9.1

Maintaining and Migrating IBM MQ

IBM

Note

Before using this information and the product it supports, read the information in "Notices" on page 261.

This edition applies to version 9 release 1 of IBM® MQ and to all subsequent releases and modifications until otherwise indicated in new editions.

Contents

Maintaining and migrating IBM MQ

Maintenance, upgrade, and migration have three distinct meanings for IBM MQ. The definitions are described here. The following sections describe the various concepts associated with migration, followed by the various tasks needed; these tasks are platform-specific where needed.

About this task

⚠ **Attention:** The information in this section applies to both Continuous Delivery (CD) and Long Term Support (LTS) releases.

Any information that applies specifically to an LTS or CD release is marked with the appropriate icon.

IBM MQ uses the terms *maintenance*, *upgrade* and *migration* as follows:

Maintenance is the application of a fix pack, cumulative security update (CSU), interim fix or Program Temporary Fix (PTF).

Maintenance has one main characteristic. Those fixes, whether they are applied by using a maintenance installation tool, or installed by 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. However, you should test applications with the new level of IBM MQ code.

For more information, see "Applying maintenance to IBM MQ" on page 8.

Upgrading is the process of taking an existing IBM MQ installation and upgrading to a new level of code.

Unless you are upgrading the fix level of IBM MQ, but not its command level, an upgrade must be followed by migration. 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 IBM MQ require queue manager migration before applications can reconnect.

For more information, see "Upgrading IBM MQ" on page 65.

Migration is the process of updating queue manager data to match a newer level of code.

Migration occurs the first time a queue manager is started with the newer level of code, and 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. Once migration has occurred, the queue manager can no longer be started by an earlier code level. On most platforms, queue manager migration is not reversible:

- Multi Migration cannot be reversed on IBM MQ for Multiplatforms. This restriction applies whether your enterprise uses the Long Term Support (LTS) release or Continuous Delivery (CD) release model.
- z/OS IBM MQ for z/OS® queue managers can only be backwards migrated from an LTS release to an earlier LTS release. For more information, see IBM MQ release types.

For more information, see "Migrating IBM MQ" on page 72.

Related concepts

"Characteristics of upgrades and fixes" on page 7
For IBM MQ, the term upgrade applies to changing the version V, release R, or modification M of a product. The term fix applies to a change in the F digit.

Related reference

z/OS IBM MQ for z/OS Program Directory PDF files

Where to find more information about maintaining and migrating

Where to look for more information, for example if you are getting started with migrating and maintaining IBM MQ.

Getting started with maintaining and migrating IBM MQ

If you are not familiar with IBM MQ migration, start by reading the "Migration concepts and methods" on page 76 section. Use these topics to find out more about the concepts that you must understand before planning migration tasks, including the difference between maintenance, migration, and upgrading and which migration paths are supported.

Linux AIX Windows For tutorials to help you with installing and upgrading, see A collection of tutorials for installing and upgrading IBM MQ on AIX®, Linux®, and Windows. The tutorials cover:

- Preparing a host for IBM MQ.
- Downloading the IBM MQ code.
- Installing and uninstalling the IBM MQ code, and applying fix packs.
- Upgrading from one version of IBM MQ to another, and moving a queue manager from one host to another.

Important: z/OS V 9.1.0 If you are migrating your IBM WebSphere® MQ 7.1 system, you must migrate your system to IBM MQ 8.0.0, or IBM MQ 9.0.0, before you migrate to the latest version. See the appropriate version of the product documentation for information on how to carry out the task.

New features and changes in this release

For information about new features and changes in this release, see the following information:

- V 9.1.0 What's new and changed in IBM MQ 9.1
- CD What's new and changed in IBM MQ 9.1.x Continuous Delivery
- LTS What's new and changed in IBM MQ 9.1.0 Long Term Support

New features and changes in earlier releases

Some new features and changes from earlier releases might have an impact on planning your migration because they affect the behavior of existing applications or the automation of management tasks. For information on where to find details of these changes in the product documentation for earlier releases, see What was new and changed in earlier releases.

System requirements and prerequisites

You can use the Software Product Compatibility Reports (SPCR) tool to find information on supported operating systems, system requirements, prerequisites, and optional supported software for IBM MQ. For more information about the SPCR tool and links to reports for each supported platform, see the System Requirements for IBM MQ web page.

For information about limitations and known problems for the current and earlier versions of IBM MQ, see the appropriate product readme file, which is available from the IBM MQ, WebSphere MQ, and MQSeries® product readmes web page.

Related concepts

IBM MQ in the IBM Documentation Offline app

IBM MQ 9.0 PDF files for product documentation and Program Directories

Characteristics of upgrades and fixes

For IBM MQ, the term upgrade applies to changing the version V, release R, or modification M of a product. The term fix applies to a change in the F digit.

Characteristics of fixes

The application of a fix pack, cumulative security update (CSU), or interim fix on Multiplatforms, or a program temporary fix (PTF) on z/OS is called a fix. You apply fixes by using a maintenance installation tool.

On the following platforms, fixes that are applied by using a maintenance installation tool can be rolled back completely if no queue manager migration has taken place:

- AIX AIX
- Windows Windows
- z/OS z/OS

and IBM MQ is returned to its previous code level.

Attention: z/OS CD On z/OS Continuous Delivery releases, certain PTFs will increase the modification level, and therefore, should be considered an upgrade.

On all other platforms you must reinstall the product.

Characteristics of different types of upgrade

An upgrade can take one of three different forms:

1. 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.
2. 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.
3. Side by side installation.

   - z/OS On z/OS you can install different code levels alongside each other on the same server. In the JCL to start a subsystem, you select the code level to use.
   - ULW On UNIX, Linux, and Windows, you associate a queue manager with an installation, and start the queue manager. In IBM MQ, running multiple queue managers at different command levels on the same server is termed queue manager coexistence.

   You must not infer from this, that 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.

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

z/OS On z/OS, reversibility of an 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. See "Upgrade and migration of IBM MQ on z/OS " on page 197 for more information.

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

On the following platforms, changes in version, release, or modification level are not fully reversible, but changes in fix level are reversible under certain conditions.

- UNIX UNIX
- Linux Linux
- Windows Windows
- IBM i IBM i

An irreversible upgrade implies that you must back up the queue managers, or your system, before upgrading, 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 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.

Related concepts

"Upgrade and migration of IBM MQ on z/OS " on page 197
You can install new releases of IBM MQ to upgrade IBM MQ to a new release, or version level. Multiple installations at the same or different levels can coexist on the same z/OS instance. Running a queue manager at a higher level requires migration.

Related tasks

"Applying maintenance to IBM MQ" on page 8
Maintenance is the application of a reversible fix. Any changes to queue manager data are compatible with the previous code level.

"Upgrading IBM MQ" on page 65
Upgrading is the process of taking an existing IBM MQ installation and upgrading to a new level of code.

"Migrating IBM MQ" on page 72
Migration is the conversion of programs and data to work with a new code level of IBM 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.

Related reference

"OPMODE on z/OS" on page 196
In earlier versions of IBM MQ for z/OS the availability of new functions and backward migration was controlled by the **OPMODE** parameter in the CSQ6SYSP macro. In IBM MQ 9.1.0 this no longer applies, and the **OPMODE** parameter is ignored.

Applying maintenance to IBM MQ

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

Before you begin

This task assumes that you understand the difference between Long Term Support and Continuous Delivery releases, and the maintenance delivery model that applies in each case. For more information, see IBM MQ Release Types and versioning.

About this task

About applying maintenance

Maintenance deliveries for a particular version/release are cumulative, from the initial release. You can apply any higher numbered fix pack or cumulative security update (CSU) of the same version/release to upgrade directly to that version level. You do not have to apply the intervening fixes.

You can also refresh the full version of IBM MQ by installing a manufacturing refresh that is available online or on physical media. The result of installing a manufacturing refresh is almost the same as applying a maintenance delivery to an earlier fix level of IBM MQ. There is one important difference: Fix packs and CSUs are applied using a maintenance procedure, manufacturing refreshes are installed using an installation procedure. You can "unapply" a fix pack or CSU to return to the previous fix level you had installed. You can only uninstall a manufacturing refresh, which removes IBM MQ from your system.

In addition to manufacturing refreshes and maintenance deliveries, you might occasionally be directed by the IBM Support team to apply an interim fix. Interim fixes are also known as emergency or test fixes, and are used to apply urgent updates that cannot wait for the next maintenance delivery. Interim fixes are known by a unique fix name, which will include the target version and platform, and other identifying context, such as a support case reference or APAR number. When you apply a new CSU, fix pack, or manufacturing refresh, all interim fixes are removed. The documentation with maintenance delivery or manufacturing refresh includes a "fix list" page, stating which APAR fixes are included in the deliverable. This list tells you if the APARs associated with the interim fixes that you have applied have been fixed in the latest maintenance. If they have not, check to see if there are new interim fixes, at the new level, for the APARs that concern you. If there are not, consult IBM Support. They might tell you to reapply the interim fix, or they might supply a new interim fix.

You get manufacturing refreshes, maintenance deliveries and interim fixes through Passport Advantage® and Fix Central. See Where to find downloadable installation images.

- Manufacturing refreshes are available through Passport Advantage.
- Fix packs and CSUs are available through Fix Central.
- Interim fixes are typically provided directly by the IBM Support team through a support case, and occasionally provided through Fix Central.

About removing maintenance

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 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.

On a smaller scale, a fix pack, CSU, or interim fix might introduce a new configuration parameter to solve a problem. If you remove the fix pack, CSU, or interim fix, although the new interface introduced by the change is not available any more, IBM 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.

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 updates to update an earlier release to the same maintenance level, have different results.

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

Maintenance levels and fix levels are both supplied from the service site, Fix Central. For information on where to find direct links to specific fix packs, CSUs, and other IBM MQ resources on Fix Central, see IBM MQ downloads.

Procedure

- To check the IBM MQ maintenance level:
  - Type the command **dspmqver**, or **DSPMQMVER** on IBM i. The returned messages include the three-digit VRM or, if maintenance has been applied, the four-digit VRMF.
  - Use the REST API GET method.
  - Linux Windows View the queue manager property panel in IBM MQ Explorer.
  - z/OS Look at the CSQY000I message in the queue manager job log. This message is issued when the queue manager starts, and shows the release level and release type.
- To apply or remove maintenance level updates, follow the appropriate links for the platforms that your enterprise uses.

Related concepts

"Multi-installation queue manager coexistence on UNIX, Linux, and Windows" on page 98
You can install multiple copies of IBM MQ for UNIX, Linux, and Windows on the same server. These IBM MQ copies can be at the same or different version levels. This is called a multi-installation. Multi-installation is particularly useful when you upgrade from one IBM MQ version to a later version, because it allows you to run the earlier version alongside the later version.

"Queue manager coexistence" on page 95
Queue managers, with different names, can coexist on any server as long as they use the same IBM MQ installation. On z/OS, UNIX, Linux, and Windows, different queue managers can coexist on the same server and be associated with different installations.

Related tasks

Backing up and restoring a queue manager

Linux UNIX Applying and removing maintenance on AIX

Maintenance tasks associated with AIX are grouped in this section.

Related tasks

"Applying maintenance to IBM MQ" on page 8
Maintenance is the application of a reversible fix. Any changes to queue manager data are compatible with the previous code level.

AIX Applying maintenance level updates on AIX

You apply maintenance level updates to IBM MQ for AIX by using **installp**.

Before you begin

1. Ensure that 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 MQ installations, you must identify the installation. Make sure that the commands you enter run against the correct installation; see **setmqenv**.

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.

Important: You cannot go back from a later version of the product to a prior version of the product, for example from IBM MQ 9.0 to IBM MQ 8.0.

You can apply and remove maintenance from an IBM 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 to 3 in the following maintenance procedure.

Major full versions of the base product are COMMITTED by default. Fix packs on a full base version can be in APPLIED state, and you can go back one release level.

If you need the ability to revert to an earlier version, you should perform a side-by-side migration, and migrate your queue managers to the later version at any time. See "Migrating on UNIX and Linux: side-by-side" on page 153 for further information.

However, if you start a queue manager under IBM MQ 8.0 or later, that queue manager is automatically migrated, and cannot be downgraded to the previous version.

Procedure

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

   If you use the 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. Stop the mqweb server that is associated with the IBM MQ installation:
   a) Check whether the mqweb server is running by entering the following command:

   ```
   dspmqweb status
   ```

   b) Stop the mqweb server by entering the following command:

   ```
   endmqweb
   ```
4. End all the activity of queue managers associated with the IBM 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
   ```

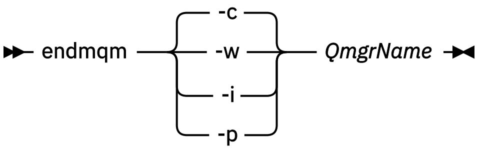
   dspmq -o installation -o status displays the installation name and status of queue managers associated with all installations of IBM MQ.

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

   b) Use the MQSC command **DISPLAY LSSTATUS** to list the status of listeners associated with a queue manager, as shown in the following example:

```
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 MQ libraries they have loaded. If they do not, you must find another way to force applications to release IBM 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 MQ. The application is not informed about queue managers in the current installation being shut down.

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

Note: "Applying maintenance level updates to multi-instance queue managers on Linux" on page 35 describes how to apply maintenance to a multi-instance queue manager. A multi-instance queue manager can continue to run on one server, while maintenance is applied to another server.

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

```
endmqlsr -m QMgrName
```

5. Log in as root, or switch to the superuser using the **su** command.
6. 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 MQ is installed underneath the directory. For example, if /USIL1 is specified, the IBM 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

AIX Reverting to the previous maintenance level on AIX

You can revert to a previous maintenance level by using the *System Management Interface Tool* (SMIT).

Before you begin

If you are running on a server with multiple IBM MQ installations, you must identify the installation. Make sure that the commands you enter run against the correct installation; see **setmqenv**.

About this task

You can back out maintenance updates (fix packs) and restore your system to the previous maintenance or installation level, for any component of IBM MQ for AIX that is in the **APPLIED** state.

The procedure detailed within this topic removes all the fix packs installed. See "Uninstalling a single fix pack on AIX" on page 15 for details on removing a single fix pack from your system.

You can apply and remove maintenance from an IBM 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 to 3 in the following maintenance procedure.

Use the following command to display the current state of the IBM 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 MQ installation.

   If you use the 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 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:

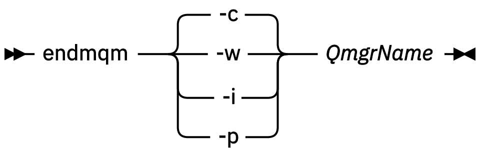
      ```
      dspmq -o installation -o status
      dspmq -a
      ```

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

      dspmq -a displays the status of active queue managers associated with the installation from which the command is run.
   b) Use the MQSC command **DISPLAY LSSTATUS** to list the status of listeners associated with a queue manager, as shown in the following example:

```
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 MQ libraries they have loaded. If they do not, you must find another way to force applications to release IBM 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 MQ. The application is not informed about queue managers in the current installation being shut down.

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

Note: "Applying maintenance level updates to multi-instance queue managers on Linux" on page 35 describes how to apply maintenance to a multi-instance queue manager. A multi-instance queue manager can continue to run on one server, while maintenance is applied to another server.

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 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 installation 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 MQ for AIX have been rejected

Related tasks

Stopping a queue manager

“Applying maintenance level updates to multi-instance queue managers on AIX” on page 19
On UNIX and Linux, you can use multi-instance queue managers to reduce the outage caused by applying maintenance updates.

Related reference
dspmq
DISPLAY LSSTATUS
endmqm (end queue manager)
endmqlsr (end listener)

AIX Uninstalling a single fix pack on AIX

You can remove a single fix pack using the **installp** command.

About this task

For example, you have installed IBM MQ for AIX 9.1.0.0 base, the IBM MQ for AIX 9.1.0.7 fix pack and the IBM MQ for AIX 9.1.0.8 fix pack.

You want to remove the 9.1.0.8 fix pack and leave the 9.1.0.7 fix pack.

Look at the installp command in the AIX manual, and specifically the information on the **-r** (reject) parameter.

The AIX manual lists a number of options for the **installp -r** parameter, and the one that is relevant is the **-f** *ListFile* option.

The description of the **-f** *ListFile* option includes the following statement: *Output from the installp -l command is suitable for input to this flag*.

You need to obtain a text file using the **installp -l** command. The AIX manual shows the following invocation for **installp -l** parameter:

```
installp { -l | -L } [ -eLogFile ] [ -d Device ] [ -B ] [ -I ] [ -q ] [-E ]
[ -zBlockSize ] [ -O { [ s ] [ u ] } ]
```

Note: The principles apply to a User Specified Installation Location (USIL) by using the **-R** *usil-directory* option, and to other IBM MQ fix packs.

See Life cycle for a USIL in AIX for non-default installations of MQ for more information on a USIL.

Carry out the following procedure to remove the latest 9.1.0.8 fix pack, and leave the base IBM MQ for AIX 9.1.0.0 and 9.1.0.7 fix packs in place.

Procedure

1. Issue the following command, `# lslpp -la "mqm*"`:
   You see the following output:

```
+----------------------------------------------------------------------------+
INSTALL ROOT PATH = /
+----------------------------------------------------------------------------+
Fileset Level State Description
----------------------------------------------------------------------------
Path: /usr/lib/objrepos
mqm.amqp.rte             9.1.0.0 COMMITTED IBM MQ AMQP Service
                  9.1.0.7 APPLIED IBM MQ AMQP Service
                  9.1.0.8 APPLIED IBM MQ AMQP Service
mqm.ams.rte              9.1.0.0 COMMITTED IBM MQ Advanced - Advanced Message Security
                  9.1.0.7 APPLIED IBM MQ Advanced - Advanced Message Security
                  9.1.0.8 APPLIED IBM MQ Advanced - Advanced Message Security
mqm.base.runtime         9.1.0.0 COMMITTED IBM MQ Runtime for Client and Server
                  9.1.0.7 APPLIED IBM MQ Runtime for Client and Server
                  9.1.0.8 APPLIED IBM MQ Runtime for Client and Server
```

2. Go to the original directory where the IBM MQ for AIX `tar.Z` file with the fix pack code, that was downloaded, was stored in the machine and unpacked; for example, `cd /downloads/mq9108`.

   Expand `9.1.0-IBM-MQ-AixPPC64-FP0008.tar.Z` and you see:

```
+++ROOT+++ aglet: /downloads/mq9108
# ls
.toc mqm.msg.Zh_CN.9.1.0.8.U202341
9.1.0-IBM-MQ-AixPPC64-FP0008.tar mqm.msg.Zh_TW.9.1.0.8.U202343
mq9108.installpl.txt mqm.msg.cs_CZ.9.1.0.8.U202327
mqm.amqp.rte.9.1.0.8.U202313 mqm.msg.de_DE.9.1.0.8.U202328
mqm.ams.rte.9.1.0.8.U202312 mqm.msg.en_US.9.1.0.8.U202329
mqm.base.runtime.9.1.0.8.U202314 mqm.msg.es_ES.9.1.0.8.U202330
mqm.base.samples.9.1.0.8.U202315 mqm.msg.fr_FR.9.1.0.8.U202331
mqm.base.sdk.9.1.0.8.U202316 mqm.msg.hu_HU.9.1.0.8.U202332
mqm.client.rte.9.1.0.8.U202317 mqm.msg.it_IT.9.1.0.8.U202333
mqm.ft.agent.9.1.0.8.U202318 mqm.msg.ja_JP.9.1.0.8.U202334
mqm.ft.base.9.1.0.8.U202319 mqm.msg.ko_KR.9.1.0.8.U202336
mqm.ft.logger.9.1.0.8.U202320 mqm.msg.pl_PL.9.1.0.8.U202337
mqm.ft.service.9.1.0.8.U202321 mqm.msg.pt_BR.9.1.0.8.U202338
mqm.ft.tools.9.1.0.8.U202322 mqm.msg.ru_RU.9.1.0.8.U202339
mqm.gskit.rte.9.1.0.8.U202323 mqm.msg.zh_CN.9.1.0.8.U202340
mqm.java.rte.9.1.0.8.U202324 mqm.msg.zh_TW.9.1.0.8.U202342
mqm.jre.rte.9.1.0.8.U202325 mqm.server.rte.9.1.0.8.U202344
mqm.man.en_US.data.9.1.0.8.U202326 mqm.web.rte.9.1.0.8.U202346
mqm.msg.Ja_JP.9.1.0.8.U202335 mqm.xr.service.9.1.0.8.U202345
```

3. Issue the following command to obtain the text file `mq9108.installpl.txt`, to be used later in the procedure: `# installp -l -d /downloads/mq9108 > mq9108.installpl.txt`

   The output text file looks like the following text.

   Note: Only the first few lines are shown here.

```
 Fileset Name Level I/U Q Content
 ====================================================================
 mqm.amqp.rte 9.1.0.8 S N usr (R)
# IBM MQ AMQP Service
 mqm.ams.rte 9.1.0.8 S N usr (R)
# IBM MQ Advanced - Advanced Message Security
 mqm.base.runtime 9.1.0.8 S N usr,root (R)
# IBM MQ Runtime for Client and Server
 mqm.base.samples 9.1.0.8 S N usr (R)
# IBM MQ Samples
```

4. Use the output file from Step “3” on page 16 as input to the following command: `# installp -r -f mq9108.installpl.txt`

```
Verifying selections...
done
Verifying requisites...done
Results...
SUCCESSES
---------
 Filesets listed in this section passed pre-reject verification
 and will be rejected.
 Selected Filesets
 -----------------Page 5 of 5
 mqm.amqp.rte 9.1.0.8 # IBM MQ AMQP Service
 mqm.ams.rte 9.1.0.8 # IBM MQ Advanced - Advanced M...
 mqm.base.runtime 9.1.0.8 # IBM MQ Runtime for Client an...
...
+-----------------------------------------------------------------------------+
Installation Summary
--------------------
Name         Level   Part       Event Result
-------------------------------------------------------------------------------
mqm.amqp.rte 9.1.0.8 USR REJECT SUCCESS
mqm.ams.rte 9.1.0.8 USR REJECT SUCCESS
mqm.base.runtime 9.1.0.8 ROOT REJECT SUCCESS
```

   Attention: Note the following:

   a. There will be around 30-40 blank lines following "*Verifying selections...*" and it took some time. you need to wait until the command produces meaningful output.

b. The output in the preceding text shows only a few lines of the final summary at the end and the result column shows SUCCESS.

5. Issue the command `# lslpp -la "mqm*"` again, and you see that the ones for 9.1.0.8 are no longer included:

   You see the following output:

```
+-----------------------------------------------------------------------------+
INSTALL ROOT PATH = /
+-----------------------------------------------------------------------------+
Fileset Level State Description
----------------------------------------------------------------------------
Path: /usr/lib/objrepos
mqm.amqp.rte             9.1.0.0 COMMITTED IBM MQ AMQP Service
                  9.1.0.7 APPLIED IBM MQ AMQP Service
mqm.ams.rte              9.1.0.0 COMMITTED IBM MQ Advanced - Advanced Message Security
                  9.1.0.7 APPLIED IBM MQ Advanced - Advanced Message Security
mqm.base.runtime         9.1.0.0 COMMITTED IBM MQ Runtime for Client and Server
                  9.1.0.7 APPLIED IBM MQ Runtime for Client and Server
```

Related tasks

"Reverting to the previous maintenance level on AIX" on page 13
You can revert to a previous maintenance level by using the *System Management Interface Tool* (SMIT).

Linux UNIX Staging maintenance level updates on AIX

On AIX, you can use multiple installations of IBM MQ on the same server to control the release of maintenance level updates.

Before you begin

The steps in this task are based on an example scenario in which it is assumed that you have two copies of IBM MQ named `Inst_1` and `Inst_2`, and a number of applications and two queue managers, `QM1` and `QM2`, running on a server. To set up your configuration for this scenario, complete the following steps:

1. Install two copies of IBM MQ. 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.

Note: From 1Q 2023, for Multiplatforms, there are two types of maintenance:

- Fix packs, which contain roll-ups of all defects fixed since the previous fix pack delivery (or GA). Fix packs are produced exclusively for Long Term Support (LTS) releases during their normal support lifecycle.
- Cumulative security updates (CSUs), which are smaller updates and contain security patches released since the previous maintenance (GA). CSUs are produced for LTS releases (including releases in extended support), and also for the latest IBM MQ Continuous Delivery (CD) release, as required to deliver relevant security patches.

For maintenance releases in or after 1Q 2023, the fourth digit in the VRMF represents either a fix pack number or a CSU number. Both types of maintenance are mutually cumulative (that is, they contain everything included in older CSUs and fix packs), and both are installed using the same mechanisms for applying maintenance. Both types of maintenance update the F-digit of the VRMF to a higher number than any previous maintenance: fix packs use "F" values divisible by 5, CSUs use "F" values not divisible by 5.

For maintenance releases before 1Q 2023, the fourth digit in the VRMF always represents the fix pack level. For example, the first fix pack of the IBM MQ 9.1.0 LTS release is numbered 9.1.0.1.

For more information, see Changes to IBM MQ's maintenance delivery model.

About this task

You can install multiple copies of IBM MQ on a server to stage the release of maintenance level updates. For example, as in the scenario that is described in the task steps, by using two installations to roll out maintenance level updates, you maintain two maintenance levels on a server, with the aim of getting all queue managers and applications to the production maintenance level before replacing the previous level of maintenance 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 re-associate a queue manager with either of the installations running any of the fix packs or cumulative security updates (CSUs).

In the example, an application links to the primary installation. When it connects to a queue manager, IBM 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 98.

For applications built with the link options described in the product documentation, the simplest way to configure the link library search path for IBM MQ applications is to make an installation primary. Only if it is important to pick up a fix in the IBM MQ link library itself, must you review the search path. Either you must make the installation with the IBM 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.

Procedure

Apply the first maintenance level update to Inst_2.

1. Download the first fix pack or cumulative security update (CSU) when it is released.

   See Where to find downloadable installation images.
2. Apply the fix pack or cumulative security update (CSU) you downloaded to Inst_2.

   See "Applying maintenance level updates on AIX" on page 10.
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.

   Note: "Applying maintenance level updates to multi-instance queue managers on AIX" on page 19 describes how to apply maintenance to a multi-instance queue manager. A multi-instance queue manager can continue to run on one server, while maintenance is applied to another server.

   b) Set up the local environment to the installation Inst_2.

   ```
   . 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.

5. Set Inst_2 primary.

   ```
   Inst_2_INSTALLATION_PATH/bin/setmqinst -i -n Inst_2
   ```

Apply the second maintenance level update to Inst_1.

6. Download the next fix pack or cumulative security update (CSU), for the version of your product when it is released.

   See Where to find downloadable installation images.
7. Apply the fix pack or cumulative security update (CSU), that you have just downloaded to Inst_1.
8. Verify Inst_1.
9. Transfer queue managers to Inst_1 one at a time.
   a) Follow the procedure in step “4” on page 18

      Replacing Inst_2 by Inst_1 in the instructions.
10. Set Inst_1 primary.

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

For subsequent maintenance updates, alternate between Inst_2 and Inst_1.

11. Alternate between repeating steps “1” on page 18 to “5” on page 19 for Inst_2 and steps “6” on page 19 to “10” on page 19 for Inst_1.

Related concepts

“Queue manager coexistence” on page 95
Queue managers, with different names, can coexist on any server as long as they use the same IBM MQ installation. On z/OS, 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 98
You can install multiple copies of IBM MQ for UNIX, Linux, and Windows on the same server. These IBM MQ copies can be at the same or different version levels. This is called a multi-installation. Multi-installation is particularly useful when you upgrade from one IBM MQ version to a later version, because it allows you to run the earlier version alongside the later version.

Related tasks

Installing IBM MQ server on Windows

Associating a queue manager with an installation

Changing the primary installation

Related reference

setmqenv

setmqinst

setmqm

Linux UNIX Applying maintenance level updates to multi-instance queue managers on AIX

On UNIX and Linux, you can use multi-instance queue managers to reduce the outage caused by applying maintenance updates.

Before you begin

Before starting this task, read through the prerequisites described in *Before you begin* in “Applying maintenance level updates on AIX” on page 10.

Before starting this task, see that the maintenance is applied to the IBM MQ installation on a server and not to individual queue managers. Before you apply maintenance, you must stop all the queue managers, and any IBM MQ service, on a server.

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 the queue manager that you want to keep running is an existing single instance queue manager, you must convert it to a multi-instance queue manager. For prerequisites and guidance how to create a multi-instance queue manager, see Multi-instance queue managers.

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 update. 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 an IBM MQ server installation.

Procedure

1. Where the maintenance update procedure instructs you to stop all running queue managers, or quiesce IBM MQ do the following instead:
   a) If the queue manager is running as standby:
      - End the standby with the **endmqm** `-x` *`QMgrName`* command.

   b) If the queue manager is running as the active instance:

      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 IBM MQ and apply maintenance to the IBM MQ server.
3. When you have completed the maintenance update, restart all the queue managers on the IBM MQ server, permitting standby instances:

   Use the following command:

   ```
   strmqm -x QmgrName
   ```
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:

Use the **endmqm** *-shutdown_option* -s *QMