using the publish/subscribe engine in the broker. Second, you would test telemetry with WebSphere Message Broker v7, using IBM WebSphere MQ as the publish/subscribe engine.

About this task

You have created telemetry applications that use the MQTT protocol and the Supervisory Control And Data Acquisition (SCADA) nodes in WebSphere Message Broker v6. You want to upgrade your WebSphere Message Broker deployment from v6 to v7, but SCADA nodes are not supported in v7 of WebSphere Message Broker. Connectivity support for the MQTT protocol has been moved from WebSphere Message Broker to IBM WebSphere MQ. You want to leave as much of the telemetry application unchanged as you can. You must migrate the telemetry connectivity between the MQTT client and WebSphere Message Broker. Migrate it from the SCADA nodes in WebSphere Message Broker to the telemetry channels in IBM WebSphere MQ Version 7.1.

Procedure

1. Upgrade IBM WebSphere MQ

   See “Migration planning on UNIX, Linux, and Windows” on page 51.
2. Install the IBM WebSphere MQ Telemetry feature.

   See WebSphere MQ Telemetry installation.
3. Upgrade WebSphere Message Broker to v7.

   See Upgrade to WebSphere Message Broker v7.
4. Modify existing MQTT client applications to use v3 of the MQTT protocol.

   If your client applications are already using v3 of the MQTT protocol, no modifications to the applications are required.

   If you are using an MQTT client library with your applications, the library might not support MQTT v3. Migrate the library to use one of the C or Java MQTT v3 libraries provided with IBM WebSphere MQ Telemetry .
5. Modify your message flows from using SCADA nodes to using MQInput or JMSInput nodes.

Related concepts

WebSphere MQ Telemetry

Telemetry concepts and scenarios for monitoring and control

Related tasks

Installing WebSphere MQ Telemetry

Administering WebSphere MQ Telemetry

Developing applications for WebSphere MQ Telemetry

Troubleshooting for WebSphere MQ Telemetry

Related reference

WebSphere MQ Telemetry Reference

Changes that affect migration

Any changes in the behavior of IBM WebSphere MQ that affect existing applications or configurations are listed in the following topics. The changes are listed by migration path. For each migration path, all changes are listed. Study the changes to identify what migration tasks you must perform.

Changes are listed for currently supported migration paths. To migrate a queue manager from a release that is not listed, see “Migration paths” on page 6.

Changes in earlier releases

For information about what changed in an earlier release of the product, see the *What's changed* section in the product documentation for that release.

- For IBM WebSphere MQ Version 7.1, see What's changed in IBM WebSphere MQ Version 7.1.
- For older, out of support, versions of the product, the documentation is not available in the online IBM Documentation, but is available for you to download for offline use. For more information, see Documentation for older versions of WebSphere MQ.

Related concepts

“Migration paths” on page 6
Where can you find topics describing how to migrate WebSphere MQ from one release to another? Can you revert a migration to an earlier release? Can you migrate from an unsupported WebSphere MQ release to any later release directly?

Related tasks

“Migrating IBM WebSphere MQ to a later version on UNIX, Linux, and Windows ” on page 83

Coexistence

Queue managers, with different names, can coexist on any server as long as they use the same IBM WebSphere MQ installation. On UNIX, Linux, and Windows, different queue managers can coexist on the same server and be associated with different installations. In addition to queue managers coexisting on a server, objects, and commands must work correctly with different queue managers running at different command levels.

The coexistence section lists restrictions in the use of objects and commands when they are used with queue managers at multiple command levels. The queue managers might be running on a single server, or in a cluster.

Related concepts

“Queue manager coexistence in Version 7.5” on page 28
Queue managers, with different names, can coexist on any server as long as they use the same IBM WebSphere MQ installation. On UNIX, Linux, and Windows, different queue managers can coexist on the same server and be associated with different installations.

“Multi-installation queue manager coexistence on UNIX, Linux, and Windows” on page 31
You can install multiple copies of IBM WebSphere MQ for UNIX, Linux, and Windows on the same server. The installations must be at Version 7.1 or later, with one exception. One Version 7.0.1 installation, at fix pack level 6, or later, can coexist with multiple Version 7.1, or later installations.

Related tasks

“Migrating IBM WebSphere MQ library loading from Version 7.0.1 to Version 7.5” on page 108
No change in the way IBM WebSphere MQ libraries are loaded is normally required if you upgrade from Version 7.0.1 to Version 7.5. You must have followed the instructions on building IBM WebSphere MQ applications in Version 7.0.1 and you must replace IBM WebSphere MQ Version 7.0.1 with IBM WebSphere MQ Version 7.5. If you choose to take advantage of multi-installation in IBM WebSphere MQ Version 7.5, based on the side-by-side or multi-stage migration scenarios, you must modify the

environment for the operating system to resolve IBM WebSphere MQ dependencies for an application. Typically, you can modify the runtime environment, rather than relink the application.

JMS: Administered objects

Version 6.0 JMS administered objects, such as connection factories and destinations are compatible with later releases.

JMS applications can use connection factory or destination objects created in IBM WebSphere MQ Version 6.0. Any new properties, that did not exist in Version 6.0, assume their default value.

IBM WebSphere MQ Version 6.0 JMS applications can use connection factory or destination objects created in later releases. Any of the new properties that do not exist in Version 6.0 are ignored.

Cluster workload balancing in a mixed cluster

A cluster can contain queue managers running at IBM WebSphere MQ Version 7.5, and any currently supported level of IBM WebSphere MQ, but new features cannot be exploited from queue managers at an earlier level however.

As IBM WebSphere MQ Version 7.5 adds a new **DEFBIND** value called GROUP to queues, applications on queue managers earlier than Version 7.5 must not open or put messages to queues specifying the new value. When an application ignores this limitation, the workload balancing behaviour (for example: BIND_ON_OPEN or BIND_NOT_FIXED) is undefined.

Changes in IBM WebSphere MQ Version 7.5

All IBM WebSphere MQ changes that affect the migration of a queue manager from Version 7.1 to Version 7.5 are listed.

New capabilities that do not affect existing IBM WebSphere MQ applications are not listed. For more information about new features, see What's new in IBM WebSphere MQ Version 7.5.

Changes in Version 7.5

Table 10. Summary of Version 7.5 changes

What changed	Link to more information
New channel state: Switching	Display channel and cluster status: Switching
Command level	Command level changed to 750
endmqm command	Change of behavior of the endmqm command
V 7.5.0.8 V 7.5.0.8 JMS ExceptionListener (Version 7.5.0, Fix Pack 8)	JMS: Exception listener changes in Version 7.5

Stabilizations, deprecations and removals in Version 7.5

Table 11. Summary of stabilizations, deprecations and removals at Version 7.5

Stabilized, deprecated or removed feature	Link to more information
dspmqsver command (Advanced Message Security)	Removal of dspmqsver command

Display channel and cluster status: Switching

From IBM WebSphere MQ Version 7.5, a cluster-sender channel that is switching its configuration to a different cluster transmission queue has a new channel state: Switching.

When migrating from a release before Version 7.5 to Version 7.5 or later, existing application programs are not affected by the new state.

System management programs that monitor channel or cluster status might receive the new state as a result of a inquiry.

The state is set during the short interval while the channel modifies the destination transmission queue that messages are stored on. Before the switching state is set, messages are stored on the previously associated transmission queue. After the switching state, messages are stored on the newly configured transmission queue. The channel enters the switching state if a cluster-sender channel is starting, a configuration change is required, and the conditions for starting the switch are met.

Related concepts

Channel states

Cluster transmission queues and cluster-sender channels

Related reference

DISPLAY CHSTATUS

Inquire Channel Status (Response)

Inquire Cluster Queue Manager (Response)

MQCHS_* (Command format Channel Status)

Command level changed to 750

The command level on platforms other than z/OS and IBM i changes to 750 in Version 7.5. z/OS and IBM i are at command level 710.

Related reference

MQCMDL_* (Command Levels)

CommandLevel (MQLONG)

Change in behavior of the `endmqm` command

Issuing an **endmqm** command and **dspmq** command immediately after each other might return misleading status.

When issuing an **endmqm -c** or **endmqm -w** command, in the unlikely event that a dspmq command is issued in the small timeframe between the applications disconnecting and the queue manager actually stopping, the dspmq command might report the status as `Ending immediately`, even though a controlled shutdown is actually happening.

Related reference

endmqm

dspmq

Removal of `dspmqsver` command

Before IBM WebSphere MQ Version 7.5, the **dspmqsver** command was used to display the version of IBM WebSphere MQ Advanced Message Security.

From IBM WebSphere MQ Version 7.5, IBM WebSphere MQ Advanced Message Security is a component of the main product and the version information is displayed as part of the **dspmqver** command.

V 7.5.0.8 JMS: Exception listener changes in Version 7.5

In IBM WebSphere MQ Version 7.5, when using the IBM WebSphere MQ classes for JMS, changes are made to the way in which an application's JMS ExceptionListener is invoked.

Exception listener changes in Version 7.5

The following changes only apply to JMS applications that register MessageListener objects to consume messages asynchronously.

APAR IT14820, included from IBM WebSphere MQ Version 7.5.0, Fix Pack 8, fixed a defect where an application's JMS ExceptionListener would not be invoked for non-connection broken exceptions (for example MQRC_GET_INHIBITED) even though the ASYNC_EXCEPTIONS property on the JMS Connection Factory used by the application, was set to ASYNC_EXCEPTIONS_ALL. This was the default value prior to Version 7.5.0, Fix Pack 8.

From Version 7.5.0, Fix Pack 8, to maintain behavior for current JMS applications that configure a JMS MessageListener and a JMS ExceptionListener, and to ensure that the IBM WebSphere MQ classes for JMS are consistent with the JMS specification, the default value for the ASYNC_EXCEPTIONS JMS ConnectionFactory property has been changed to ASYNC_EXCEPTIONS_CONNECTIONBROKEN for the IBM WebSphere MQ classes for JMS. As a result, by default, only exceptions corresponding to broken connection error codes are delivered to an application's JMS ExceptionListener.

From Version 7.5.0, Fix Pack 8, the IBM WebSphere MQ classes for JMS have also been updated such that JMSExceptions relating to non-connection broken errors, which occur during message delivery to asynchronous message consumers, are still delivered to a registered ExceptionListener when the JMS ConnectionFactory used by the application has the ASYNC_EXCEPTIONS property set to the value ASYNC_EXCEPTIONS_ALL.

Exception listeners changes in Version 7.0

In later releases of the product, JMS exception listeners behave differently than they did in Version 6.0. Applications might receive more or fewer exceptions than they did in Version 6.0.

In Version 6.0 of IBM WebSphere MQ, an exception listener was called to inform the application of any error condition that occurred asynchronously to the application execution. During the processing of a message for an asynchronous consumer, the application might have had no other means to discover the exception. The errors that resulted in the exception listener being called included connection broken events, or if an attempt was made to process an unreadable message.

Previously, with Version 6.0 JMS clients, a property had to be explicitly set if connection broken type exceptions that occurred during synchronous API calls were to be passed to the application's ExceptionListener code. However, from Version 7.0, connection broken exceptions are expected to be passed, without the need to set any properties, to an application's ExceptionListener regardless of whether they were making synchronous API calls or receiving messages asynchronously to a MessageListener.

From Version 7.0, you can set the **ASYNC_EXCEPTIONS** property on JMS Connection Factory objects using the method:

```
setIntProperty(JmsConstants.ASYNC_EXCEPTIONS, int)
```

.

V 7.5.0.8 Before APAR IT14820, included from IBM WebSphere MQ Version 7.5.0, Fix Pack 8, the default value of this property was ASYNC_EXCEPTIONS_ALL. With APAR IT14820, the default value is changed to ASYNC_EXCEPTIONS_CONNECTIONBROKEN. For more information, see “JMS: Exception listener changes in Version 7.5” on page 140.

If you set the **ASYNC_EXCEPTIONS** property to ASYNC_EXCEPTIONS_ALL, the exception listener is called for all broken connection exceptions. The exception listener is also called for all exceptions that occur outside the scope of a synchronous JMS API call. Creating a JMS Connection

with an ExceptionListener from a JMS Connection Factory that has **ASYNC_EXCEPTIONS** set to ASYNC_EXCEPTIONS_ALL should provide broadly the same behavior as a Version 6.0 JMS client that had the `activateExceptionListener` Java system property set to the value true.

If you set the **ASYNC_EXCEPTIONS** property 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. They 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. It is no longer possible to use the connection to send or receive messages.

Applications must take appropriate action, such as attempting to remake the connection, when exception listener calls are made.

Related concepts

Exceptions in WebSphere MQ classes for JMS

Related reference

ASYNCEXCEPTION

IBM WebSphere MQ maintenance tasks

This section describes how to apply and remove maintenance to IBM WebSphere MQ.

Related information

Let's troubleshoot

Applying and removing maintenance level updates on UNIX, Linux, and Windows

When you apply and remove maintenance level updates to IBM WebSphere MQ, no migration is required. Maintenance level updates are applied either as a fix pack, or by manually applying an interim fix. The following topics describe how to apply and remove fix packs on UNIX, Linux, and Windows.

About this task

Follow a link in Table 12 on page 141.

Table 12. Applying and removing maintenance

Apply	Remove
AIX	AIX
HP-UX	HP-UX
Linux	Linux
Solaris	Solaris
Windows	Windows

AIX: Applying maintenance level updates

How to apply maintenance level updates to IBM WebSphere MQ for AIX using **installp**.

Before you begin

If you are running on a server with multiple IBM WebSphere MQ installations, that is, IBM WebSphere MQ Version 7.0.1, Fix Pack 6 (or later), you must identify the installation. Make sure that the commands you enter run against the correct installation; see setmqenv.

You can apply and remove maintenance from a IBM WebSphere MQ MQI client that is not installed on the same server as a queue manager. You do not have to stop any queue managers or logon as administrator. Because you do not have to stop any queue managers, do not do steps “1” on page 151 to 3 in the following maintenance procedure.

Important: pax and **rpmbuild** are not supplied as part of the product. You must obtain these from your Linux distribution supplier.

About this task

Stop applications using the installation and use the **installp** command, to install maintenance level updates to clients and servers. Alternatively, if the installation is in the default installation location, you can use the *System Management Interface Tool*, SMIT.

Procedure

1. Log in as a user in `group mqm`.
2. Stop all applications using the IBM WebSphere MQ installation.

   If you use the IBM WebSphere MQ Managed File Transfer (MFT) component, ensure that any MFT agents have finished all of the file transfers that they were engaged in. There should be no incomplete transfers associated with the agents, and their SYSTEM.FTE.STATE queues should contain no messages.
3. End all the activity of queue managers associated with the IBM WebSphere MQ installation.
   a) Run the **dspmq** command to list the state of all the queue managers on the system.

      Run either of the following commands from the installation that you are updating:

      ```
      dspmq -o installation -o status
      dspmq -a
      ```

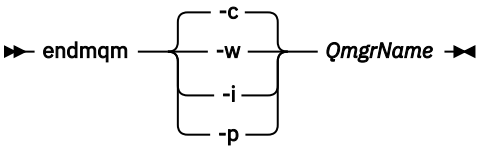
      dspmq -o installation -o status displays the installation name and status of queue managers associated with all installations of IBM WebSphere MQ.

      dspmq -a displays the status of active queue managers associated with the installation from which the command is run.

   b) Run the **MQSC** command, `DISPLAY LSSTATUS(*) STATUS` to list the status of listeners associated with a queue manager.

      ```
      echo "DISPLAY LSSTATUS(*) STATUS" | runmqsc QmgrName
      ```

   c) Run the **endmqm** command to stop each running queue manager associated with this installation.

      The **endmqm** command informs an application that the queue manager it is connected to is stopping; see Stopping a queue manager.

      For the maintenance to proceed, applications must respond to an **endmqm** command by disconnecting from the queue manager and releasing any IBM WebSphere MQ libraries they have loaded. If they do not, you must find another way to force applications to release IBM WebSphere MQ resources, such as by stopping the applications.
      You must also stop applications that are using the client libraries that are part of the installation. Client applications might be connected to a different queue manager, running a different installation of IBM WebSphere MQ. The application is not informed about queue managers in the current installation being shut down.

Any applications that continue to have IBM WebSphere MQ shared libraries from the installation loaded prevent you applying IBM WebSphere MQ maintenance. An application might disconnect from a queue manager, or be forcibly disconnected, but keep a IBM WebSphere MQ shared library loaded.

d) Stop any listeners associated with the queue managers, using the command:

```
endmqlsr -m QMgrName
```

4. Log in as root, or switch to the superuser using the **su** command.
5. Install the update in one of the following ways:
   - Update the whole installation in the default location:

     ```
     installp -agXYd . all
     ```

   - Update selected filesets in the default location:

     ```
     installp -agXYd . list of file sets
     ```

   - Update the whole product in a non-default location using the -R flag:

     ```
     installp -R USIL_Directory -agXYd . all
     ```

   - Update selected filesets in a non-default location using the -R flag:

     ```
     installp -R USIL_Directory -agXYd . list of file sets
     ```

   USIL_Directory is the installation parent directory. IBM WebSphere MQ is installed underneath the directory. For example, if `/USIL1` is specified, the IBM WebSphere MQ product files are located in `/USIL1/usr/mqm`. `/USIL1/usr/mqm` is known as the *MQ_INSTALLATION_PATH*.

Related tasks

Stopping a queue manager

Related reference

dspmq

Updating installed software on a WebSphere MQ for AIX Version 7.r system

If you want to migrate to an IBM WebSphere MQ, Version 7.`r` system (where `r` in this situation is 5) AIX from an earlier version of the WebSphere MQ system using SMIT, you must perform certain tasks.

If you are running on a server with multiple IBM WebSphere MQ installations, that is, IBM WebSphere MQ Version 7.0.1.6 (or later), you must identify the installation. Make sure that the commands you enter run against the correct installation; see setmqenv.

You can use the *System Management Interface Tool*, SMIT, only if the installation is in the default installation location.

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 system. 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 new version.

Migration from an earlier version of WebSphere MQ for AIX involves updating any currently installed file sets. It also involves installing any new file sets that might be required.

To update currently installed file sets:

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. Click **List** to display the Single Select List window.
4. Click **/dev/cd0 (CD Drive)**.
5. Click **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. Click **Enter**.
8. Click **OK** in the confirmation window to start updating the software.

When all previously installed file sets have been updated to the latest level, you can install any additional file sets.

AIX: Restoring the previous maintenance level

How to restore a previous maintenance level using the *System Management Interface Tool* (SMIT).

Before you begin

If you are running on a server with multiple IBM WebSphere MQ installations, that is, IBM WebSphere MQ Version 7.0.1, Fix Pack 6 (or later), you must identify the installation. Make sure that the commands you enter run against the correct installation; see setmqenv.

You can apply and remove maintenance from a IBM WebSphere MQ MQI client that is not installed on the same server as a queue manager. You do not have to stop any queue managers or logon as administrator. Because you do not have to stop any queue managers, do not do steps "1" on page 151 to 3 in the following maintenance procedure.

Important: pax and **rpmbuild** are not supplied as part of the product. You must obtain these from your Linux distribution supplier.

About this task

You can back out maintenance updates and restore your system to the previous maintenance/install level, for any component of IBM WebSphere MQ for AIX that is in the **APPLIED** state.

Use the following command to display the current state of the IBM WebSphere MQ for AIX filesets:

```
lslpp [ -R usil ] -l "mqm*"
```

To back out a maintenance update, as the user root, issue the command:

```
installp [ -R usil ] -r "mqm*"
```

Otherwise:

Procedure

1. Log in as a user in `group mqm`.
2. Stop all applications using the IBM WebSphere MQ installation.

   If you use the IBM WebSphere MQ Managed File Transfer (MFT) component, ensure that any MFT agents have finished all of the file transfers that they were engaged in. There should be no

incomplete transfers associated with the agents, and their SYSTEM.FTE.STATE queues should contain no messages.

3. End all the activity of queue managers associated with the IBM WebSphere MQ installation.
   a) Run the **dspmq** command to list the state of all the queue managers on the system.

      Run either of the following commands from the installation that you are updating:

      ```
      dspmq -o installation -o status
      dspmq -a
      ```

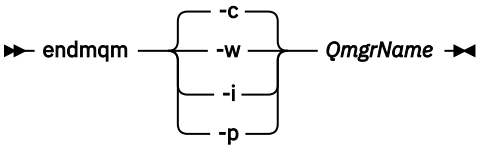
      dspmq -o installation -o status displays the installation name and status of queue managers associated with all installations of IBM WebSphere MQ.

      dspmq -a displays the status of active queue managers associated with the installation from which the command is run.

   b) Run the **MQSC** command, `DISPLAY LSSTATUS(*) STATUS` to list the status of listeners associated with a queue manager.

      ```
      echo "DISPLAY LSSTATUS(*) STATUS" | runmqsc QmgrName
      ```

   c) Run the **endmqm** command to stop each running queue manager associated with this installation.

      The **endmqm** command informs an application that the queue manager it is connected to is stopping; see Stopping a queue manager.

      For the maintenance to proceed, applications must respond to an **endmqm** command by disconnecting from the queue manager and releasing any IBM WebSphere MQ libraries they have loaded. If they do not, you must find another way to force applications to release IBM WebSphere MQ resources, such as by stopping the applications.

      You must also stop applications that are using the client libraries that are part of the installation. Client applications might be connected to a different queue manager, running a different installation of IBM WebSphere MQ. The application is not informed about queue managers in the current installation being shut down.

      Any applications that continue to have IBM WebSphere MQ shared libraries from the installation loaded prevent you applying IBM WebSphere MQ maintenance. An application might disconnect from a queue manager, or be forcibly disconnected, but keep a IBM WebSphere MQ shared library loaded.

   d) Stop any listeners associated with the queue managers, using the command:

      ```
      endmqlsr -m QMgrName
      ```

4. Log in as root, or switch to the superuser using the **su** command.
5. Open the appropriate **smit** panel using this sequence:

   ```
   Software Installation and Maintenance
     Software Maintenance and Utilities
       Reject Applied Software Updates (Use Previous Version)
   ```

   Alternatively, use a fast path command, `smit[ty] install_update`.
6. Complete the **SOFTWARE** name field.

   Enter `mqm*` to restore all applicable file set updates to your installation.

   Note: If an option to restore only selected file set updates for IBM WebSphere MQ for AIX appears, avoid it. The option results in all applicable file set updates for the maintenance update being restored.

7. Click **Enter** to reject the current maintenance level and reinstate the previous maintenance or install level.
   a) Accept displayed default values for all other fields
   b) Dismiss the confirmation message

   The reject process starts. While the command runs, it displays progress messages terminating with an **Install Summary** table.
   a) Check the table to see which components of IBM WebSphere MQ for AIX have been rejected

Related reference
dspmq
Related information
Stopping a queue manager

HP-UX: Applying maintenance level updates

How to apply maintenance level updates to IBM WebSphere MQ for HP-UX using **swinstall** .

Before you begin

1. Ensure you have enough disk space to apply maintenance level updates. A maintenance level update requires hard disk space for installation. In addition, the installation process might require a similar amount of disk space to save the previous level. For example, a 16 MB update might require 32 MB of space. The additional space allows a maintenance level update to be removed, and the previous level to be restored automatically.
2. If you are running on a server with multiple IBM WebSphere MQ installations, that is, IBM WebSphere MQ Version 7.0.1, Fix Pack 6 (or later), you must identify the installation. Make sure that the commands you enter run against the correct installation; see setmqenv.
3. You can apply and remove maintenance from a IBM WebSphere MQ MQI client that is not installed on the same server as a queue manager. You do not have to stop any queue managers or logon as administrator. Because you do not have to stop any queue managers, do not do steps “1” on page 151 to 3 in the following maintenance procedure.

About this task

1. If you want to install both the base package and the maintenance update packages, install the base package separately first. Then install the maintenance update packages.
2. If you are using the interactive installer, click **Options > Change Options**. Then clear the **autoselect dependencies when marking software** check box before selecting the maintenance update package for installation.
3. Error messages might be seen when running **swinstall** , even when successfully updating an installation.

   There are two approaches you can take to handling errors in the application of maintenance.

   a. Aim for an error-free update by applying maintenance only to those components that are installed.
   b. Apply the whole maintenance package and check the error logs, error by error, ignoring the insignificant errors.

   Both approaches are described.

   Many of the insignificant errors are caused by **swinstall** trying to apply updates to components that are not installed. Consider whether there are any significant errors reported with the insignificant ones.

- The following errors might not indicate a serious problem. They are written to the console, or to the **swinstall** panel.

```
ERROR:   "hpux11.mycompany.com:/":
The software dependencies for 15 products or filesets cannot be resolved.
```

```
ERROR:   "hpux11.mycompany.com:/":
17 filesets were determined to be skipped in the analysis phase.
The execution phase failed for "hpux11.mycompany.com:/".
Analysis and Execution had errors.
```

- The following errors might not indicate a serious problem. They are written to the swjob output for a **swinstall** session.

```
ERROR:   17 of 20 filesets had Errors.
3 of 20 filesets had no Errors or Warnings.
```

```
ERROR:   The Execution Phase had errors.
See the above output for details.
```

Procedure

1. Log in as a user in group mqm.
2. Stop all applications using the IBM WebSphere MQ installation.

   If you use the IBM WebSphere MQ Managed File Transfer (MFT) component, ensure that any MFT agents have finished all of the file transfers that they were engaged in. There should be no incomplete transfers associated with the agents, and their SYSTEM.FTE.STATE queues should contain no messages.
3. End all the activity of queue managers associated with the IBM WebSphere MQ installation.
   a) Run the **dspmq** command to list the state of all the queue managers on the system.

   Run either of the following commands from the installation that you are updating:

   ```
   dspmq -o installation -o status
   dspmq -a
   ```

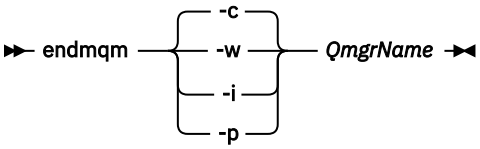
   dspmq -o installation -o status displays the installation name and status of queue managers associated with all installations of IBM WebSphere MQ.

   dspmq -a displays the status of active queue managers associated with the installation from which the command is run.

   b) Run the **MQSC** command, DISPLAY LSSTATUS(*) STATUS to list the status of listeners associated with a queue manager.

   ```
   echo "DISPLAY LSSTATUS(*) STATUS" | runmqsc QmgrName
   ```

   c) Run the **endmqm** command to stop each running queue manager associated with this installation.



   The **endmqm** command informs an application that the queue manager it is connected to is stopping; see Stopping a queue manager.

   For the maintenance to proceed, applications must respond to an **endmqm** command by disconnecting from the queue manager and releasing any IBM WebSphere MQ libraries they have loaded. If they do not, you must find another way to force applications to release IBM WebSphere MQ resources, such as by stopping the applications.

You must also stop applications that are using the client libraries that are part of the installation. Client applications might be connected to a different queue manager, running a different installation of IBM WebSphere MQ. The application is not informed about queue managers in the current installation being shut down.

Any applications that continue to have IBM WebSphere MQ shared libraries from the installation loaded prevent you applying IBM WebSphere MQ maintenance. An application might disconnect from a queue manager, or be forcibly disconnected, but keep a IBM WebSphere MQ shared library loaded.

d) Stop any listeners associated with the queue managers, using the command:

```
endmqlsr -m QMgrName
```

4. Log in as root, or switch to the superuser using the **su** command.
5. Make your current directory the location of the *Service_update_package*.

   The file name of the *Service_update_package* follows the pattern `hp-Uxxxx.v11`. You must prefix *Service_update_package* with the absolute path to the installation file. To save typing, construct the path using the `$PWD` variable.
6. Run the HP-UX command **swlist** `l=`*MQ_INSTALLATION_PATH* `MQSERIES` to list all of the IBM WebSphere MQ components that are installed.
7. Decide whether to install the updates interactively, and if you want to control which components are updated.

   You can update in the following ways:

   - Silently update all the installed IBM WebSphere MQ components by installing the whole maintenance package.

     ```
     swinstall -s $PWD/service_update_package
      MQSERIES,l=MQ_INSTALLATION_PATH
     ```

     The **swinstall** command attempts to find an installed component for every component in the update package, and updates it. **swinstall** writes out error messages for components that it cannot find.
   - Silently update some IBM WebSphere MQ components by installing only the required updates from the maintenance package.

     If you specify *update_components* correctly, the update procedure can be error-free. **swinstall** only updates components that you have listed and components that are dependent on components you have listed.

     a. Using the list of installed IBM WebSphere MQ components, create a space separated list of the components you want to update (*update_components*). This list requires the installation path of each component to be specified, in the form: *component*`,l=`*MQ_INSTALLATION_PATH*

     b.
     ```
     swinstall -s $PWD/service_update_package
      update_components
     ```
   - Interactively update some IBM WebSphere MQ components from the maintenance package, selecting only the update components that are required.

     ```
     swinstall -s $PWD/
     service_update_package
     ```

     a. Open **MQSERIES** and mark the update components you want to apply. Correctly marked, there are no errors when the updates are applied. The installer resolves dependencies automatically.

     b. Select **Actions > Change Product Location** to change the IBM WebSphere MQ installation you intend to update.

     c. Select **Actions > Install**. The log file tells you if there are any problems that need fixing.

Related reference
dspmq
Related information
Stopping a queue manager

HP-UX: Restoring the previous maintenance level

How to restore a previous maintenance level by using **swremove**.

Before you begin

If you are running on a server with multiple IBM WebSphere MQ installations, that is, IBM WebSphere MQ Version 7.0.1, Fix Pack 6 (or later), you must identify the installation. Make sure that the commands you enter run against the correct installation; see setmqenv.

You can apply and remove maintenance from a IBM WebSphere MQ MQI client that is not installed on the same server as a queue manager. You do not have to stop any queue managers or logon as administrator. Because you do not have to stop any queue managers, do not do steps "1" on page 151 to 3 in the following maintenance procedure.

Important: pax and **rpmbuild** are not supplied as part of the product. You must obtain these from your Linux distribution supplier.

Procedure

1. Log in as a user in `group mqm`.
2. Stop all applications using the IBM WebSphere MQ installation.

   If you use the IBM WebSphere MQ Managed File Transfer (MFT) component, ensure that any MFT agents have finished all of the file transfers that they were engaged in. There should be no incomplete transfers associated with the agents, and their SYSTEM.FTE.STATE queues should contain no messages.
3. End all the activity of queue managers associated with the IBM WebSphere MQ installation.
   a) Run the **dspmq** command to list the state of all the queue managers on the system.

      Run either of the following commands from the installation that you are updating:

      ```
      dspmq -o installation -o status
      dspmq -a
      ```

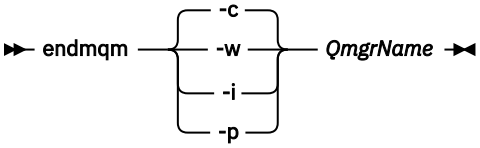
      dspmq -o installation -o status displays the installation name and status of queue managers associated with all installations of IBM WebSphere MQ.

      dspmq -a displays the status of active queue managers associated with the installation from which the command is run.

   b) Run the **MQSC** command, `DISPLAY LSSTATUS(*) STATUS` to list the status of listeners associated with a queue manager.

      ```
      echo "DISPLAY LSSTATUS(*) STATUS" | runmqsc QmgrName
      ```

   c) Run the **endmqm** command to stop each running queue manager associated with this installation.

      The **endmqm** command informs an application that the queue manager it is connected to is stopping; see Stopping a queue manager.

For the maintenance to proceed, applications must respond to an **endmqm** command by disconnecting from the queue manager and releasing any IBM WebSphere MQ libraries they have loaded. If they do not, you must find another way to force applications to release IBM WebSphere MQ resources, such as by stopping the applications.

You must also stop applications that are using the client libraries that are part of the installation. Client applications might be connected to a different queue manager, running a different installation of IBM WebSphere MQ. The application is not informed about queue managers in the current installation being shut down.

Any applications that continue to have IBM WebSphere MQ shared libraries from the installation loaded prevent you applying IBM WebSphere MQ maintenance. An application might disconnect from a queue manager, or be forcibly disconnected, but keep a IBM WebSphere MQ shared library loaded.

d) Stop any listeners associated with the queue managers, using the command:

```
endmqlsr -m QMgrName
```

4. Log in as root, or switch to the superuser using the **su** command.
5. Run the `swremove` command to remove the maintenance package from the system.

   For example, to remove the 7.R.0.1 maintenance level, use the command:

   ```
   swremove MQSERIES,r=7.R.0.1,l=MQ_INSTALLATION_PATH
   ```

   where:

   - R is the number of the Release
   - *MQ_INSTALLATION_PATH* is the installation path for IBM WebSphere MQ

   Details of the **swremove** command can be found in the *HP-UX Administration Guide* or by using the **man swremove** command.

Related reference

dspmq

Related information

Stopping a queue manager

Linux: Applying maintenance level updates

How to apply maintenance level updates to IBM WebSphere MQ for Linux using RPM.

Before you begin

If you are running on a server with multiple IBM WebSphere MQ installations, that is, IBM WebSphere MQ Version 7.0.1, Fix Pack 6 (or later), you must identify the installation. Make sure that the commands you enter run against the correct installation; see setmqenv.

You can apply and remove maintenance from a IBM WebSphere MQ MQI client that is not installed on the same server as a queue manager. You do not have to stop any queue managers or logon as administrator. Because you do not have to stop any queue managers, do not do steps “1” on page 151 to 3 in the following maintenance procedure.

Important: pax and **rpmbuild** are not supplied as part of the product. You must obtain these from your Linux distribution supplier.

About this task

Maintenance level updates are delivered in the form of Red Hat Package Manager (RPM) update images, which are applied using the RPM installation tool.

Additional disk space is required for the update images to allow maintenance level updates to be removed and the previous level restored. The updated files are kept in *MQ_INSTALLATION_PATH*/maintenance directory. Do not delete or move this directory or the files it contains.

MQ_INSTALLATION_PATH represents the high-level directory in which IBM WebSphere MQ is installed.

Updates are cumulative. You can apply your chosen update directly, without applying any previous updates first. The maintenance level updates might contain updates for one or more packages. You must apply those parts of an update that correspond to the packages that are applied in your installation.

Important: Although it is possible to install a fix pack at the same level as an installation performed from a manufacturing refresh image at that level, you should not attempt this process. Installing a fix pack at the same level as the one already on your system, can leave the package management database of your system in an inconsistent state with respect to the installation of IBM WebSphere MQ.

Procedure

1. Log in as a user in `group mqm`.
2. Stop all applications using the IBM WebSphere MQ installation.

   If you use the IBM WebSphere MQ Managed File Transfer (MFT) component, ensure that any MFT agents have finished all of the file transfers that they were engaged in. There should be no incomplete transfers associated with the agents, and their SYSTEM.FTE.STATE queues should contain no messages.
3. End all the activity of queue managers associated with the IBM WebSphere MQ installation.
   a) Run the **dspmq** command to list the state of all the queue managers on the system.

      Run either of the following commands from the installation that you are updating:

      ```
      dspmq -o installation -o status
      dspmq -a
      ```

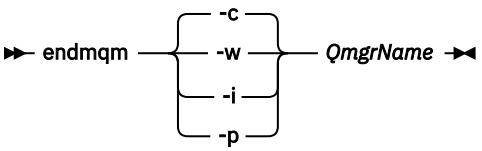
      dspmq -o installation -o status displays the installation name and status of queue managers associated with all installations of IBM WebSphere MQ.

      dspmq -a displays the status of active queue managers associated with the installation from which the command is run.

   b) Run the **MQSC** command, `DISPLAY LSSTATUS(*) STATUS` to list the status of listeners associated with a queue manager.

      ```
      echo "DISPLAY LSSTATUS(*) STATUS" | runmqsc QmgrName
      ```

   c) Run the **endmqm** command to stop each running queue manager associated with this installation.

      The **endmqm** command informs an application that the queue manager it is connected to is stopping; see Stopping a queue manager.

      For the maintenance to proceed, applications must respond to an **endmqm** command by disconnecting from the queue manager and releasing any IBM WebSphere MQ libraries they have loaded. If they do not, you must find another way to force applications to release IBM WebSphere MQ resources, such as by stopping the applications.

      You must also stop applications that are using the client libraries that are part of the installation. Client applications might be connected to a different queue manager, running a different installation of IBM WebSphere MQ. The application is not informed about queue managers in the current installation being shut down.

Any applications that continue to have IBM WebSphere MQ shared libraries from the installation loaded prevent you applying IBM WebSphere MQ maintenance. An application might disconnect from a queue manager, or be forcibly disconnected, but keep a IBM WebSphere MQ shared library loaded.

d) Stop any listeners associated with the queue managers, using the command:

```
endmqlsr -m QMgrName
```

4. Log in as root, or switch to the superuser using the **su** command.
5. Change into the directory containing the maintenance packages.
6. Run the `ls` command to list the available updates.

   For example, if there are level 1 maintenance updates for the Runtime, SDK and Server packages, you see the following:

```
MQSeriesRuntime-Uxxxx-7.R.0-1.i386.rpm
MQSeriesSDK-Uxxxx-7.R.0-1.i386.rpm
MQSeriesServer-Uxxxx-7.R.0-1.i386.rpm
```

   where R is the number of the Release.
7. Run the **rpm** command to find out which packages are installed on your server.

   Enter the following command:

```
rpm -qa | grep MQSeries
```

   For example, if you have a minimum IBM WebSphere MQ installation and SDK component, at level 0, the **rpm** command returns:

```
MQSeriesRuntime-7.R.0-0
MQSeriesSDK-7.R.0-0
MQSeriesServer-7.R.0-0
```

   where R is the number of the Release.
8. If this fix pack is to be upgraded on an installation, other than the first installation on the system, run the **crtmqfp** command to create and use a unique set of packages to install on the system.

   Note, that if this is the first, or only, IBM WebSphere MQ installation on the system, you can ignore this step.

   You must install the **pax** command in order for the **crtmqfp** command to run on Linux.

   a) Run the command `./crtmqfp` *<suffixname>* where *suffixname* is the same as the suffix used during renaming of the base level IBM WebSphere MQ installation.

   b) Set your current directory to the location specified when the **crtmqfp** command completes.

      This directory is a subdirectory of `/var/tmp/mq_rpms`, in which the unique set of packages is created. The packages have the suffix value contained within the filename.

      For example, if you used suffix 1 during repackaging of the base level IBM WebSphere MQ installation, enter the command: **crtmqfp** 1.

      There is now a subdirectory named `/var/tmp/mq_rpms/1/xxxx`, and the packages will be renamed, for example, from `MQSeriesRuntime-7.R.0-1.xxxx.rpm` to `MQSeriesRuntime_1-7.R.0-1.xxxx.rpm`. Where R is the number of the Release.
9. Run the **rpm** command to apply all available updates for the packages you have on your system:

   - To update an installation in the default location, `/opt/mqm`:

```
rpm -ivh MQSeriesRuntime-Uxxxx-7.R.0-1.i386.rpm
  MQSeriesSDK-Uxxxx-7.R.0-1.i386.rpm
  MQSeriesServer-Uxxxx-7.R.0-1.i386.rpm
```

   where R is the number of the Release.

- To update an installation in a custom location, specify the **rpm** `prefix` option:

```
rpm --prefix /opt/customLocation -ivh MQSeriesRuntime-Uxxxx-7.R.0-1.i386.rpm
  MQSeriesSDK-Uxxxx-7.R.0-1.i386.rpm
  MQSeriesServer-Uxxxx-7.R.0-1.i386.rpm
```

  where R is the number of the Release.

You must apply all packages in a maintenance update that correspond to those packages that are currently installed on your system.

10. Repeat step "7" on page 152 to list the packages that are now available.

    The Runtime, SDK, and Server packages are now at level 1:

```
MQSeriesRuntime-7.R.0-0
MQSeriesSDK-7.R.0-0
MQSeriesServer-7.R.0-0
MQSeriesRuntime-Uxxxx-7.R.0-1
MQSeriesSDK-Uxxxx-7.R.0-1
MQSeriesServer-Uxxxx-7.R.0-1
```

    where R is the number of the Release.

    Note:

    After the installation of IBM WebSphere MQ fix packs, if you run the `rpm-verify` or `rpm -V` command, it does not return the correct results. It produces spurious results relating to missing files in *MQ_INSTALLATION_PATH*/maintenance.

    This error message can be ignored because it is a known limitation in the IBM WebSphere MQ fix pack installation code. For further information about this error, see WebSphere MQ Fix Pack install errors - Linux reports errors

What to do next

For further information about using RPM to install software packages, see your Linux documentation.

Related reference

dspmq

Related information

Stopping a queue manager

Linux: Restoring the previous maintenance level

How to remove updates and restore the previous maintenance level using **RPM**.

Before you begin

If you are running on a server with multiple IBM WebSphere MQ installations, that is, IBM WebSphere MQ Version 7.0.1, Fix Pack 6 (or later), you must identify the installation. Make sure that the commands you enter run against the correct installation; see setmqenv.

You can apply and remove maintenance from a IBM WebSphere MQ MQI client that is not installed on the same server as a queue manager. You do not have to stop any queue managers or logon as administrator. Because you do not have to stop any queue managers, do not do steps "1" on page 151 to 3 in the following maintenance procedure.

Important: pax and **rpmbuild** are not supplied as part of the product. You must obtain these from your Linux distribution supplier.

About this task

When maintenance is applied, the original versions of replaced files are saved to allow the updates to be removed if necessary. To restore the previous maintenance level, run an Red Hat Package Manager, RPM, uninstall command for all the packages that were updated by the maintenance package as follows:

Procedure

1. Log in as a user in `group mqm`.
2. Stop all applications using the IBM WebSphere MQ installation.

   If you use the IBM WebSphere MQ Managed File Transfer (MFT) component, ensure that any MFT agents have finished all of the file transfers that they were engaged in. There should be no incomplete transfers associated with the agents, and their SYSTEM.FTE.STATE queues should contain no messages.
3. End all the activity of queue managers associated with the IBM WebSphere MQ installation.
   a) Run the **dspmq** command to list the state of all the queue managers on the system.

      Run either of the following commands from the installation that you are updating:

      ```
      dspmq -o installation -o status
      dspmq -a
      ```

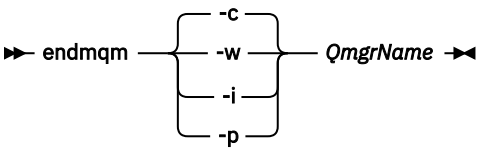
      dspmq -o installation -o status displays the installation name and status of queue managers associated with all installations of IBM WebSphere MQ.

      dspmq -a displays the status of active queue managers associated with the installation from which the command is run.

   b) Run the **MQSC** command, `DISPLAY LSSTATUS(*) STATUS` to list the status of listeners associated with a queue manager.

      ```
      echo "DISPLAY LSSTATUS(*) STATUS" | runmqsc QmgrName
      ```

   c) Run the **endmqm** command to stop each running queue manager associated with this installation.

      The **endmqm** command informs an application that the queue manager it is connected to is stopping; see Stopping a queue manager.

      For the maintenance to proceed, applications must respond to an **endmqm** command by disconnecting from the queue manager and releasing any IBM WebSphere MQ libraries they have loaded. If they do not, you must find another way to force applications to release IBM WebSphere MQ resources, such as by stopping the applications.

      You must also stop applications that are using the client libraries that are part of the installation. Client applications might be connected to a different queue manager, running a different installation of IBM WebSphere MQ. The application is not informed about queue managers in the current installation being shut down.

      Any applications that continue to have IBM WebSphere MQ shared libraries from the installation loaded prevent you applying IBM WebSphere MQ maintenance. An application might disconnect from a queue manager, or be forcibly disconnected, but keep a IBM WebSphere MQ shared library loaded.

   d) Stop any listeners associated with the queue managers, using the command:

      ```
      endmqlsr -m QMgrName
      ```

4. Log in as root, or switch to the superuser using the **su** command.
5. Run the **rpm** command to find out which packages are installed on your server.

   Enter the following command:

   ```
   rpm -qa | grep MQSeries
   ```

Using the example given in “Linux: Applying maintenance level updates” on page 150, returns:

```
MQSeriesRuntime-7.R.0-0
MQSeriesSDK-7.R.0-0
MQSeriesServer-7.R.0-0
MQSeriesRuntime-Uxxxx-7.R.0-1
MQSeriesSDK-Uxxxx-7.R.0-1
MQSeriesServer-Uxxxx-7.R.0-1
```

where R is the number of the Release.

6. Run the **rpm** command to remove all the updates applied at level 1.

   Enter the following commands:

```
rpm -ev MQSeriesRuntime-Uxxxx-7.R.0-1 MQSeriesSDK-Uxxxx-7.R.0-1
  MQSeriesServer-Uxxxx-7.R.0-1
```

   where R is the number of the Release.

7. Repeat step 5 to check that the ptf packages have been removed, leaving only the original install packages:

```
MQSeriesRuntime-7.R.0-0
MQSeriesSDK-7.R.0-0
MQSeriesServer-7.R.0-0
```

   where R is the number of the Release.

What to do next

For further information on using RPM to install software packages, see your Linux documentation.

Related reference

dspmq

Related information

Stopping a queue manager

Solaris: Applying maintenance level updates

How to apply maintenance level updates to IBM WebSphere MQ for Solaris using **pkgadd**.

Before you begin

1. Ensure you have enough disk space to apply maintenance level updates. A maintenance level update requires hard disk space for installation. In addition, the installation process might require a similar amount of disk space to save the previous level. For example, a 16 MB update might require 32 MB of space. The additional space allows a maintenance level update to be removed, and the previous level to be restored automatically.
2. If you are running on a server with multiple IBM WebSphere MQ installations, that is, IBM WebSphere MQ Version 7.0.1, Fix Pack 6 (or later), you must identify the installation. Make sure that the commands you enter run against the correct installation; see setmqenv.
3. You can apply and remove maintenance from a IBM WebSphere MQ MQI client that is not installed on the same server as a queue manager. You do not have to stop any queue managers or logon as administrator. Because you do not have to stop any queue managers, do not do steps “1” on page 151 to 3 in the following maintenance procedure.

About this task

Stop applications using the installation and use **pkgadd** to install maintenance.

Important: Although it is possible to install a fix pack at the same level as an installation performed from a manufacturing refresh image at that level, you should not attempt this process. Installing a fix pack at

the same level as the one already on your system, can leave the package management database of your system in an inconsistent state with respect to the installation of IBM WebSphere MQ.

Procedure

1. Log in as a user in `group mqm`.
2. Stop all applications using the IBM WebSphere MQ installation.

   If you use the IBM WebSphere MQ Managed File Transfer (MFT) component, ensure that any MFT agents have finished all of the file transfers that they were engaged in. There should be no incomplete transfers associated with the agents, and their SYSTEM.FTE.STATE queues should contain no messages.
3. End all the activity of queue managers associated with the IBM WebSphere MQ installation.
   a) Run the **dspmq** command to list the state of all the queue managers on the system.

      Run either of the following commands from the installation that you are updating:

      ```
      dspmq -o installation -o status
      dspmq -a
      ```

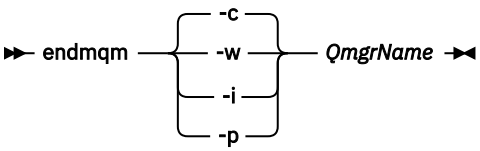
      dspmq -o installation -o status displays the installation name and status of queue managers associated with all installations of IBM WebSphere MQ.

      dspmq -a displays the status of active queue managers associated with the installation from which the command is run.

   b) Run the **MQSC** command, `DISPLAY LSSTATUS(*) STATUS` to list the status of listeners associated with a queue manager.

      ```
      echo "DISPLAY LSSTATUS(*) STATUS" | runmqsc QmgrName
      ```

   c) Run the **endmqm** command to stop each running queue manager associated with this installation.

      The **endmqm** command informs an application that the queue manager it is connected to is stopping; see Stopping a queue manager.

      For the maintenance to proceed, applications must respond to an **endmqm** command by disconnecting from the queue manager and releasing any IBM WebSphere MQ libraries they have loaded. If they do not, you must find another way to force applications to release IBM WebSphere MQ resources, such as by stopping the applications.

      You must also stop applications that are using the client libraries that are part of the installation. Client applications might be connected to a different queue manager, running a different installation of IBM WebSphere MQ. The application is not informed about queue managers in the current installation being shut down.

      Any applications that continue to have IBM WebSphere MQ shared libraries from the installation loaded prevent you applying IBM WebSphere MQ maintenance. An application might disconnect from a queue manager, or be forcibly disconnected, but keep a IBM WebSphere MQ shared library loaded.

   d) Stop any listeners associated with the queue managers, using the command:

      ```
      endmqlsr -m QMgrName
      ```
4. Log in as root, or switch to the superuser using the **su** command.
5. Change into the directory containing the maintenance packages.

6. Run the **crtmqfp** command to create and use a unique set of packages to install on the system, if this fix pack is to be upgraded on a installation that is not the first installation on the system.

   This command creates and uses a unique set of packages to install on the system.

   a) Run the command **crtmqfp** mqm-*<suffixname>* where *suffixname* is the same as the suffix used during renaming of the base level IBM WebSphere MQ installation.

      Note that this command creates a full copy of the installation packages in a subdirectory of `/var/tmp`.

   b) Set your current directory to the location specified when the **crtmqfp** command completes.

      This directory is a subdirectory of `/var/spool`, in which the unique set of packages is created. The packages have the suffix value contained within the filename.

7. Enter the following command:

   ```
   pkgadd -d packagename
   ```

   where `packagename` corresponds to the image file name. For example:

   ```
   mqm-U1234.img
   ```

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

8. Follow the on-screen instructions.

Related reference

dspmq

Related information

Stopping a queue manager

Solaris: Applying maintenance level updates in non-interactive mode

You can install IBM WebSphere MQ for Solaris non-interactively by creating a response file and an admin file.

Before you begin

1. Ensure you have enough disk space to apply maintenance level updates. A maintenance level update requires hard disk space for installation. In addition, the installation process might require a similar amount of disk space to save the previous level. For example, a 16 MB update might require 32 MB of space. The additional space allows a maintenance level update to be removed, and the previous level to be restored automatically.
2. If you are running on a server with multiple IBM WebSphere MQ installations, that is, IBM WebSphere MQ Version 7.0.1, Fix Pack 6 (or later), you must identify the installation. Make sure that the commands you enter run against the correct installation; see setmqenv.
3. You can apply and remove maintenance from a IBM WebSphere MQ MQI client that is not installed on the same server as a queue manager. You do not have to stop any queue managers or logon as administrator. Because you do not have to stop any queue managers, do not do steps “1” on page 151 to 3 in the following maintenance procedure.

About this task

Stop applications using the installation and use **pkgadd** to install maintenance.

Important: Although it is possible to install a fix pack at the same level as an installation performed from a manufacturing refresh image at that level, you should not attempt this process. Installing a fix pack at the same level as the one already on your system, can leave the package management database of your system in an inconsistent state with respect to the installation of IBM WebSphere MQ.

Procedure

1. Log in as a user in `group mqm`.
2. Stop all applications using the IBM WebSphere MQ installation.

   If you use the IBM WebSphere MQ Managed File Transfer (MFT) component, ensure that any MFT agents have finished all of the file transfers that they were engaged in. There should be no incomplete transfers associated with the agents, and their SYSTEM.FTE.STATE queues should contain no messages.
3. End all the activity of queue managers associated with the IBM WebSphere MQ installation.
   a) Run the **dspmq** command to list the state of all the queue managers on the system.

      Run either of the following commands from the installation that you are updating:

      ```
      dspmq -o installation -o status
      dspmq -a
      ```

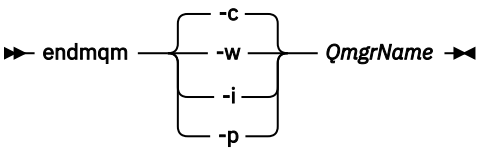
      dspmq -o installation -o status displays the installation name and status of queue managers associated with all installations of IBM WebSphere MQ.

      dspmq -a displays the status of active queue managers associated with the installation from which the command is run.

   b) Run the **MQSC** command, `DISPLAY LSSTATUS(*) STATUS` to list the status of listeners associated with a queue manager.

      ```
      echo "DISPLAY LSSTATUS(*) STATUS" | runmqsc QmgrName
      ```

   c) Run the **endmqm** command to stop each running queue manager associated with this installation.

      The **endmqm** command informs an application that the queue manager it is connected to is stopping; see Stopping a queue manager.

      For the maintenance to proceed, applications must respond to an **endmqm** command by disconnecting from the queue manager and releasing any IBM WebSphere MQ libraries they have loaded. If they do not, you must find another way to force applications to release IBM WebSphere MQ resources, such as by stopping the applications.

      You must also stop applications that are using the client libraries that are part of the installation. Client applications might be connected to a different queue manager, running a different installation of IBM WebSphere MQ. The application is not informed about queue managers in the current installation being shut down.

      Any applications that continue to have IBM WebSphere MQ shared libraries from the installation loaded prevent you applying IBM WebSphere MQ maintenance. An application might disconnect from a queue manager, or be forcibly disconnected, but keep a IBM WebSphere MQ shared library loaded.

   d) Stop any listeners associated with the queue managers, using the command:

      ```
      endmqlsr -m QMgrName
      ```
4. Log in as root, or switch to the superuser using the **su** command.
5. Change into the directory containing the maintenance packages.
6. Create the non-interactive install response file using the **pkgask** command:

   ```
   pkgask -d <location_to_image>/packagename -r response.txt mqm-07-05-00-02
   ```

where packagename corresponds to the image file name, for example mqm-U200403.img, response.txt is the name of the response file to create, and mqm-07-05-00-02 is the fix pack package name.

7. Find the admin_file from the server installation media located at <install_media>/silent/admin or create an admin_file in the following format:

```
mail=
instance=unique
partial=ask
runlevel=ask
idepend=ask
rdepend=ask
space=ask
setuid=nocheck
conflict=nocheck
action=nocheck
basedir=default
```

8. Run the **pkgadd** command to apply the maintenance level update IBM WebSphere MQ for Solaris in non-interactive mode:

```
 pkgadd -v -n -r response.txt -a admin_file -d <location_to_image>/packagename
mqm-07-05-00-02
```

   where admin_file is a path qualified name of the admin file you created, and mqm-07-05-00-02 corresponds to the fix pack package being installed.

9. Follow the on-screen instructions.

Related reference

dspmq

Related information

Stopping a queue manager

Solaris: Restoring the previous maintenance level

How to restore a previous maintenance level by stopping WebSphere MQ and using **pkgrm**.

Before you begin

If you are running on a server with multiple IBM WebSphere MQ installations, that is, IBM WebSphere MQ Version 7.0.1, Fix Pack 6 (or later), you must identify the installation. Make sure that the commands you enter run against the correct installation; see setmqenv.

You can apply and remove maintenance from a IBM WebSphere MQ MQI client that is not installed on the same server as a queue manager. You do not have to stop any queue managers or logon as administrator. Because you do not have to stop any queue managers, do not do steps “1” on page 151 to 3 in the following maintenance procedure.

Important: pax and **rpmbuild** are not supplied as part of the product. You must obtain these from your Linux distribution supplier.

About this task

When maintenance is applied, the original versions of replaced files are saved to allow the updates to be removed if necessary. To restore the previous maintenance level, run **pkgrm** command for all the packages that were updated by the maintenance package as follows:

Procedure

1. Log in as a user in group mqm.
2. Stop all applications using the IBM WebSphere MQ installation.

   If you use the IBM WebSphere MQ Managed File Transfer (MFT) component, ensure that any MFT agents have finished all of the file transfers that they were engaged in. There should be no

incomplete transfers associated with the agents, and their SYSTEM.FTE.STATE queues should contain no messages.

3. End all the activity of queue managers associated with the IBM WebSphere MQ installation.
   a) Run the **dspmq** command to list the state of all the queue managers on the system.

      Run either of the following commands from the installation that you are updating:

      ```
      dspmq -o installation -o status
      dspmq -a
      ```

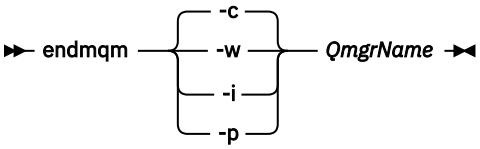
      dspmq -o installation -o status displays the installation name and status of queue managers associated with all installations of IBM WebSphere MQ.

      dspmq -a displays the status of active queue managers associated with the installation from which the command is run.

   b) Run the **MQSC** command, `DISPLAY LSSTATUS(*) STATUS` to list the status of listeners associated with a queue manager.

      ```
      echo "DISPLAY LSSTATUS(*) STATUS" | runmqsc QmgrName
      ```

   c) Run the **endmqm** command to stop each running queue manager associated with this installation.

      The **endmqm** command informs an application that the queue manager it is connected to is stopping; see Stopping a queue manager.

      For the maintenance to proceed, applications must respond to an **endmqm** command by disconnecting from the queue manager and releasing any IBM WebSphere MQ libraries they have loaded. If they do not, you must find another way to force applications to release IBM WebSphere MQ resources, such as by stopping the applications.

      You must also stop applications that are using the client libraries that are part of the installation. Client applications might be connected to a different queue manager, running a different installation of IBM WebSphere MQ. The application is not informed about queue managers in the current installation being shut down.

      Any applications that continue to have IBM WebSphere MQ shared libraries from the installation loaded prevent you applying IBM WebSphere MQ maintenance. An application might disconnect from a queue manager, or be forcibly disconnected, but keep a IBM WebSphere MQ shared library loaded.

   d) Stop any listeners associated with the queue managers, using the command:

      ```
      endmqlsr -m QMgrName
      ```

4. Log in as root, or switch to the superuser using the **su** command.
5. Run the **pkgrm** command to remove the latest maintenance update from the system:

   ```
   pkgrm packagename
   ```

   packagename is the name of the package that you want to remove; for example, `mqm-07-R-00-01.img`, where R is the number of the Release.

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

   If you do not know the name of the package to remove, try listing the packages that are installed using the following command: `pkginfo | grep mqm`

Note: Ignore any error messages of the form `<shared pathname not removed>`.

What to do next

If you have installed a IBM WebSphere MQ MQI client, and the client was updated after installing the maintenance level that is being removed, you must specifically update your IBM WebSphere MQ MQI client installation again, after the maintenance level has been removed

Related reference

dspmq

Related information

Stopping a queue manager

Windows: Applying maintenance level updates

How to apply maintenance level updates to WebSphere MQ for Windows.

Before you begin

1. If you are running on a server with multiple IBM WebSphere MQ installations, that is, IBM WebSphere MQ Version 7.0.1, Fix Pack 6 (or later), you must identify the installation. Make sure that the commands you enter run against the correct installation; see setmqenv.
2. You can apply and remove maintenance from a IBM WebSphere MQ MQI client that is not installed on the same server as a queue manager. You do not have to stop any queue managers or logon as administrator. Because you do not have to stop any queue managers, do steps “1” on page 151, “2” on page 151, “5” on page 152, and “6” on page 152 in the following maintenance procedure.
3. Download the maintenance package from the IBM WebSphere MQ Support website.
4. If User Account Control (UAC) is enabled, the user who does the installation must have Administrative authority. You must elevate any command or command prompt by selecting **Run as Administrator**. If you do not, the error AMQ4353 is written in the installation log

Procedure

1. Log on as an Administrator.
2. Stop all applications using the IBM WebSphere MQ installation.

   If you use the IBM WebSphere MQ Managed File Transfer (MFT) component, ensure that any MFT agents have finished all of the file transfers that they were engaged in. There should be no incomplete transfers associated with the agents, and their SYSTEM.FTE.STATE queues should contain no messages.
3. End all the activity of queue managers associated with the IBM WebSphere MQ installation.
   a) Run the **dspmq** command to list the state of all the queue managers on the system.

      Run either of the following commands from the installation that you are updating:

      ```
      dspmq -o installation -o status
      dspmq -a
      ```

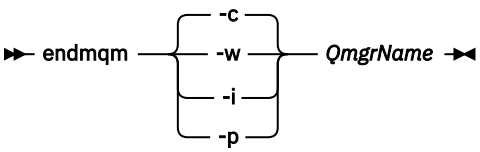
      dspmq -o installation -o status displays the installation name and status of queue managers associated with all installations of IBM WebSphere MQ.

      dspmq -a displays the status of active queue managers associated with the installation from which the command is run.

   b) Run the **MQSC** command, `DISPLAY LSSTATUS(*) STATUS` to list the status of listeners associated with a queue manager.

      ```
      echo DISPLAY LSSTATUS(*) STATUS | runmqsc QmgrName
      ```

   c) Run the **endmqm** command to stop each running queue manager associated with this installation.



The **endmqm** command informs an application that the queue manager it is connected to is stopping; see Stopping a queue manager.

For the maintenance to proceed, applications must respond to an **endmqm** command by disconnecting from the queue manager and releasing any IBM WebSphere MQ libraries they have loaded. If they do not, you must find another way to force applications to release IBM WebSphere MQ resources, such as by stopping the applications.

You must also stop applications that are using the client libraries that are part of the installation. Client applications might be connected to a different queue manager, running a different installation of IBM WebSphere MQ. The application is not informed about queue managers in the current installation being shut down.

Any applications that continue to have IBM WebSphere MQ shared libraries from the installation loaded prevent you applying IBM WebSphere MQ maintenance. An application might disconnect from a queue manager, or be forcibly disconnected, but keep a IBM WebSphere MQ shared library loaded.

d) Stop any listeners associated with the queue managers, using the command:

```
endmqlsr -m QMgrName
```

4. Stop the IBM WebSphere MQ service for the installation.
   a) Right-click the **WebSphere MQ** icon in the taskbar > click **Stop WebSphere MQ**.
5. Load and apply the maintenance files for server or client installations:

- Interactively:
  a. Open the folder where the maintenance package has been extracted.
  b. Right-click on the maintenance program and select **Run as administrator** to start the loading process.
  c. Choose your installation language, and click **OK** .
  d. Continue to follow the instructions on screen.

  If you choose to load the files without applying them to an installation, you can apply the files later, as described in step “6” on page 163
- Silently:
  a. Open the folder where the maintenance package has been extracted.
  b. Modify the response file, `silent_install.resp` . For details on the properties you can specify in the response file, see Table 13 on page 162

Table 13. Properties used to install or uninstall a maintenance update

Property	Value	Description
MQPLOG	*path\file_name*	Pass a valid path to specify the log to be used during installation/uninstallation, for example `MQPLOG="C:\TEMP\UPDATEINSTALL.LOG"`

Table 13. Properties used to install or uninstall a maintenance update (continued)

Property	Value	Description
MQPINSTALLATION NAME	*Installation name*	The name of the installation to maintain. If there is only 1 installation (of any level) on the machine, this argument can be safely omitted. If there is more than 1 installation on the machine, `amqicsdn.exe` checks the value of `MQPINSTALLATIONNAME`. If one is not supplied, or the one that is supplied is unsuitable, then a GUI selection box appears. This selection box provides a list of installations to which this fix pack is applicable. If none are applicable, then `amqicsdn.exe` issues error message AMQ4781 and ends.
MQPBACKUPPATH	*path*	Specifies the directory to back up into during installation, for example `MQPBACKUPPATH="C:\BACKUP"` The directory, and any intermediate directories, you specify must already exist. If any one of the directories does not already exist, the install fails.
MQPREBOOT	0|1	Specifies what to do when a reboot is required, for example `MQPREBOOT=1`. If no value is supplied, you are prompted for what to do next. If `MQPREBOOT` is set to 0, a reboot is suppressed If `MQPREBOOT` is set to 1, the reboots go ahead without prompting.
MQPINUSEOK	0|1	Specifies whether to continue even if a file is found to be currently locked by another application. If you choose to continue even if a file is locked by another application, then you must reboot to complete fix pack installation. If no value is supplied, or if `MQPINUSEOK` is set to `0`, the installation fails if files are found to be in use by other applications. If `MQPINUSEOK` is set to 1, the installation is deferred until you reboot.

c. Open an elevated command prompt in the directory where the maintenance program was extracted.

d. Start the silent loading by entering the following command:

```
executableName -f responseFile
```

where:

– *executableName* is the name of the maintenance package. For example, for Version 7.5.0, Fix Pack 1: `7.5.0-WS-MQ-Windows-FP0001.exe`.

– *responseFile* is the full path and name of the response file.

6. Optional: Apply the maintenance to other server or client installations on the system:

• Interactively:

a. From the Windows start menu, select **Start > Programs > IBM WebSphere MQ > Apply Fix Pack *<V.R.M.L>***

   where

   V is the version number
   R is the release number
   M is the modification number
   L is the level of modification

b. Continue to follow the instructions on screen.

- Silently:

a. Open an elevated command prompt and navigate to the directory where the maintenance program was loaded. By default, the path is `C:\Program Files (x86)\IBM\source\WebSphere MQ <V.R.M.L>`

   where

   V is the version number
   R is the release number
   M is the modification number
   L is the level of modification

b. Enter the following command:

```
amqicsdn MQPINSTALLATIONNAME=
name MQPSILENT=1
```

   where *name* is the name of the installation that you want to apply maintenance to.

   You can add other properties to the command, as listed in Table 13 on page 162.

Applying maintenance to a client installation using MSI

7. As an alternative method for applying silent maintenance to client installations, you can use the MSI command **msiexec** from the command line.

   - To upgrade a client system with only a single installation, you can use a command similar to the following example, where PATH is the path of the client installation media:

```
msiexec /i "PATH\Windows\MSI\IBM WebSphere MQ.msi" /l*v <install_log_path>
 /q TRANSFORMS="1033.mst" REINSTALL=ALL REINSTALLMODE=vomus
```

   - For a multi installation client system, you can upgrade by using a command similar to the following example:

```
msiexec /i "PATH\Windows\MSI\IBM WebSphere MQ.msi" /l*v <install_log_path>
 /q TRANSFORMS=":InstanceId2.mst;1033.mst" REINSTALL=ALL REINSTALLMODE=vomus
```

What to do next

On a server installation, you must restart the IBM WebSphere MQ taskbar application manually after the maintenance application completes.

The IBM WebSphere MQ service is restarted automatically on the server, but the taskbar application is not restarted for any logged in sessions. Start the taskbar application in one of three ways:

1. Start the taskbar application manually from the start menu.
2. Log off and log back on again.
3. Run the command:

```
MQ_INSTALLATION_PATH\bin\amqmtbrn.exe -Startup
```

Related tasks

Stopping a queue manager

Related reference

dspmq

Windows V 7.5.0.9 *Installation of Microsoft Visual Studio 2013 C/C++ runtimes in a server fix pack*

From Version 7.5.0, Fix Pack 9, some elements of IBM WebSphere MQ require the Microsoft Visual Studio 2013 C/C++ (VS2013) runtimes. If the VS2013 runtimes are not present on the machine, they are installed when you install a fresh client, update a client to the 7.5.0.9 maintenance level or apply the 7.5.0.9 fix pack. In most cases, the installation of the runtimes does not require any manual intervention. However, if the installation fails when you are applying a server fix pack, you can set the **IBM_MQ_SKIP_VS2013** environment variable so that you can install the runtimes manually instead.

IBM_MQ_SKIP_VS2013 environment variable

The **IBM_MQ_SKIP_VS2013** environment variable is provided in case there is a problem with the runtime installation provided by the patch applier, `amqicsdn.exe`.

You can optionally set **IBM_MQ_SKIP_VS2013** before the applying the IBM WebSphere MQ server fix pack to influence the installation of the VS2103 runtimes. Setting **IBM_MQ_SKIP_VS2013** to any value causes `amqicsdn.exe` to skip the installation of the VS2103 runtimes.

Note: IBM WebSphere MQ will not function properly without the VS2103 runtimes. If you set **IBM_MQ_SKIP_VS2013**, it is then your responsibility to install these runtimes manually.

If you do have an issue with the installation of the runtimes with `amqicsdn.exe`, you must complete the following steps:

1. Install the runtimes yourself.

   The Microsoft redistributable runtime installers can be downloaded from the Microsoft web site, or can be found in the loaded fix pack files at the following (default) location: `C:\Program Files\IBM\source\MQ 7.5.0.9\prereqs\VS2013`.
2. Set **IBM_MQ_SKIP_VS2013** (to any value).
3. Rerun `amqicsdn.exe`.

Related messages

There are three IBM WebSphere MQ messages related to the installation of the VS2103 runtimes that might be issued by the server fix pack:

AMQ4754

`Install Microsoft Visual Studio 2013 C/C++ runtimes?....`

This message is issued when the patch applier, `amqicsdn.exe`, has checked your machine and determined that it does need to install the 32 bit runtimes, the 64 bit runtimes, or both. As the runtimes are required, answering N0 to this prompt exits the installation. (If you do not want this to happen, for example because you want install the runtimes yourself, set **IBM_MQ_SKIP_VS2013** as already described before rerunning `amqicsdn.exe`.)

AMQ4755

`Failed to install Microsoft Visual Studio 2013 C/C++ runtimes....`

The patch applier, `amqicsdn.exe`, determined that the 32 bit and/or 64 bit runtimes were not present on this machine and tried to install them. However, something went wrong during the installation and the process ended. If this happens you have two options:

- Consult the logs produced by `amqicsdn.exe` and the Microsoft redistributable installers, both of which can be found in your %TEMP% directory, and diagnose why this occurred.

- Install the runtimes yourself, then set **IBM_MQ_SKIP_VS2013** to any value, and rerun `amqicsdn.exe`.

AMQ4741
`The MQ %s update was stopped....`

This message is issued after you answer N0 to the message AMQ4754. It reminds you that you must install the runtimes.

Windows: Restoring the previous backup version

How to remove updates and restore the previous maintenance level using the Windows installer

Before you begin

1. If you are running on a server with multiple IBM WebSphere MQ installations, that is, IBM WebSphere MQ Version 7.0.1, Fix Pack 6 (or later), you must identify the installation. Make sure that the commands you enter run against the correct installation; see setmqenv.
2. You can apply and remove maintenance from a IBM WebSphere MQ MQI client that is not installed on the same server as a queue manager. You do not have to stop any queue managers or logon as administrator. Because you do not have to stop any queue managers, do not do steps "1" on page 151 to 3 in the following maintenance procedure.
3. If User Account Control (UAC) is enabled, the user who does the installation must have Administrative authority. You must elevate any command or command prompt by selecting **Run as Administrator**. If you do not, the error AMQ4353 is written in the installation log

About this task

If you applied maintenance to IBM WebSphere MQ, you can restore IBM WebSphere MQ to a previous level of maintenance. If you installed IBM WebSphere MQ at a particular maintenance level, a *Manufacturing Refresh*, you cannot restore IBM WebSphere MQ to an earlier maintenance level.

Procedure

1. Log on as an Administrator.
2. Stop all applications using the IBM WebSphere MQ installation.

   If you use the IBM WebSphere MQ Managed File Transfer (MFT) component, ensure that any MFT agents have finished all of the file transfers that they were engaged in. There should be no incomplete transfers associated with the agents, and their SYSTEM.FTE.STATE queues should contain no messages.
3. End all the activity of queue managers associated with the IBM WebSphere MQ installation.
   a) Run the **dspmq** command to list the state of all the queue managers on the system.

      Run either of the following commands from the installation that you are updating:

      ```
      dspmq -o installation -o status
      dspmq -a
      ```

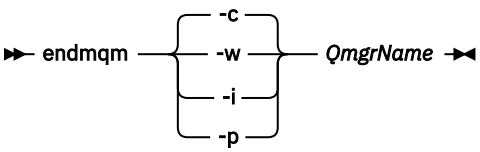
      dspmq -o installation -o status displays the installation name and status of queue managers associated with all installations of IBM WebSphere MQ.

      dspmq -a displays the status of active queue managers associated with the installation from which the command is run.

   b) Run the **MQSC** command, `DISPLAY LSSTATUS(*) STATUS` to list the status of listeners associated with a queue manager.

      ```
      echo DISPLAY LSSTATUS(*) STATUS | runmqsc QmgrName
      ```

   c) Run the **endmqm** command to stop each running queue manager associated with this installation.



The **endmqm** command informs an application that the queue manager it is connected to is stopping; see Stopping a queue manager.

For the maintenance to proceed, applications must respond to an **endmqm** command by disconnecting from the queue manager and releasing any IBM WebSphere MQ libraries they have loaded. If they do not, you must find another way to force applications to release IBM WebSphere MQ resources, such as by stopping the applications.

You must also stop applications that are using the client libraries that are part of the installation. Client applications might be connected to a different queue manager, running a different installation of IBM WebSphere MQ. The application is not informed about queue managers in the current installation being shut down.

Any applications that continue to have IBM WebSphere MQ shared libraries from the installation loaded prevent you applying IBM WebSphere MQ maintenance. An application might disconnect from a queue manager, or be forcibly disconnected, but keep a IBM WebSphere MQ shared library loaded.

d) Stop any listeners associated with the queue managers, using the command:

```
endmqlsr -m QMgrName
```

4. Stop the IBM WebSphere MQ service for the installation.
   a) Right-click the **WebSphere MQ** icon in the taskbar > click **Stop WebSphere MQ**.
5. Remove the maintenance interactively, or silently using a command.

- Interactively:
  a. For each installation of IBM WebSphere MQ that has had maintenance applied, you are presented with one of the following icons in the Windows start menu:
     i) **Start > Programs > IBM WebSphere MQ > Remove Refresh Pack 7.5.n.0 *(installation name)***
     ii) **Start > Programs > IBM WebSphere MQ > Remove Fix Pack 7.5.n.n *(installation name)***
  b. Select the installation you want to maintain and click **Remove** to start the process.

  This returns the installation to the state it was in before the maintenance package was applied.
- Silently:
  a. Open an elevated command prompt and enter the following command:

```
amqicsdn.exe MQPINSTALLATIONNAME=name MQPUNINST=1 MQPSILENT=1
```

  where *name* is the name of the installation that you want to remove maintenance from.

  You can add other properties to the command, as listed in Table 14 on page 167.

Table 14. Properties used to install or uninstall a maintenance update

Property	Value	Description
MQPLOG	*path\file_name*	Pass a valid path to specify the log to be used during installation/uninstallation, for example MQPLOG="C:\TEMP\UPDATEINSTALL.LOG"

Table 14. Properties used to install or uninstall a maintenance update (continued)		
Property	**Value**	**Description**
MQPINSTALLATION NAME	*Installation name*	The name of the installation to maintain. If there is only 1 installation (of any level) on the machine, this argument can be safely omitted. If there is more than 1 installation on the machine, `amqicsdn.exe` checks the value of `MQPINSTALLATIONNAME`. If one is not supplied, or the one that is supplied is unsuitable, then a GUI selection box appears. This selection box provides a list of installations to which this fix pack is applicable. If none are applicable, then `amqicsdn.exe` issues error message AMQ4781 and ends.
MQPBACKUPPATH	*path*	Specifies the directory to back up into during installation, for example `MQPBACKUPPATH="C:\BACKUP"` The directory, and any intermediate directories, you specify must already exist. If any one of the directories does not already exist, the install fails.
MQPREBOOT	0|1	Specifies what to do when a reboot is required, for example `MQPREBOOT=1`. If no value is supplied, you are prompted for what to do next. If `MQPREBOOT` is set to 0, a reboot is suppressed If `MQPREBOOT` is set to 1, the reboots go ahead without prompting.
MQPINUSEOK	0|1	Specifies whether to continue even if a file is found to be currently locked by another application. If you choose to continue even if a file is locked by another application, then you must reboot to complete fix pack installation. If no value is supplied, or if `MQPINUSEOK` is set to `0`, the installation fails if files are found to be in use by other applications. If `MQPINUSEOK` is set to 1, the installation is deferred until you reboot.

6. Optional: If you no longer need the maintenance files that were loaded onto the system before maintenance was applied, you can remove them using **Add/Remove programs** or **Programs and Features** from the Control Panel.

   If you want to remove a maintenance file silently, run the following command:

   ```
   <patch_install_files>\_IBM WebSphere MQ (fix pack 7.5.0.1 files)_installation\Change IBM
   WebSphere MQ (fix pack 7.5.0.1 files) Installation.exe" -i silent
   ```

   where <patch_install_files> is the installation directory where maintenance files are installed.

   By default, this directory is `c:\Program Files [(x86)]\IBM\source\WebSphere MQ 7.5.0.1`

   Notes:

   a. Run the command from outside the directory, otherwise the directory is not removed.

   b. If you omit **-i silent**, the command initiates the Graphical User Interface installer.

What to do next

On a server installation, you must restart the IBM WebSphere MQ taskbar application manually after the maintenance application completes.

The IBM WebSphere MQ service is restarted automatically on the server, but the taskbar application is not restarted for any logged in sessions. Start the taskbar application in one of three ways:

1. Start the taskbar application manually from the start menu.
2. Log off and log back on again.
3. Run the command:

```
MQ_INSTALLATION_PATH\bin\amqmtbrn.exe -Startup
```

Related reference

dspmq

Related information

Stopping a queue manager

Applying maintenance level upgrades to multi-instance queue managers

Use multi-instance queue managers to reduce the outage caused by applying maintenance updates. Follow these steps to apply maintenance to a multi-instance queue manager.

Before you begin

Maintenance is applied to the WebSphere MQ installation on a server and not to individual queue managers. You must stop all the queue managers, and any WebSphere MQ service, on a server before you apply maintenance.

If you want a queue manager to keep running while maintenance is applied, you must configure it as a *multi-instance* queue manager, and have a standby instance running on another server. If a queue manager is an existing single instance queue manager, you must convert it to a multi-instance queue manager. See the topic, Multi-instance queue managers for prerequisites and guidance how to create a multi-instance queue manager.

You can create a multi-instance queue manager from v7.0.1 onwards. If you are running multi-instance queue managers, you then can apply a maintenance update to a *running* queue manager by switching the active instance to a different server.

Typically active and standby installations are maintained at the same maintenance level. Consult the maintenance instructions for each upgrade. Consult the instructions to see if it is possible to run the active and standby instances at different maintenance levels. Check whether fail over from higher to lower, or only lower to higher maintenance level is possible.

The instructions for applying a maintenance update might require you to stop a multi-instance queue manager completely.

If you have a primary server for running active queue manager instances, and a secondary server that runs standby instances, you have a choice of updating the primary or secondary server first. If you update the secondary server first, you must switch back to the primary server when both servers have been updated.

If you have active and standby instances on several servers, you must plan in what order you update the servers to minimize the disruption caused by ending the active instances on each server you update.

About this task

Combine the steps in this task with the maintenance update procedure for applying maintenance to a WebSphere MQ server installation.

Procedure

1. Where the maintenance update procedure instructs you to stop all running queue managers, or quiesce WebSphere MQ do the following instead:

   The maintenance update procedure varies by platform; see "Applying and removing maintenance level updates on UNIX, Linux, and Windows" on page 141.

   a) If the queue manager is running as standby:

   - On Windows, UNIX, and Linux platforms, end the standby with the **endmqm** `-x` *QMgrName* command.

   b) If the queue manager is running as the active instance:

   - On Windows, UNIX, and Linux platforms, end the instance and transfer control to the standby instance with the **endmqm** command. For example, **endmqm** *-shutdown_option* `-s` *QMgrName*, where *-shutdown_option* is an optional parameter specifying the type of shutdown. For more information, see endmqm.

     If there is no standby instance running, the command fails, and you must start a standby instance on a different server.

   c) If a queue manager is running as a single instance queue manager, you have no alternative but to stop the queue manager before applying the maintenance update.

   When you complete this step, no queue manager instances are left running on the server you intend to update.
2. Continue with the maintenance update procedure, following the step to issue the **endmqm** command, or quiesce WebSphere MQ and apply maintenance to the WebSphere MQ server.
3. When you have completed the maintenance update, restart all the queue managers on the WebSphere MQ server, permitting standby instances:

   - On Windows, UNIX, and Linux platforms, use the **strmqm** `-x` *QmgrName* command.
4. Repeat the procedure on the standby server, to update its maintenance level.
5. If necessary, switch the active instances back to the primary servers:

   - On Windows, UNIX, and Linux platforms, use the **endmqm** *-shutdown_option* `-s` *QMgrName* command, and the restart the instances using the **strmqm** `-x` *QmgrName* command.

Related tasks

"Applying and removing maintenance level updates on UNIX, Linux, and Windows" on page 141
When you apply and remove maintenance level updates to IBM WebSphere MQ, no migration is required. Maintenance level updates are applied either as a fix pack, or by manually applying an interim fix. The following topics describe how to apply and remove fix packs on UNIX, Linux, and Windows.

Migrating queue managers to new-function fix packs

This scenario illustrates running different levels of queue manager from a single installation using new-function fix packs. New function fix-packs are available on platforms other than z/OS. It contrasts migrating a queue manager to new command levels in new-function fix packs, to migrating a queue manager to a new command level in a new release. The scenario explains the relationship between new-function fix packs and maintenance fix packs.

Before you begin

In this section, IBM WebSphere MQ Version 7.1 is used as the current release, and the release is denoted by `r`; the subsequent release is denoted by R.

The scenario starts with a single installation of IBM WebSphere MQ Version 7.1, `Inst_1`. `Inst_1` is the primary installation; see Figure 26 on page 174. For illustration, there are two queue managers, `QM1` and `QM2` . `QM1` stays at the `7r0` command level, `QM2` moves to the highest command level available.

The use of version numbers and command levels is illustrative, and does not imply anything about future releases.

About this task

Figure 25 on page 171 has time advancing down the Y-Axis, as new fix packs are released. On the X-Axis are different command levels. As a queue manager is migrated to a new command level, it shifts across the diagram. Each column represents the fix levels a queue manager at a particular command level can run at.

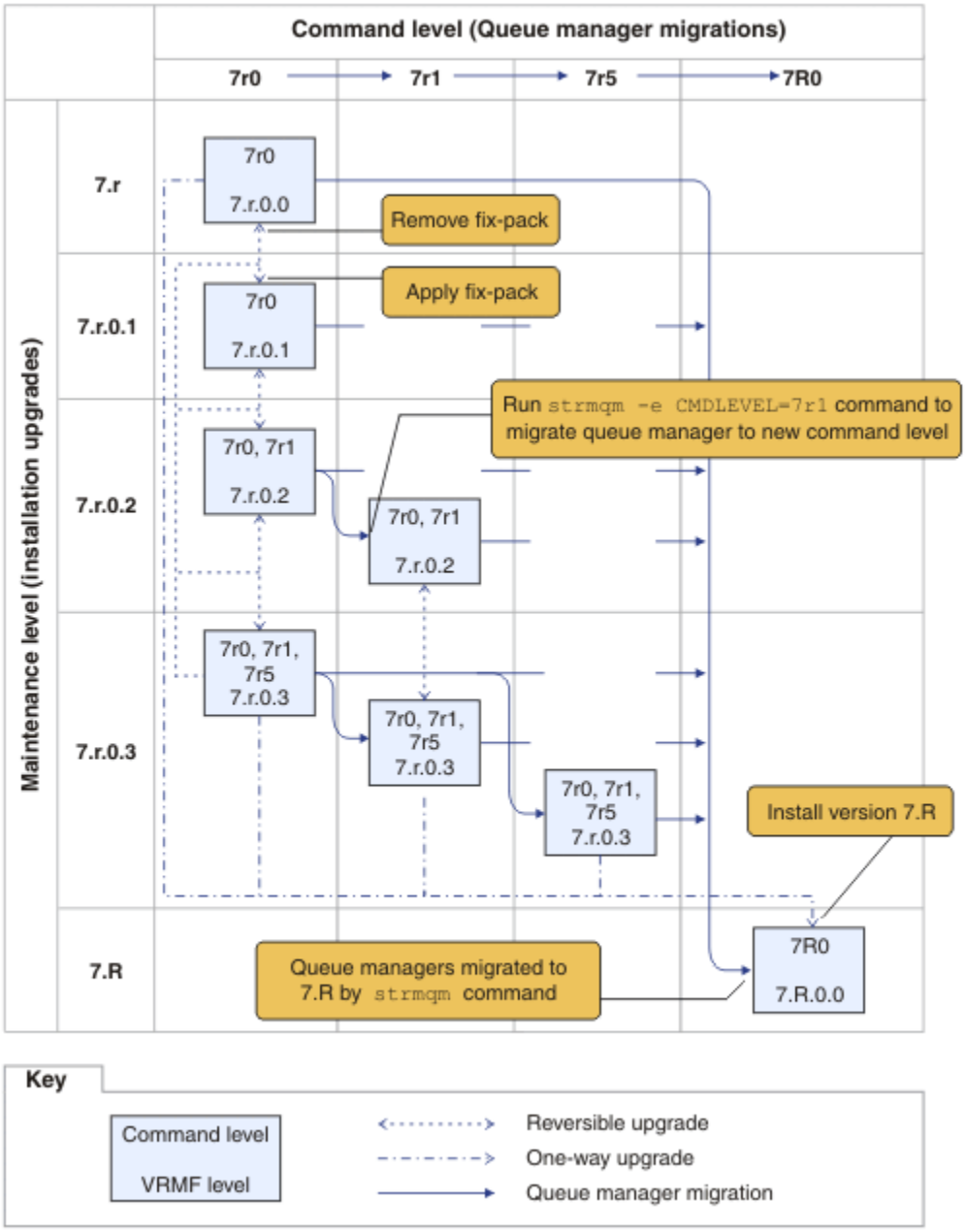


Figure 25. Migration of queue managers to new command levels using new-function fix packs

Figure 25 on page 171 is a little complicated, but it captures lots of details about new-function fix packs to help you remember them. The steps in the task explain the details in the figure. Some of the principle features of Figure 25 on page 171 are explained in the following list:

Maintenance level and Command level

The maintenance level is a fix pack with a V.R.M.F. code; see “The version naming scheme for IBM WebSphere MQ on UNIX, Linux, and Windows” on page 8. V.R.M.F codes are one to four digits, always punctuated by periods. Trailing zeros are sometimes omitted in descriptions, but never when a V.R.M.F code is used to label a fix pack. Version 7.5 is an example of using a V.R.M.F code to describe the version of IBM WebSphere MQ.

The command level is the command level property of a queue manager; see CommandLevel (MQLONG). Command levels are three-digit codes.

Command levels and versions are related. Up to Version 7.1 the command level and the first three digits of the V.R.M.F. code always matched. From Version 7.1, with the introduction of new-function fix packs, the command level of a queue manager can be greater than the first three digits of an installation. The difference arises, if the queue manager has been associated with a new command level using the **strmqm** command.

From Version 7.1 the rule that links command levels and V.R.M.F levels has been changed. The rule is that when a new version of IBM WebSphere MQ is released, it has a command level greater than released in a new-function fix pack in the previous release. Usually this means that a new release of IBM WebSphere MQ changes the version or release level, rather than the maintenance level.

In Figure 25 on page 171, the maintenance level, on the Y-Axis is labeled with V.R.M.F codes, and the command level, on the X-Axis, with command levels. Note how the illustrative release of `7.R` increases the released command level from `7r0` to `7R0`, and `7R0` exceeds the highest command level shipped in a new-function fix pack, `7r5` .

Reversible and One-way upgrades

The mechanism to apply and remove fix packs varies by platform. You can apply any fix pack that changes only the maintenance or fix level of a release to an installation. Fix pack application is reversible. When you remove a fix pack, you restore the previous release level. So applying `7.r.0.3` to `7.r.0.1`, and then removing it, restores the installation to `7.r.0.1` .

Sometimes, you can change an installation to a particular V.R.M.F level by upgrading the installation with a "manufacturing refresh". If you install a manufacturing refresh, you can only return to the earlier release level by uninstalling, and reinstalling; see “Upgrade, migration, and maintenance of IBM WebSphere MQ on UNIX, Linux, and Windows” on page 25.

Applying a manufacturing refresh to modify the maintenance and fix level of a release is the same process as upgrading to a new version or release of IBM WebSphere MQ. Neither can be reversed without uninstalling.

However there is a particular aspect of upgrading to a new version or release that is different from upgrading to a new maintenance or fix level. If you start a queue manager after a version or release upgrade, the command level of the queue manager is automatically increased. You can then no longer start the queue manager with the installation from the previous release.

On the diagram, an irreversible upgrade is shown by the "One-way" arrow between `7.r` and `7.R`. To prevent an accidental migration, you can rename the new installation. After renaming, rerun the **setmqm** command to associate a queue manager with the new release before running the **strmqm** command to migrate it.

If the upgrade applies only to the maintenance or fix level, you can restart the queue manager with the previous installation, if you reinstall it.

Manufacturing refresh maintenance releases are not distinguished from applying and removing fix packs on the diagram. Both are represented by reversible arrows in Figure 25 on page 171.

Multiple installations

You might choose to have a different installation for each maximum command level supported by an installation. Each column on the diagram would represent a different installation.

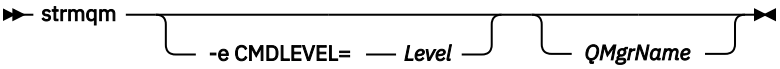
You need only one installation at Version 7.1 to be able to select any command level released with Version 7.1 for a queue manager. Eventually, if you intend to run Version 7.1 and version 7.R in parallel, you must have two installations. The scenario that follows uses a single installation.

Another variation is to follow the "rolling fix pack" approach described in “UNIX, Linux, and Windows: Staging maintenance fixes” on page 183. You can maintain two installations at Version 7.1, one at the current fix level, and one either at a later or earlier fix level. You might then install version 7.R as a third installation, or replace the Version 7.1 installation at the older fix level.

Migrating queue managers

The migration paths for queue managers are shown by solid arrows on the diagram. Some of the solid arrows are broken, to avoid cluttering the diagram with too many lines. If the migration to a higher command level jumps command levels, you do not have to migrate it through the intervening commands levels.

To migrate a queue manager to a higher command level in a new-function fix pack, you must start the queue manager with a special parameter:



Level is the three-digit command level.

The queue manager stops immediately the migration process is complete. When you next start it, it runs at the new command level. The queue manager cannot be restarted at a lower command level. This rule means that you must associate the queue manager with an installation that includes a command level at least as great as the current command level of the queue manager.

Restoring queue managers

To restore a queue manager to a lower command level, you must back up the queue manager before you migrate it to the higher command level.

Procedure

This procedure keeps both `QM1` and `QM2` at the current maintenance level, `QM1` at command level `7r0`, and `QM2` at the latest command level.

1. Download fix pack 7.r.0.1, when it is released.

   See Fix Central.

   The initial system has two queue managers running `7.r.0.0` at command level `7r0`; see Figure 26 on page 174.

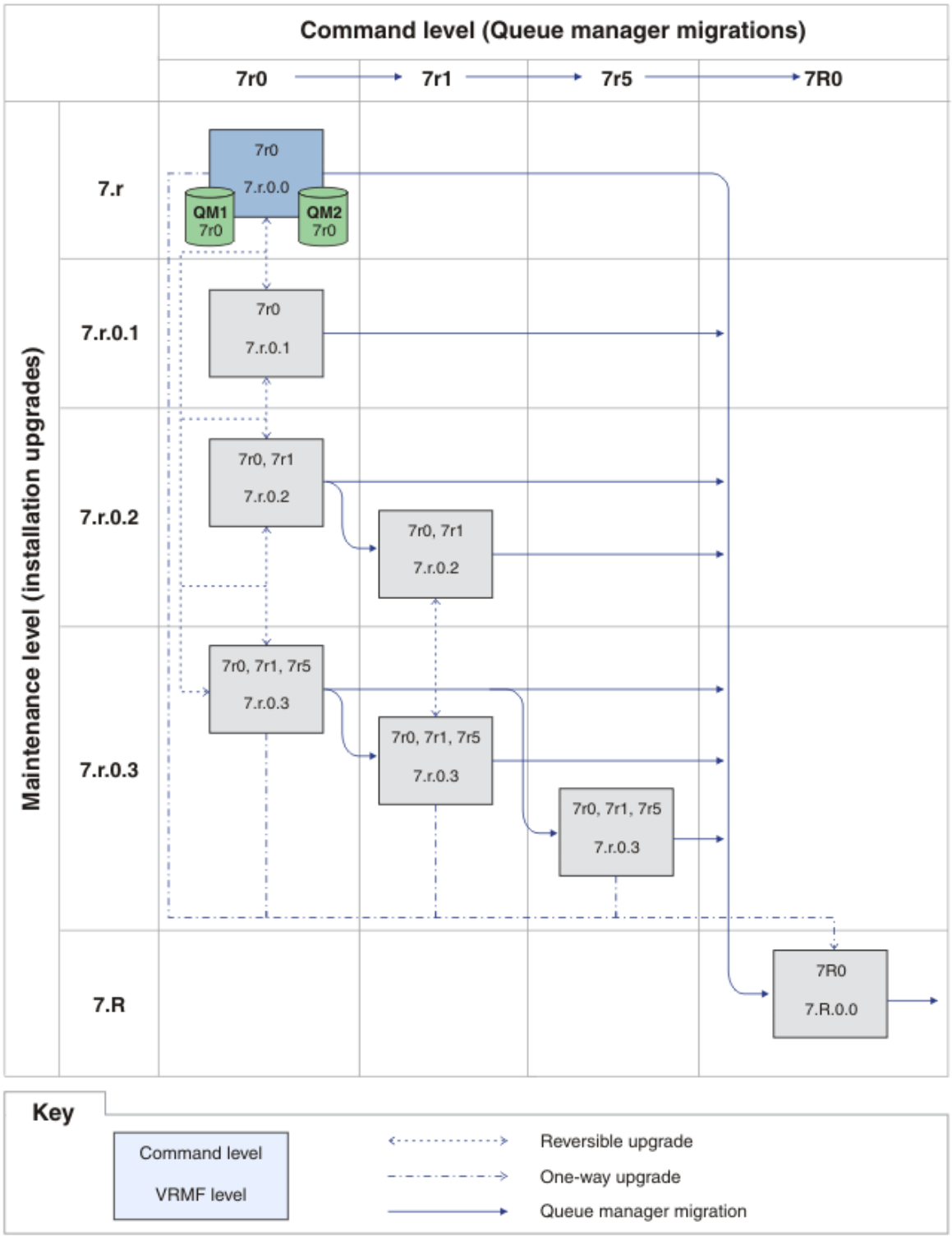


Figure 26. Initial state, QM1 and QM2 at command level 7r0, and fix level 7.r.0.0

2. Apply fix pack 7.r.0.1 to `Inst_1`.

   Follow the task for your platform in "Applying and removing maintenance level updates on UNIX, Linux, and Windows" on page 141.

3. Restart the queue managers.

   Both queue managers are now running using `Inst_1` at the `7.r.0.1` maintenance level, and the `7r0` command level; see Figure 27 on page 175 .

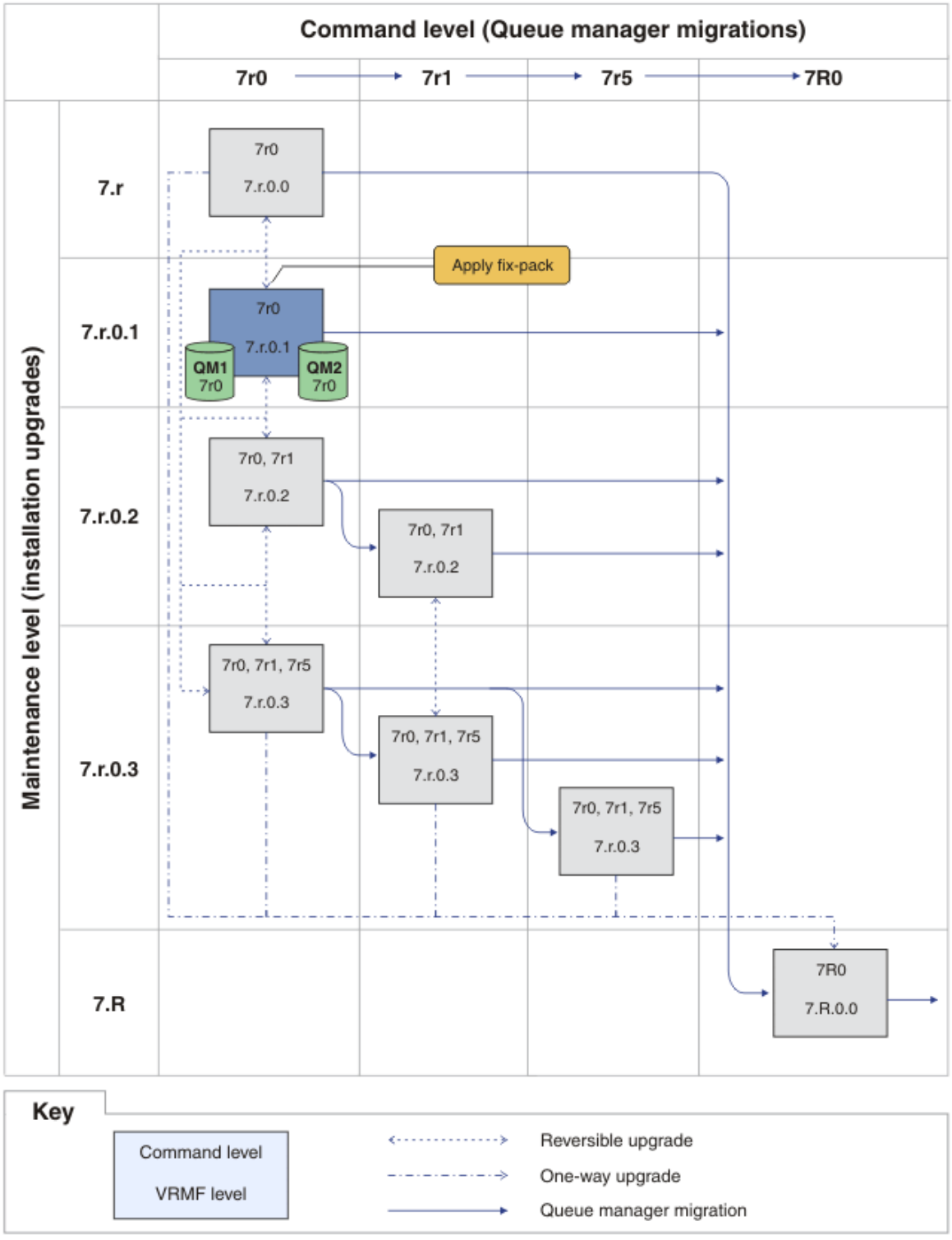


Figure 27. QM1 and QM2 at command level `7r0`, and fix level `7.r.0.1`

4. Apply fix pack 7.r.0.2.
   a) Repeat steps “1” on page 173 and “2” on page 174 with fix pack 7.r.0.2 .
5. Restart QM1.

   QM1 is now running using Inst_1 at the 7.r.0.2 maintenance level, and the 7r0 command level. The queue manager is not automatically migrated to the 7r1 command level.
6. Migrate QM2 to the 7r1 command level.

   ```
   strmqm -e CMDLEVEL=711 QM2
   ```

   QM2 is using Inst_1 at the 7.r.0.2 maintenance level, and has been migrated to the 7r1 command level.
7. Restart QM2.

   QM2 is now running using Inst_1 at the 7.r.0.2 maintenance level, and the 7r1 command level; see Figure 28 on page 177 .

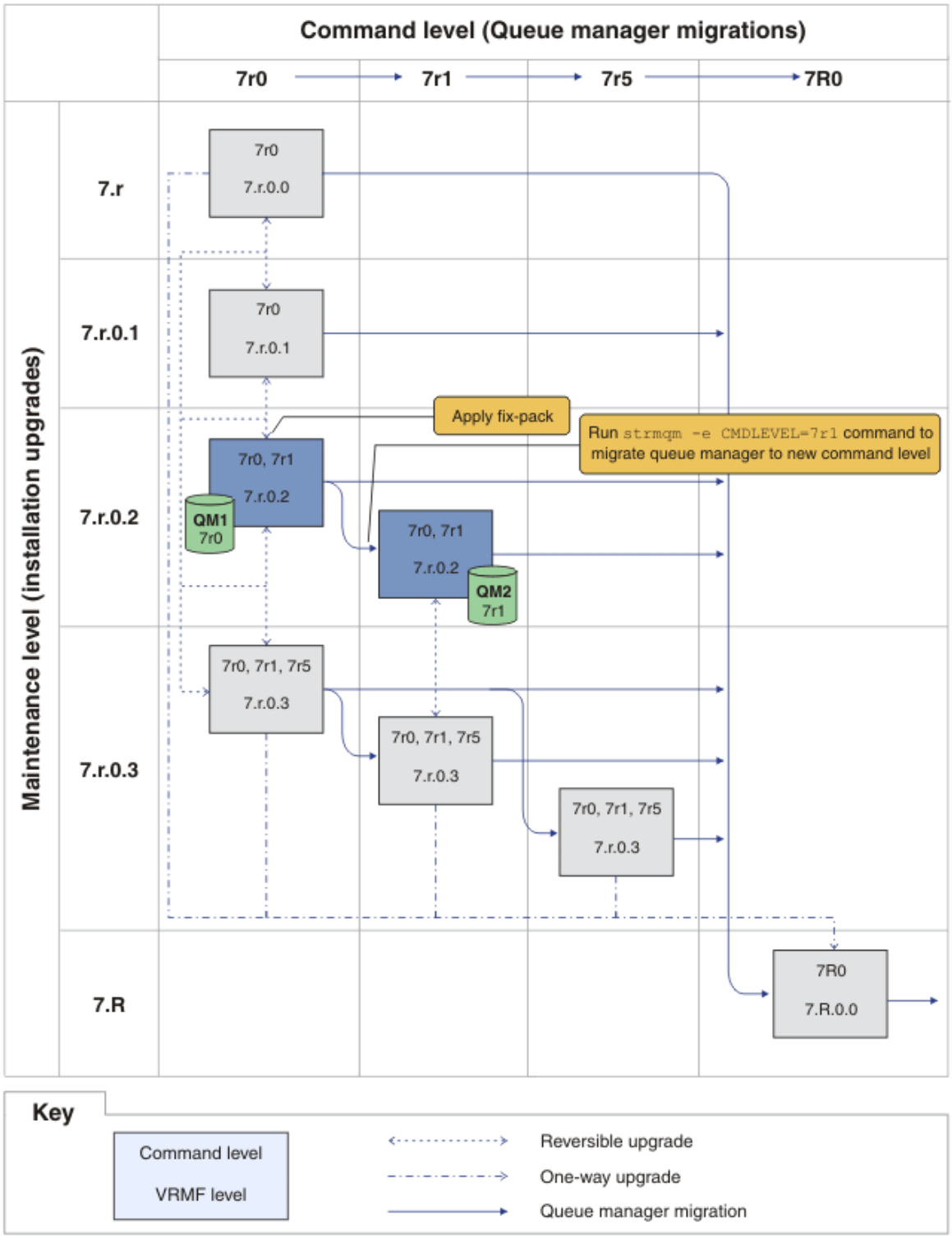


Figure 28. QM1 at command level 7r0 and fix level 7.r.0.2; QM2 at command level 7r1 and fix level 7.r.0.2

8. Apply fix pack 7.r.0.3 and migrate QM2 to the 7r5 command level.
   a) Repeat steps "4" on page 176 to "5" on page 176 with fix pack 7.r.0.3 .

b) Repeat steps “6” on page 176 to “7” on page 176 with command level `7r5` .

`QM1` is using `Inst_1` at the `7.r.0.3` maintenance level, and is running at the `7r0` command level.
`QM2` is using `Inst_1` at the `7.r.0.3` maintenance level, and has been migrated to the `7r5` command level; see Figure 29 on page 179 .

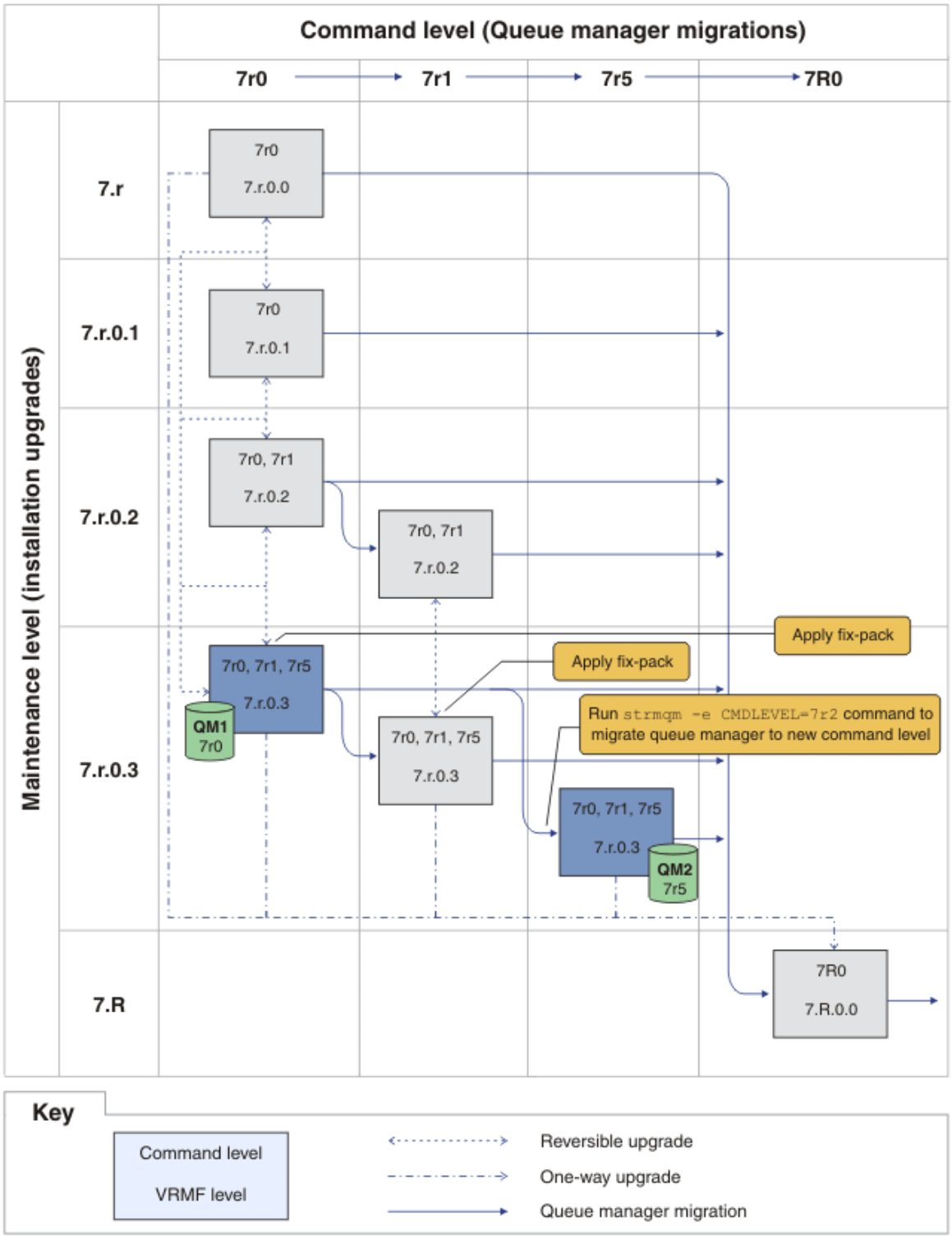


Figure 29. QM1 at command level 7r0 and fix level 7.r.0.3; QM2 at command level 7r5 and fix level 7.r.0.3

9. Migrate QM2 to version 7.R
   - On UNIX, Linux, and Windows:

a) Install version 7.R, with the installation name Inst_2, alongside Version 7.1 .

b) Set up the local environment to the installation Inst_2.

Windows:

```
"Inst_2_INSTALLATION_PATH
\bin\setmqenv" -s
```

The -s option sets up the environment for the installation that runs the **setmqenv** command.

UNIX:

```
.
Inst_2_INSTALLATION_PATH/bin/setmqenv -s
```

c) Run the **setmqm** command to associate QM2 with Inst_2.

```
setmqm -m QM2 -n Inst_2
```

d) Run the **strmqm** command to start QM2 and migrate it to version 7.R.

```
strmqm QM2
```

QM1 is using Inst_1 at the 7.r.0.3 maintenance level, and is running at the 7r0 command level.
QM2 is using Inst_2 at the 7.R.0.0 maintenance level, and has been migrated to the 7R0 command level; see Figure 29 on page 179 .
Inst_1 remains the primary installation.

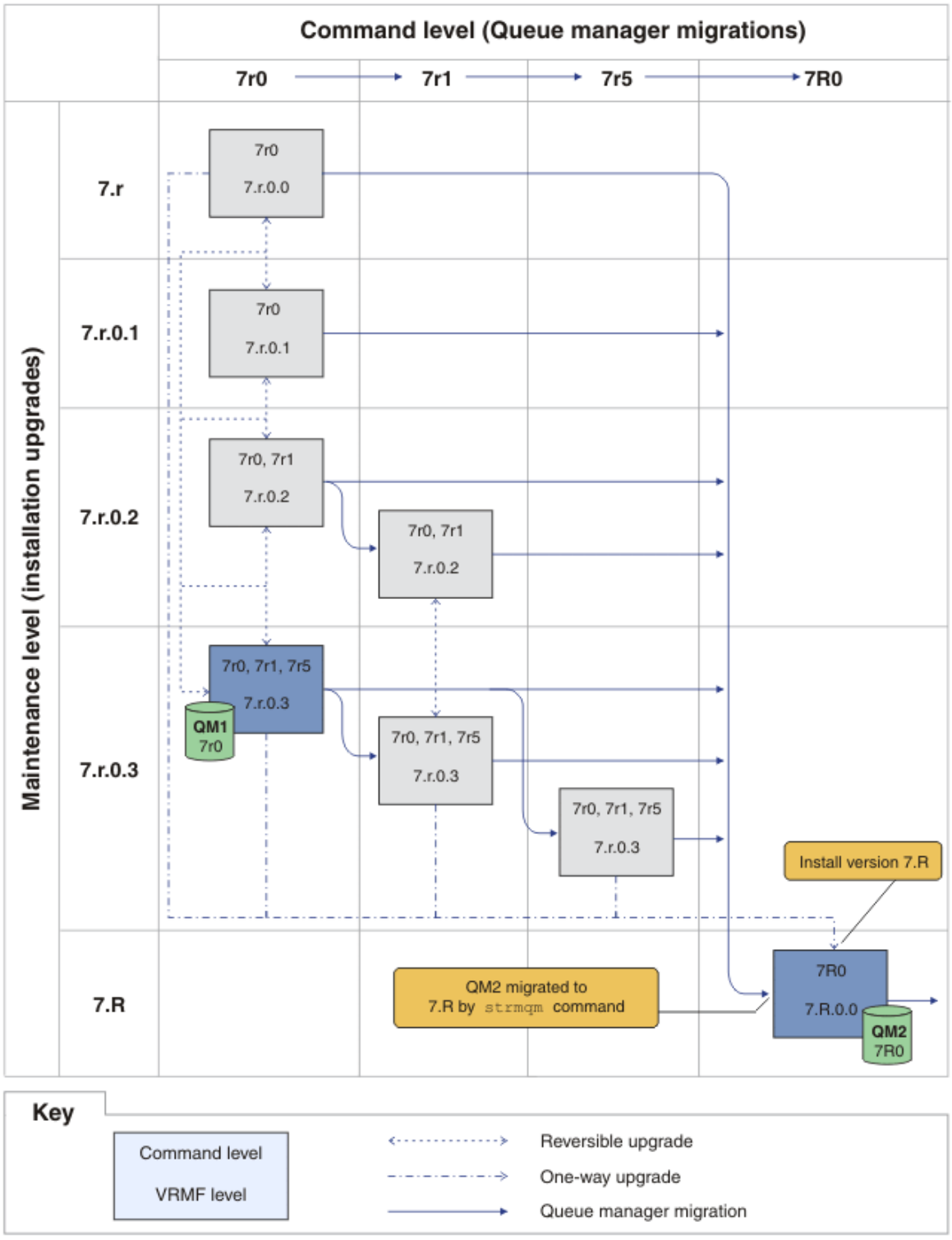


Figure 30. QM1 at command level 7r0 and fix level 7.r.0.3; QM2 at command level 7R0 and fix level 7.R.0.0

Related concepts

"Multi-installation queue manager coexistence on UNIX, Linux, and Windows" on page 31

You can install multiple copies of IBM WebSphere MQ for UNIX, Linux, and Windows on the same server. The installations must be at Version 7.1 or later, with one exception. One Version 7.0.1 installation, at fix pack level 6, or later, can coexist with multiple Version 7.1, or later installations.

“New function in maintenance level upgrades” on page 24
On platforms other than z/OS, IBM might introduce new functions between releases in maintenance level upgrades such as fix packs. A maintenance level upgrade including new function increases the maximum command level of an installation. When you apply the maintenance level upgrade, the installation supports the new command level. A queue manager associated with the installation is not able to use the new function until you have run **strmqm** to set the queue manager to the new command level.

“Queue manager coexistence in Version 7.5” on page 28
Queue managers, with different names, can coexist on any server as long as they use the same IBM WebSphere MQ installation. On UNIX, Linux, and Windows, different queue managers can coexist on the same server and be associated with different installations.

“The version naming scheme for IBM WebSphere MQ on UNIX, Linux, and Windows” on page 8
On UNIX, Linux, and Windows, IBM WebSphere MQ releases have a four-digit Version, Release, Modification, and Fix (VRMF) level code. The version and release parts of the code are significant; they identify the service life of a release. To run a queue manager at a different VR level, you must migrate the queue manager, its applications, and the environment in which it runs. Depending on the migration path, the migration might require more or less effort.

“Upgrade, migration, and maintenance of IBM WebSphere MQ on UNIX, Linux, and Windows” on page 25
You can install new releases of IBM WebSphere MQ to upgrade IBM WebSphere MQ to a new maintenance, release, or version level. Multiple installations at the same or different levels can coexist on the same UNIX, Linux, and Windows server. You can apply maintenance level upgrades to upgrade the maintenance or fix level. Applying maintenance level upgrades cannot change the version or release level of IBM WebSphere MQ. Maintenance level upgrades can be reversed, installations cannot be reversed.

“IBM WebSphere MQ maintenance” on page 20
Maintenance is the application of a reversible fix. Any changes to queue manager data are compatible with the previous code level.

“IBM WebSphere MQ migration” on page 22

“IBM WebSphere MQ upgrade” on page 21
The term upgrade, used loosely, covers both applying maintenance and modifying the version or release of a product. Upgrading is a change in any of the four V.R.M.F digits. After an upgrade has been applied, as long as no queue managers have been restarted, it can be backed out. The details vary by platform and how the upgrade was applied. Once a queue manager has been restarted, it depends on a number of factors whether the upgrade can be reversed. It depends on the platform, the V.R.M.F level, and the functions and command levels used by the queue manager at the new level.

Related tasks

“Applying and removing maintenance level updates on UNIX, Linux, and Windows” on page 141
When you apply and remove maintenance level updates to IBM WebSphere MQ, no migration is required. Maintenance level updates are applied either as a fix pack, or by manually applying an interim fix. The following topics describe how to apply and remove fix packs on UNIX, Linux, and Windows.

“UNIX, Linux, and Windows: Staging maintenance fixes” on page 183
Use multiple installations of IBM WebSphere MQ on the same server to control the release of maintenance fixes.

Related reference

CommandLevel (MQLONG)

Querying the maintenance level

Query the IBM WebSphere MQ maintenance level by running the **dspmqver** command

About this task

After an update to the initial installation, the version indicates the maintenance level to which the product has been updated. For example, before applying any maintenance, the version is 7.0.1.0. As maintenance is applied the last two digits will be updated, for example to 7.0.1.3.

Procedure

To view the version use the `dspmqver` command. At a command prompt, enter the following command: `dspmqver`.

The resulting messages include the IBM WebSphere MQ version number, which shows the maintenance level.

Related reference

dspmqver

UNIX, Linux, and Windows: Staging maintenance fixes

Use multiple installations of IBM WebSphere MQ on the same server to control the release of maintenance fixes.

Before you begin

Set up your configuration modelled on the first row of Figure 31 on page 184. You can apply this scenario to any version of IBM WebSphere MQ fromIBM WebSphere MQ Version 7.1 onwards. In this scenario it is assumed you have a number of applications and two queue managers, `QM1` and `QM2`, running on a server. IBM WebSphere MQ Version 7.0.1 is not installed on the server.

1. Install two copies of IBM WebSphere MQ Version 7.1. In the example, they are named `Inst_1` and `Inst_2`.
2. Make `Inst_1` primary by running **setmqinst**.
3. Associate all the queue managers on the server with `Inst_1` by running **setmqm**.
4. Start all the queue managers on the server.
5. Show and connect all direct connections with the queue managers associated with `Inst_1` in IBM WebSphere MQ Explorer.
6. Set up remote connections to all the queue managers in each instance of IBM WebSphere MQ Explorer.

About this task

You can install multiple copies of IBM WebSphere MQ on a server to stage the release of IBM WebSphere MQ fixes. Figure 31 on page 184 illustrates a way of using two installations to roll out fixes. In this approach, you maintain two fix levels on a server, with the aim of getting all queue managers and applications to the production fix level before replacing the previous level on fix pack with the next level.

Which installation an application uses is driven by the queue manager an application connects to. The **setmqm** command associates a queue manager with an installation. You can associate a queue manager with a different installation as long as the installation is at the same or higher command level. In this example, all the installations are at the same command level. You can associate or reassociate a queue manager with either of the installations running any of the fix packs.

In the example, an application links to the primary installation. When it connects to a queue manager, IBM WebSphere MQ switches the linkage to the installation associated with the queue manager; see “Multi-installation queue manager coexistence on UNIX, Linux, and Windows” on page 31.

For applications built with the link options described in the product documentation, the simplest way to configure the link library search path for IBM WebSphere MQ applications is to make an installation primary. Only if it is important to pick up a fix in the IBM WebSphere MQ link library itself, must you review the search path. Either you must make the installation with the IBM WebSphere MQ link library fix primary, or make a local adjustment for the application, perhaps by running the **setmqenv** command.

Running commands is a different matter. Commands are always run from the primary installation, or the installation you have selected by running the **setmqenv** command. If you run a command from the wrong installation, the command fails. For example, if `QM1` is associated with `Inst_1`, running the Windows command, `Inst_2_Installation_path/bin/strmqm QM1` fails.

If you are using IBM WebSphere MQ Explorer and you have two installations, you also have two IBM WebSphere MQ Explorer instances. One linked to one installation, and one to the other. Each IBM WebSphere MQ Explorer shows locally connected queue managers that are associated with the same installation as the instance of IBM WebSphere MQ Explorer. To monitor all the queue managers on a server, set up remote connections to the queue managers associated with the other installations.

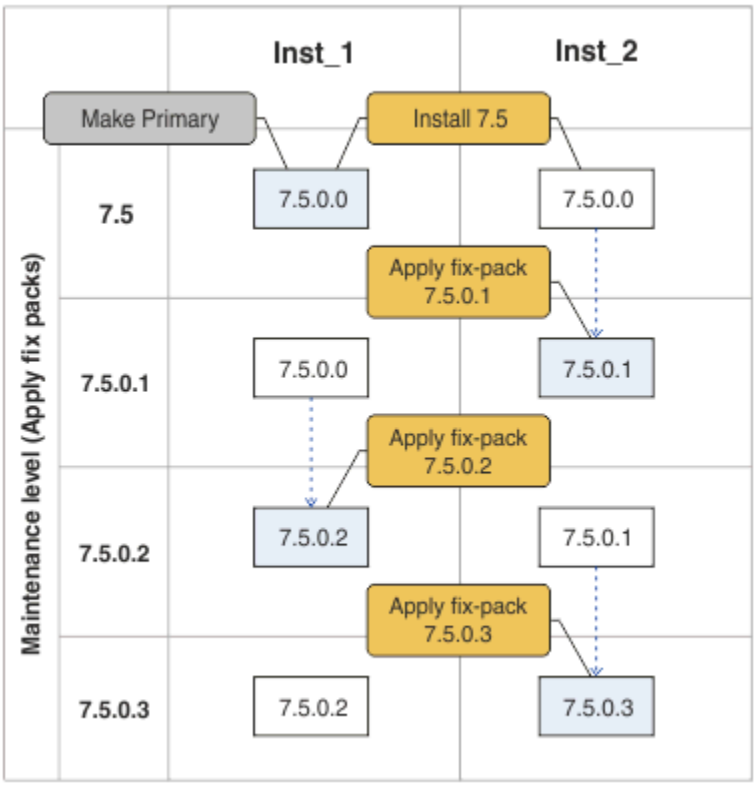


Figure 31. Rolling fix releases

Procedure

1. Download fix pack 7.1.0.1, when it is released.

   See Fix Central.
2. Apply fix pack 7.1.0.1 to `Inst_2`.

   Follow the task for your platform in “Applying and removing maintenance level updates on UNIX, Linux, and Windows” on page 141.
3. Verify `Inst_2`.
4. Transfer the queue managers to `Inst_2` one at a time.
   a) Stop `QM1` and the applications connected to it.

The **endmqm** command informs an application that the queue manager it is connected to is stopping; see Stopping a queue manager.

b) Set up the local environment to the installation `Inst_2`.

Windows:

```
"Inst_2_INSTALLATION_PATH\bin\setmqenv" -s
```

The `-s` option sets up the environment for the installation that runs the **setmqenv** command.

UNIX:

```
. Inst_2_INSTALLATION_PATH/bin/setmqenv -s
```

c) Associate the queue manager with `Inst_2`.

```
setmqm -m QM1 -n Inst_2
```

d) Start `QM1`

```
strmqm QM1
```

e) Repeat substeps c and d for `QM2`.

f) Set up IBM WebSphere MQ Explorer for `Inst_2`.

i) Start the `Inst_2` instance of IBM WebSphere MQ Explorer

Tip: On Windows, hover over the IBM WebSphere MQ icon in the system tray. The hover help shows the installation name associated with the IBM WebSphere MQ Explorer instance.

ii) Click **IBM WebSphere MQ** > **Queue Managers** > **Show/Hide Queue Managers...** >

iii) Click each directly connected queue manager listed in the **Hidden Queue Managers** list > **Show**.

iv) Click **Close**.

5. Set `Inst_2` primary.

Windows:

```
"Inst_2_INSTALLATION_PATH\bin\setmqinst" -i -n Inst_2
```

UNIX:

```
Inst_2_INSTALLATION_PATH/bin/setmqinst -i -n Inst_2
```

6. Down load fix pack 7.1.0.2, when it is released.

See Fix Central.

7. Apply fix pack 7.1.0.2 to `Inst_1`.

Follow the task for your platform in “Applying and removing maintenance level updates on UNIX, Linux, and Windows” on page 141.

8. Verify `Inst_1`.
9. Transfer queue managers to `Inst_1` one at a time.

a) Follow the procedure in step “4” on page 184

Replacing `Inst_2` by `Inst_1` in the instructions.

10. Set `Inst_1` primary.

Windows:

```
"Inst_1_INSTALLATION_PATH\bin\setmqinst" -i -n Inst_1
```

UNIX:

```
Inst_1_INSTALLATION_PATH/bin/setmqinst -i -n Inst_1
```

11. Repeat steps “1” on page 184 to “5” on page 185 for 7.1.0.3, 7.1.0.5, and so on
12. Repeat steps “6” on page 185 to “10” on page 185 for 7.1.0.4, 7.1.0.6, and so on.

Related concepts

“Queue manager coexistence in Version 7.5” on page 28
Queue managers, with different names, can coexist on any server as long as they use the same IBM WebSphere MQ installation. On UNIX, Linux, and Windows, different queue managers can coexist on the same server and be associated with different installations.

“Multi-installation queue manager coexistence on UNIX, Linux, and Windows” on page 31
You can install multiple copies of IBM WebSphere MQ for UNIX, Linux, and Windows on the same server. The installations must be at Version 7.1 or later, with one exception. One Version 7.0.1 installation, at fix pack level 6, or later, can coexist with multiple Version 7.1, or later installations.

Related tasks

“Migrating IBM WebSphere MQ library loading from Version 7.0.1 to Version 7.5” on page 108
No change in the way IBM WebSphere MQ libraries are loaded is normally required if you upgrade from Version 7.0.1 to Version 7.5. You must have followed the instructions on building IBM WebSphere MQ applications in Version 7.0.1 and you must replace IBM WebSphere MQ Version 7.0.1 with IBM WebSphere MQ Version 7.5. If you choose to take advantage of multi-installation in IBM WebSphere MQ Version 7.5, based on the side-by-side or multi-stage migration scenarios, you must modify the environment for the operating system to resolve IBM WebSphere MQ dependencies for an application. Typically, you can modify the runtime environment, rather than relink the application.

Related reference

setmqenv
setmqinst
setmqm

Related information

Installing WebSphere MQ server on Windows
Associating a queue manager with an installation
Changing the primary installation

Migration commands, utilities, and reference information

A selection of commands, utilities, and application reference information related to migration are collected together in the following subtopics.

JMS PROVIDERVERSION property

The JMS **PROVIDERVERSION** property selects whether a Java application publishes and subscribes using the queued command message interface, or the integrated call interface.

Rules for selecting the IBM WebSphere MQ messaging provider mode

The IBM WebSphere MQ messaging provider has two modes of operation: IBM WebSphere MQ messaging provider normal mode, and IBM WebSphere MQ messaging provider migration mode. You can select which mode a JMS application uses to publish and subscribe.

The IBM WebSphere MQ messaging provider normal mode uses all the features of a MQ queue manager to implement JMS. This mode is used only to connect to a IBM WebSphere MQ queue manager and can

connect to queue managers in either client or bindings mode. This mode is optimized to use the new function.

If you are not using IBM 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 that you are using, you can use the `com.ibm.msg.client.wmq.overrideProviderVersion` property to override 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 the possible values: 7, 6, or *unspecified*. However, **PROVIDERVERSION** can be a string in any one of the following formats:

- V.R.M.F
- V.R.M
- V.R
- V

where V, R, M and F are integer values greater than or equal to zero.

7 - Normal mode
Uses the IBM WebSphere MQ messaging provider normal mode.

If you set **PROVIDERVERSION** to 8 only the IBM WebSphere MQ messaging provider normal mode of operation is available. If the queue manager specified in the connection factory settings is not a Version 7.0.1, or later, queue manager, the `createConnection` method fails with an exception.

The IBM 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 IBM WebSphere MQ messaging provider normal mode and the `createConnection` method fails with an exception.

6 - Migration mode
Uses the IBM WebSphere MQ messaging provider migration mode.

The IBM WebSphere MQ classes for JMS use the features and algorithms supplied with IBM WebSphere MQ version 6.0. If you want to connect to WebSphere Event Broker version 6.0 or WebSphere Message Broker version 6.0 or 6.1 using IBM WebSphere MQ Enterprise Transport version 6.0, you must use this mode. You can connect to an IBM WebSphere MQ version 7.0.1 queue manager using this mode, but none of the new features of a version 7.0.1 queue manager are used, for example, read ahead or streaming. If you have an IBM WebSphere MQ version 7.0.1 client connecting to an IBM WebSphere MQ version 7.0.1 queue manager, then the message selection is done by the queue manager rather than on the client system.

unspecified
The **PROVIDERVERSION** property is set to *unspecified* by default.

A connection factory that was created with a previous version of IBM WebSphere MQ classes for JMS in JNDI takes this value when the connection factory is used with the new version of IBM 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 IBM WebSphere MQ messaging provider normal mode or IBM WebSphere MQ messaging provider migration mode is required.

1. First, an attempt to use IBM WebSphere MQ messaging provider normal mode is made.
2. If the queue manager connected is not IBM WebSphere MQ version 7.0.1, or later, the connection is closed and IBM WebSphere MQ messaging provider migration mode is used instead.
3. If the **SHARECNV** property on the server connection channel is set to 0, the connection is closed and IBM WebSphere MQ messaging provider migration mode is used instead.
4. If **BROKERVER** is set to V1 or the new default *unspecified* value, IBM WebSphere MQ messaging provider normal mode continues to be used, and therefore any publish/subscribe operations use the new IBM WebSphere MQ version 7.0.1, or later, features.

See ALTER QMGR for information about the PSMODE parameter of the ALTER QMGR command for further information on compatibility.

5. If **BROKERVER** is set to V2 the action taken depends on the value of **BROKERQMGR** :
   - If the **BROKERQMGR** is `blank`:

     If the queue specified by the **BROKERCONQ** property can be opened for output (that is, `MQOPEN` for output succeeds) and **PSMODE** on the queue manager is set to `COMPAT` or `DISABLED`, then IBM WebSphere MQ messaging provider migration mode is used.
   - If the queue specified by the **BROKERCONQ** property cannot be opened for output, or the **PSMODE** attribute is set to `ENABLED`:

     IBM WebSphere MQ messaging provider normal mode is used.
   - If **BROKERQMGR** is `non-blank` :

     IBM WebSphere MQ messaging provider mode is used.

Related concepts

When to use PROVIDERVERSION

Related reference

BROKERQMGR

BROKERCONQ

PSMODE

When to use PROVIDERVERSION

In two cases you must override the default selection of **PROVIDERVERSION** for the WebSphere MQ classes for JMS to work correctly.

There are two scenarios where you cannot use the algorithm described in “Rules for selecting the IBM WebSphere MQ messaging provider mode” on page 186; 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, WebSphere Application Server Version 6.0.2, or WebSphere Application Server Version 6.1, connection factories are defined 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.

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 has an existing queue manager that 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 ever done publish/subscribe, and has a connection factory with **BROKERVER** set to 2 (if the connection factory was created in WebSphere Application Server this is the default), 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 1 or unspecified. This is dependent on your application.
- Set **PROVIDERVERSION** to 7, which 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 product documentation.

Alternatively, use the client configuration property, or modify the queue manager connected so it does not have the **BROKERCONQ**, or make the queue unusable.

strmqbrk: Migrate the IBM WebSphere MQ Version 6.0 publish/subscribe broker to Version 7.5

Migrate the persistent state of a IBM WebSphere MQ publish/subscribe broker to a Version 7.5 queue manager.

Purpose

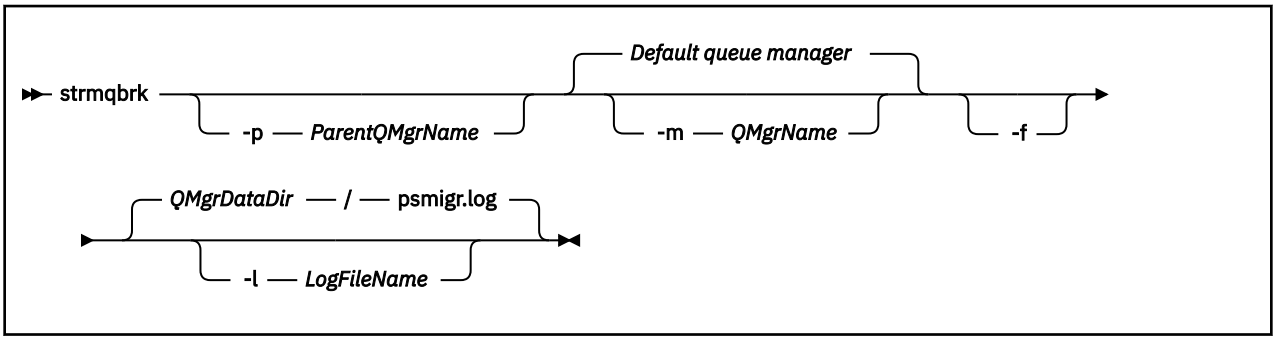
Use the strmqbrk command to migrate the state of a IBM WebSphere MQ Version 6.0 publish/subscribe broker to IBM WebSphere MQ Version 7.5 queue manager. If the queue manager has already been migrated, no action is taken.

In IBM WebSphere MQ Version 6.0, strmqbrk started a broker. IBM WebSphere MQ Version 7.5 publish/subscribe cannot be started in this manner. To enable publish/subscribe for a queue manager, use the **ALTER QMGR** command.

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



Optional parameters

AIX, HP-UX, Linux, Solaris, and Windows

-p ***ParentQMgrName***

Note: This option is deprecated. **strmqbrk** migrates the parent connection automatically.

If you specify the current parent queue manager, a warning message is issued and migration continues. If you specify a different queue manager, a error is issued and migration is not performed.

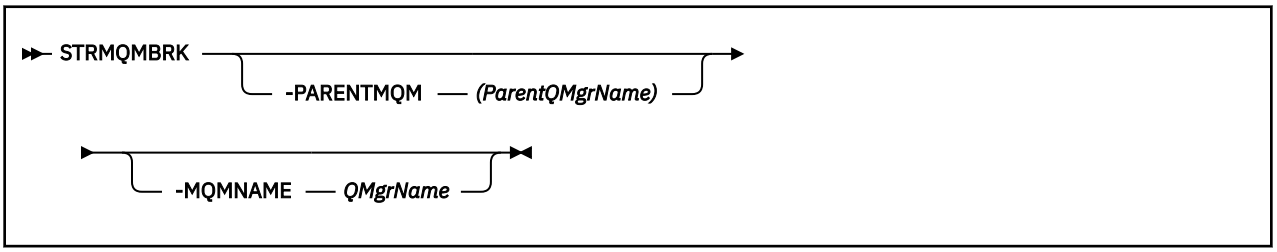
-m ***QMgrName***
The name of the queue manager to be migrated. If you do not specify this parameter, the command is routed to the default queue manager.

-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

IBM i



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 manager, 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.

Related reference
ALTER QMGR

migmbbrk

The migmbbrk command migrates publish/subscribe configuration data from WebSphere Event Broker Version 6.0 or WebSphere Message Broker Version 6.0 or 6.1 to WebSphere MQ Version 7.0.1 or later versions.

Purpose

The migmbbrk command is not supported on all of the platforms that WebSphere MQ supports. See *Supported operating systems* for details.

To use the **migmbbrk** command you must be using at least WebSphere Message Broker Version 6.0, Fix Pack 9, or WebSphere Message Broker Version 6.1, Fix Pack 4.

Use the **migmbbrk** command to migrate the publish/subscribe configuration data from a WebSphere Event Broker Version 6.0 or a WebSphere Message Broker Version 6.0 or Version 6.1 broker to a WebSphere MQ Version 7.0.1 or later queue manager. The command runs a migration process that migrates the following publish/subscribe configuration data to the queue manager that is associated with the named broker:

- Subscriptions
- Subscription points. (Subscription points are supported only when RFH2 messages are used.)
- Streams
- Retained publications

The **migmbbrk** command does not migrate the Access Control List (ACL). Instead, running the migration with the -t or -r parameters produces a file containing suggested setmqaut commands to set up a security environment in the queue manager that is equivalent to the security environment that existed in the broker. You must review and modify the security command file as needed and run the commands to set up a security environment in the queue manager, equivalent to the one that existed in the broker, before you run the migration with the -c parameter to complete the migration.

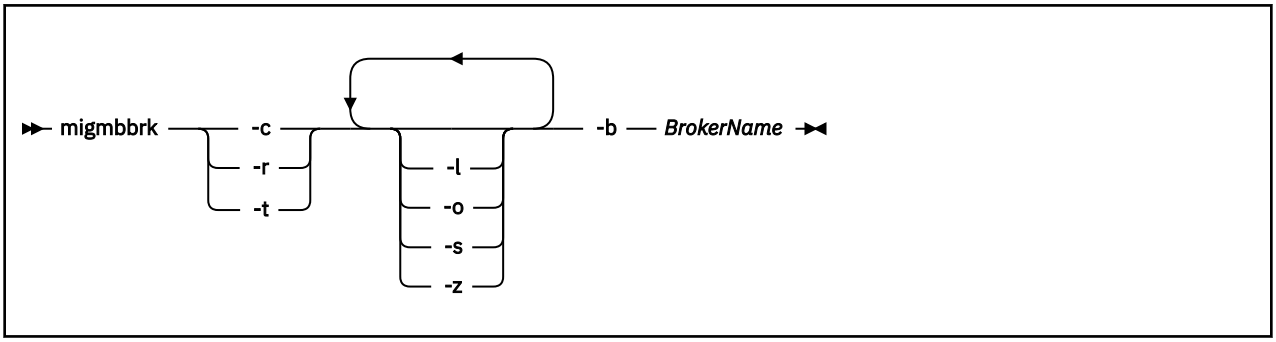
Note: On UNIX systems, all authorities are held by user groups internally, not by principals. This has the following implications:

- If you use the **setmqaut** command to grant an authority to a principal, the authority is granted to the primary user group of the principal. This means that the authority is effectively granted to all members of that user group.
- If you use the **setmqaut** command to revoke an authority from a principal, the authority is revoked from the primary user group of the principal. This means that the authority is effectively revoked from all members of that user group.

You must issue the **migmbbrk** command from a command window that can execute both WebSphere MQ and WebSphere Message Broker commands successfully. Typically this is true if the command is issued from a WebSphere Message Broker command console.

The WebSphere Event Broker Version 6.0 or WebSphere Message Broker Version 6.0 or 6.1 publish/subscribe configuration data, which is stored in the subscription database tables, is not deleted by the migration process. This configuration data is therefore available to use until you explicitly delete it.

Syntax



Required parameters

-b *BrokerName*
The name of the broker that is the source of the publish/subscribe configuration data that is to be migrated. The queue manager to which the publish/subscribe configuration data is migrated is the queue manager that is associated with the named broker.

-c
Complete the migration of the publish/subscribe configuration data. The completion phase of the migration uses the topic objects that are created in the initial -t phase. It is possible that the broker state has changed since the initial phase was run and that new additional topic objects are now required. If so, the completion phase creates new topic objects as necessary. The completion phase does not delete any topic objects that have become unnecessary; you might need to delete any topic objects that you do not require.

Before you complete the migration you must review and modify the security command file produced in the -r or -t phase as required and execute the commands to set up a security environment in the queue manager, equivalent to the one that existed in the broker.

Before you run this completion phase, you must run the initial -t phase. You cannot use the -c parameter with the -r parameter or the -t parameter. This phase also creates a migration log.

-r
Rehearse the migration process but do not change anything. You can use this parameter before running the migration with the -t parameter, to create a migration log, including any errors, so that you can observe what the result of the migration process would be, but without changing the current configurations.

Rehearsing the migration also produces a file containing suggested setmqaut commands to set up a security environment in the queue manager that is equivalent to the security environment that existed

in the broker. Before you complete the migration with the -c parameter you must review and modify the security command file as required and execute the commands to set up a security environment in the queue manager, equivalent to the one that existed in the broker.

You cannot use the -r parameter with the -c parameter or the -t parameter.

-t

Create topic objects that might be needed in the queue manager, based on the ACL entries that are defined in the broker.

Use of the -t parameter also produces a file containing suggested setmqaut commands to set up a security environment in the queue manager that is equivalent to the security environment that existed in the broker. The topic objects are created in anticipation of you executing the security commands to create ACLs for the topic objects. Before you complete the migration with the -c parameter you must review and modify the security command file as required and execute the commands to set up a security environment in the queue manager, equivalent to the one that existed in the broker.

You must run this phase before you run the completion phase with the -c parameter. You cannot use the -t parameter with the -c parameter or the -r parameter. This phase also creates a migration log.

Optional parameters

-l

Leave the broker running. If you do not specify this parameter, the broker is shut down by default at the end of the migration process.

-o

Overwrite any subscription or retained publication that exists in the queue manager and that has the same name as a subscription or retained publication that is being migrated from the broker, with the publish/subscribe configuration data that was retrieved from the broker. The -o parameter has no effect if you use it with the -r parameter.

-s

Discard any intermediate configuration data that was retained from a previous instance of the migration process that failed or was interrupted. The migration process populates private queues with temporary data. If the migration process completes successfully, the temporary data is deleted. If you do not specify this parameter and the migration process fails or is interrupted, the temporary data is retained and is used by the migration process if you restart it, so that the process resumes at the point where it previously failed or was interrupted.

-z

Run the migration process, regardless of whether it has previously run to a successful completion. If you do not specify this parameter and the migration process has previously run to a successful completion, the process recognizes this fact and exits. You can use the -o parameter with the -z parameter, but this is not mandatory. A previous rehearsal of the migration using the -r parameter does not count as a successful completion.

Return codes

Return Code	Explanation
0	Migration completed successfully
20	An error occurred during processing

Output files

The migration process writes two output files to the current directory:

amqmigrateacl.txt

A file containing a list of setmqaut commands, created in the current directory for you to review, change, and run if appropriate, to help you to reproduce your ACLs.

amqmigmbbrk.log
A log file containing a record of the details of the migration.

Examples

This command migrates the publish/subscribe configuration data of broker BRK1 into its associated queue manager and specifies that the migration process runs regardless of whether it has previously run to a successful completion. It also specifies that any subscription or retained publication that exists in the queue manager, that has the same name as a subscription or retained publication that is being migrated from the broker, must be overwritten.

```
migmbbrk -z -o -b BRK1
```

Supported operating systems

The **migmbbrk** command is supported only on the following platforms that support WebSphere Event Broker Version 6.0 or WebSphere Message Broker Version 6.0:

Microsoft Windows XP Professional with SP2, 32-bit versions only
Solaris x86-64 platform: Solaris 10
Solaris SPARC platform: Sun Solaris 9 (64-bit)
AIX Version 5.2 or later, 64-bit only
HP-UX Itanium platform: HP-UX 11i
Linux zSeries (64-bit)
Linux PowerPC® (64-bit)
Linux Intel x86
Linux Intel x86-64

On z/OS, the equivalent function to the migmbbrk command is provided by the CSQUMGMB utility.

Publish/subscribe migration log file

The publish/subscribe migration log file is called `amqmigmbbrk.log`. It shows the details of the migration.

Identifying subscriptions in the log file or error messages

The log file identifies a subscription in the log file or an error message by listing its properties, including its topic string. The topic string can be up to 10,240 characters long, and is expressed in Unicode in the broker. The topic string might be too long or contain characters that cannot be represented in the current character set.

To identify subscriptions precisely, the migration process assigns a sequence number to every broker subscription that it encounters. The sequence starts at 1. When a subscription is identified by the migration process, the identification includes the sequence number, and anything else that is known about the subscription.

You can use the **mqsireportproperties** broker command to list subscriptions in the broker in the same order as the migration process. For example, the fourth subscription that the migration process counts is also the fourth that is listed by **mqsireportproperties**. With this information, you can correlate the sequence number from the log file or error message with the subscription in the broker.

The broker command to list the subscriptions is as follows:

```
mqsireportproperties brokername -e default -o DynamicSubscriptionEngine -r
```

To store the results in a file called, for example, `outfile.txt`, the command is as follows:

```
mqsireportproperties brokername -e default -o DynamicSubscriptionEngine -r > outfile.txt
```

Example

The following log file lists the actions that have been performed by a successful migration process. It shows the migration of subscriptions, ACLs, and retained publications.

```
2009-01-28 11:43:54.187: Migrating Pub/Sub state from Websphere Message Broker: TEST_BROKER
2009-01-28 11:43:54.187: Into queue manager: TEST_QM
2009-01-28 11:43:54.187: Command switches:
2009-01-28 11:43:54.187:     -t
2009-01-28 11:43:54.187:     -z
2009-01-28 11:43:54.187:     -l
2009-01-28 11:43:55.484: Starting to parse subscriptions ...
2009-01-28 11:43:55.484: Migrating subscriptions for topic string RFH2/EU/France/Toison
2009-01-28 11:43:55.484: [1] Migrating subscription for:
2009-01-28 11:43:55.484:     Format: mqrfh2
2009-01-28 11:43:55.484:     Queue Manager: PSMIG_QM
2009-01-28 11:43:55.484:     Queue: PUBSUB.FRANCE.QUEUE
2009-01-28 11:46:23.968: Migrating subscriptions for topic string RFH2/EU/France/Carnac
2009-01-28 11:46:23.968: [2] Migrating subscription for:
2009-01-28 11:46:23.968:     Format: mqrfh2
2009-01-28 11:46:23.968:     Queue Manager: PSMIG_QM
2009-01-28 11:46:23.968:     Queue: PUBSUB.FRANCE.QUEUE
2009-01-28 11:46:23.968: Migrating subscriptions for topic string $SYS/STREAM/TEST_STREAM/RFH1/EU/France/
Pontivy
2009-01-28 11:46:23.984: [3] Migrating subscription for:
2009-01-28 11:46:23.984:     Format: mqrfh2
2009-01-28 11:46:23.984:     Queue Manager: PSMIG_QM
2009-01-28 11:46:23.984:     Queue: PUBSUB.FRANCE.QUEUE
2009-01-28 11:46:24.031: Migrating subscriptions for topic string $SYS/Broker/+/warning/expiry/
Subscription/#
2009-01-28 11:46:24.031: [4] Migrating subscription for:
2009-01-28 11:46:24.031:     Format: mqrfh2
2009-01-28 11:46:24.031:     Queue Manager: PSMIG_QM
2009-01-28 11:46:24.031:     Queue: PUBSUB.SAMPLE.QUEUE
2009-01-28 11:46:24.125: Migrating subscriptions for topic string $SYS/Broker/+/Subscription/#
2009-01-28 11:46:24.125: [5] Migrating subscription for:
2009-01-28 11:46:24.125:     Format: mqrfh2
2009-01-28 11:46:24.125:     Queue Manager: PSMIG_QM
2009-01-28 11:46:24.125:     Queue: PUBSUB.SAMPLE.QUEUE
2009-01-28 11:46:24.140: Migrating subscriptions for topic string $SYS/Broker/+/Status
2009-01-28 11:46:24.140: [6] Migrating subscription for:
2009-01-28 11:46:24.140:     Format: mqrfh2
2009-01-28 11:46:24.140:     Queue Manager: PSMIG_QM
2009-01-28 11:46:24.140:     Queue: PUBSUB.SAMPLE.QUEUE
2009-01-28 11:46:24.156: Migrating subscriptions for topic string $SYS/Broker/+/Status/ExecutionGroup/#
2009-01-28 11:46:24.156: [7] Migrating subscription for:
2009-01-28 11:46:24.156:     Format: mqrfh2
2009-01-28 11:46:24.156:     Queue Manager: PSMIG_QM
2009-01-28 11:46:24.156:     Queue: PUBSUB.SAMPLE.QUEUE
2009-01-28 11:46:24.250: Migrating subscriptions for topic string $SYS/STREAM/TEST_STREAM/RFH1/EU/France/
Kersaux
2009-01-28 11:46:24.250: [8] Migrating subscription for:
2009-01-28 11:46:24.250:     Format: mqrfh2
2009-01-28 11:46:24.250:     Queue Manager: PSMIG_QM
2009-01-28 11:46:24.250:     Queue: PUBSUB.FRANCE.QUEUE
2009-01-28 11:46:24.281: ... finished parsing subscriptions
2009-01-28 11:46:24.281: Starting to parse topics ...
2009-01-28 11:46:24.281: Migrating ACLs for topic string
2009-01-28 11:46:24.281: Migrating ACLs for topic string RFH2/EU/France/Toison
2009-01-28 11:46:24.281: Migrating ACLs for topic string RFH2/EU/France/Carnac
2009-01-28 11:46:24.281: Migrating ACLs for topic string $SYS/STREAM/TEST_STREAM/RFH1/EU/France/Pontivy
2009-01-28 11:46:24.281: Migrating ACLs for topic string $SYS/Broker/+/warning/expiry/Subscription/#
2009-01-28 11:46:24.281:        None found.
2009-01-28 11:46:24.281: Migrating ACLs for topic string $SYS/Broker/+/Subscription/#
2009-01-28 11:46:24.281:        None found.
2009-01-28 11:46:24.281: Migrating ACLs for topic string $SYS/Broker/+/Status
2009-01-28 11:46:24.281:        None found.
2009-01-28 11:46:24.281: Migrating ACLs for topic string $SYS/Broker/+/Status/ExecutionGroup/#
2009-01-28 11:46:24.281:        None found.
2009-01-28 11:46:24.281: Migrating ACLs for topic string $SYS/STREAM/TEST_STREAM/RFH1/EU/France/Kersaux
2009-01-28 11:46:24.281: ... finished parsing topics
2009-01-28 11:46:24.281: Starting to parse retained publications ...
2009-01-28 11:46:24.296: Migrating retained publications for topic string $SYS/Broker/TEST_BROKER/Status
2009-01-28 11:46:24.296: Migrating retained publication for default subscription point.
2009-01-28 11:46:24.906: ... finished parsing retained publicatons
2009-01-28 11:46:24.968:
All Pub/Sub data has been retrieved from the broker.
2009-01-28 11:46:24.968: Applying changes to queue manager Pub/Sub state.
2009-01-28 11:46:24.972: Created topic object: MIGMBBRK.TOPIC.00004
```

```
2009-01-28 11:46:24.972: Created topic object: MIGMBBRK.TOPIC.00003
2009-01-28 11:46:24.972: Created topic object: MIGMBBRK.TOPIC.00002
2009-01-28 11:46:24.972: Created topic object: MIGMBBRK.TOPIC.00001
2009-01-28 11:46:24.977: Defining subscription [1]
2009-01-28 11:46:24.977: Defining subscription [2]
2009-01-28 11:46:24.977: Defining subscription [3]
2009-01-28 11:46:24.977: Defining subscription [4]
2009-01-28 11:46:24.993: Defining subscription [5]
2009-01-28 11:46:24.993: Defining subscription [6]
2009-01-28 11:46:24.993: Defining subscription [7]
2009-01-28 11:46:24.993: Defining subscription [8]
2009-01-28 11:46:29.996: Migration completion message written.
```

PROPCTL channel options

Use **PROPCTL** channel attribute to control which message properties are included in a message that is sent from a Version 7.5 queue manager to a queue manager from an earlier version of WebSphere MQ.

Table 15. Channel message property attribute settings

PROPCTL	Description
ALL	Use this option if applications connected to the Version 6.0 queue manager are able to process any properties placed in a message by a Version 7.5 application. All properties are sent to the Version 6.0 queue manager, in addition to any name/value pairs placed in the MQRFH2. You must consider two application design issues: 1. An application connected to the Version 6.0 queue manager must be able to process messages containing MQRFH2 headers generated on a Version 7.5 queue manager. 2. The application connected to the Version 6.0 queue manager must process new message properties that are flagged with MQPD_SUPPORT_REQUIRED correctly. With the ALL channel option set, JMS applications can interoperate between WebSphere MQ Version 6.0 and Version 7.5 using the channel. New Version 7.5 applications using message properties can interoperate with Version 6.0 applications, depending on how the Version 6.0 application handles MQRFH2 headers.

Table 15. Channel message property attribute settings (continued)

PROPCTL	Description
COMPAT	Use this option to send message properties to applications connected to a Version 6.0 queue manager in some cases, but not all. Message properties are only sent if two conditions are met: 1. No property must be marked as requiring message property processing. 2. At least one of the message properties must be in a "reserved" folder; see Note. With the COMPAT channel option set, JMS applications can interoperate between WebSphere MQ Version 6.0 and Version 7.5 using the channel. The channel is not available to every application using message properties, only to those applications that use the reserved folders. The rules concerning whether the message or the property is sent are: 1. If the message has properties, but none of the properties are associated with a "reserved" folder, then no message properties are sent. 2. If any message property has been created in a "reserved" property folder, all message properties associated with the message are sent. However: a. If any of the message properties are marked as support being required, MQPD_SUPPORT_REQUIRED or MQPD_SUPPORT_REQUIRED_IF_LOCAL, the whole message is rejected. It is returned, discarded, or sent to the dead letter queue according to the value of its report options. b. If no message properties are marked as support being required, an individual property might not be sent. If any of the message property descriptor fields are set to non-default values the individual property is not sent. The message is still sent. An example of a non-default property descriptor field value is MQPD_USER_CONTEXT. **Note:** The "reserved" folders names start with mcd., jms., usr., or mqext.. These folders are created for applications that use the JMS interface. In Version 7.5 any name/value pairs that are placed in these folders are treated as message properties. Message properties are sent in an MQRFH2 header, in addition to any name/value pairs placed in an MQRFH2 header. Any name/value pairs placed in an MQRFH2 header are sent as long as the message is not rejected.
NONE	Use this option to prevent any message properties being sent to applications connected to a Version 6.0 queue manager. An MQRFH2 that contains name/value pairs and message properties is still sent, but only with the name/value pairs. With the NONE channel option set, a JMS message is sent as a JMSTextMessage or a JMSBytesMessage without any JMS message properties. If it is possible for a Version 6.0 application to ignore all properties set in a Version 7.5 application, it can interoperate with it.

PROPCTL queue options

Use the **PROPCTL** queue attribute to control how message properties are returned to an application that calls MQGET without setting any MQGMO message property options.

Table 16. Queue message property attribute settings

PROPCTL	Description
ALL	Use this option so that different applications reading a message from the same queue can process the message in different ways. • An application, migrated unchanged from Version 6.0, can continue to read the MQRFH2 directly. Properties are directly accessible in the MQRFH2 header. You must modify the application to handle any new properties, and new property attributes. It is possible that the application might be affected by changes in the layout and number of MQRFH2 headers. Some folder attributes might be removed, or that WebSphere MQ reports an error in the layout of the MQRFH2 header that it ignored in Version 6.0. • A new or changed application can use the message property MQI to query message properties, and read name/value pairs in MQRFH2 header directly. All the properties in the message are returned to the application. • If the application calls MQCRTMH to create a message handle, it must query the message properties using MQINQMP. Name/value pairs that are not message properties remain in the MQRFH2, which is stripped of any message properties. • If the application does not create a message handle, all the message properties and name/value pairs remain in the MQRFH2. ALL only has this affect if the receiving application has not set a MQGMO_PROPERTIES option, or has set it to MQGMO_PROPERTIES_AS_Q_DEF.

Table 16. Queue message property attribute settings (continued)

PROPCTL	Description
COMPAT	COMPAT is the default option. If GMO_PROPERTIES_* is not set, as in an unmodified Version 6.0 application, COMPAT is assumed. By defaulting to the COMPAT option, a Version 6.0 application that did not explicitly create an MQRFH2, works without change on Version 7.5. Use this option if you have written an Version 6.0 MQI application to read JMS messages. • The JMS properties, which are stored in an MQRFH2 header, are returned to the application in an MQRFH2 header in folders with names starting with mcd., jms., usr., or mqext.. • If the message has JMS folders, and if a Version 7.5 application adds new property folders to the message, these properties are also returned in the MQRFH2. Consequently, you must modify the application to handle any new properties, and new property attributes. It is possible that an unmodified application might be affected by changes in the layout and number of MQRFH2 headers. It might find some folder attributes are removed, or that WebSphere MQ finds errors in the layout of the MQRFH2 header that it ignored in Version 6.0. **Note:** In this scenario, the behavior of the application is the same whether it is connected to a Version 6.0 or Version 7.5 queue manager. If the channel **PROPCTL** attribute is set to COMPAT or ALL any new message properties are sent in the message to the Version 6.0 queue manager. • If the message is not a JMS message, but contains other properties, those properties are not returned to the application in an MQRFH2 header.[11] • The option also enables Version 6.0 applications that explicitly create an MQRFH2 to work correctly, in many cases. For example, An MQI program that creates an MQRFH2 containing JMS message properties continues to work correctly. If a message is created without JMS message properties, but with some other MQRFH2 folders, the folders are returned to the application. Only if the folders are message property folders are those specific folders are removed from the MQRFH2. Message property folders are identified by having the new folder attribute content='properties', or are folders with names listed in Defined property folder name or Ungrouped property folder name. • If the application calls MQCRTMH to create a message handle, it must query the message properties using MQINQMP. Message properties are removed from the MQRFH2 headers. Name/value pairs that are not message properties remain in the MQRFH2. • If the application calls MQCRTMH to create a message handle, it can query all message properties, regardless of whether the message has JMS folders. • If the application does not create a message handle, all the message properties and name/value pairs remain in the MQRFH2. If a message contains new user property folders, you can infer that the message was created by a new or changed Version 7.5 application. If the receiving application is to process these new properties directly in an MQRFH2, you must modify the application to use the ALL option. With the default COMPAT option set, an unmodified application continues to process the rest of the MQRFH2, without the Version 7.5 properties. The intent of the PROPCTL interface is to support old applications reading MQRFH2 folders, and new and changed applications using the message property interface. Aim for new applications to use the message property interface for all user message properties, and to avoid reading and writing MQRFH2 headers directly. COMPAT only has this affect if the receiving application has not set a MQGMO_PROPERTIES option, or has set it to MQGMO_PROPERTIES_AS_Q_DEF.

Table 16. Queue message property attribute settings (continued)	
PROPCTL	**Description**
FORCE	The FORCE option places all messages properties into MQRFH2 headers. All message properties and name/value pairs in the MQRFH2 headers remain in the message. Message properties are not removed from the MQRFH2, and made available through a message handle. The effect of choosing the FORCE option is to enable a newly migrated application to read message properties from MQRFH2 headers. Suppose you have modified an application to process Version 7.5 message properties, but have also retained its ability to work directly with MQRFH2 headers, as before. You can decide when to switch the application over to using message properties by initially setting the PROPCTL queue attribute to FORCE. Set the **PROPCTL** queue attribute to another value when you are ready to start using message properties. If the new function in the application does not behave as you expected, set the **PROPCTL** option back to FORCE. FORCE only has this affect if the receiving application has not set a MQGMO_PROPERTIES option, or has set it to MQGMO_PROPERTIES_AS_Q_DEF.
NONE	Use this option so that an existing application can process a message, ignoring all message properties, and a new or changed application can query message properties. • If the application calls MQCRTMH to create a message handle, it must query the message properties using MQINQMP. Name/value pairs that are not message properties remain in the MQRFH2, which is stripped of any message properties. • If the application does not create a message handle, all the message properties are removed from the MQRFH2. Name/value pairs in the MQRFH2 headers remain in the message. NONE only has this affect if the receiving application has not set a MQGMO_PROPERTIES option, or has set it to MQGMO_PROPERTIES_AS_Q_DEF.
V6COMPAT	Use this option to receive an MQRFH2 in the same format as it was sent. If the sending application, or the queue manager, creates additional message properties, these are returned in the message handle. This option has to be set on both the sending and receiving queues, and any intervening transmission queues. It overrides any other PROPCTL options set on queue definitions in the queue name resolution path. Use this option only in exceptional circumstances. For example, if you are migrating applications from Version 6.0 to Version 7.5 the option is valuable because it preserves Version 6.0 behavior. The option is likely to have an impact on message throughput. It is also more difficult to administer; you need to ensure the option is set on the sender, receiver, and intervening transmission queues. V6COMPAT only has this affect if the receiving application has not set a MQGMO_PROPERTIES option, or has set it to MQGMO_PROPERTIES_AS_Q_DEF.

Related reference
PROPCTL

[11] The existence of specific property folders created by the WebSphere MQ classes for JMS indicates a JMS message. The property folders are `mcd.`, `jms.`, `usr.`, or `mqext.`

MQGMO message property option settings

Use MQGMO message property options to control how message properties are returned to an application.

Table 17. MQGMO message property option settings

MQGMO Option	Description
MQGMO_PROPERTIES_AS_Q_DEF	Version 6.0 and Version 7.5 applications that read from the same queue, and do not set GMO_PROPERTIES_*, receive the message properties differently. Version 6.0 applications, and Version 7.5 applications that do not create a message handle, are controlled by the queue **PROPCTL** attribute. A Version 7.5 application can choose to receive message properties in the MQRFH2, or create a message handle and query the message properties. If the application creates a message handle, properties are removed from the MQRFH2. • An unmodified Version 6.0 application does not set GMO_PROPERTIES_*. Any message properties it receives are in the MQRFH2 headers. • A new or changed Version 7.5 application that does not set GMO_PROPERTIES_* or sets it to MQGMO_PROPERTIES_AS_Q_DEF can choose to query message properties. It must set MQCRTMH to create a message handle and query message properties using the MQINQMP MQI call. • If a new or changed application does not create a message handle, it behaves like a Version 6.0 application. It must read any message properties it receives directly from the MQRFH2 headers. • If the queue attribute **PROPCTL** is set to FORCE, no properties are returned in the message handle. All properties are returned in MQRFH2 headers. • If the queue attribute **PROPCTL** is set to NONE, or COMPAT, a Version 7.5 application that creates a message handle, receives all message properties. • If the queue attribute **PROPCTL** is set to V6COMPAT, and also set to V6COMPAT on all the queues the message was placed on between the sender and the receiver, properties set by MQSETMP are returned in the message handle, and properties and name/value pairs created in an MQRFH2 are returned in the MQRFH2. The format of an MQRFH2 sent in Version 7.5 is the same as Version 6.0, when sent by the same application.
MQGMO_PROPERTIES_IN_HANDLE	Force an application to use message properties. Use this option to detect if a modified application fails to create message handle. The application might be trying to read message properties directly from an MQRFH2, rather than call MQINQMP.

Table 17. MQGMO message property option settings (continued)	
MQGMO Option	**Description**
MQGMO_NO_PROPERTIES	Version 6.0 and a Version 7.5 application behave the same way, even if the Version 7.5 application creates a message handle. • All properties are removed. An unchanged Version 6.0 application connected to a Version 7.5 queue manager might behave differently to when it was connected to a Version 6.0 queue manager. Queue manager generated properties, such as JMS properties, are removed. • Properties are removed even if a message handle is created. Name/value pairs in other MQRFH2 folders are available in the message data.
MQGMO_PROPERTIES_FORCE_MQRFH2	Version 6.0 and Version 7.5 applications behave the same way. Properties are returned in the MQRFH2 headers, even if a message handle is created. • MQINQMP returns no message properties, even if a message handle is created. MQRC_PROPERTY_NOT_AVAILABLE is returned if a property is inquired upon.
MQGMO_PROPERTIES_COMPATIBILITY	An Version 6.0 application connected to a Version 7.5 queue manager behaves the same as when it is connected to a Version 6.0 queue manager. If the message is from a JMS client, the JMS properties are returned in the MQRFH2 headers. New or modified Version 7.5 applications, that create a message handle, behave differently. • All properties in any message property folders are returned if the message contains a mcd., jms., usr., or mqext. folder. • If the message contains property folders, but not a mcd., jms., usr., or mqext. folder, no message properties are returned in an MQRFH2. • If a new or modified Version 7.5 application creates a message handle, query message properties using the MQINQMP MQI call. All message properties are removed from the MQRFH2. • If a new or modified Version 7.5 application creates a message handle, all properties in the message can be queried. Even if the message does not contain a mcd., jms., usr., or mqext. folder, all message properties are queriable.

Related reference

PROPCTL

2471 (09A7) (RC2471): MQRC_PROPERTY_NOT_AVAILABLE

Notices

This information was developed for products and services offered in the U.S.A.

IBM may not offer the products, services, or features discussed in this document 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 document. The furnishing of this document does not grant 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:

Intellectual Property Licensing
Legal and Intellectual Property Law
IBM Japan, Ltd.
19-21, Nihonbashi-Hakozakicho, Chuo-ku
Tokyo 103-8510, 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 publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication 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 Corporation
Software Interoperability Coordinator, Department 49XA
3605 Highway 52 N
Rochester, MN 55901
U.S.A.

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 Program License Agreement, or any equivalent agreement between us.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurements may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

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.

All statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

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.

If you are viewing this information softcopy, the photographs and color illustrations may not appear.

Programming interface information

Programming interface information, if provided, is intended to help you create application software for use with this program.

This book contains information on intended programming interfaces that allow the customer to write programs to obtain the services of IBM WebSphere MQ.

However, this information may also contain diagnosis, modification, and tuning information. Diagnosis, modification and tuning information is provided to help you debug your application software.

Important: Do not use this diagnosis, modification, and tuning information as a programming interface because it is subject to change.

Trademarks

IBM, the IBM logo, ibm.com®, are trademarks of IBM Corporation, registered in many jurisdictions worldwide. A current list of IBM trademarks is available on the Web at "Copyright and trademark information"www.ibm.com/legal/copytrade.shtml. Other product and service names might be trademarks of IBM or other companies.

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

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

Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both.

This product includes software developed by the Eclipse Project (http://www.eclipse.org/).

Java and all Java-based trademarks and logos are trademarks or registered trademarks of Oracle and/or its affiliates.

Part Number:

(1P) P/N: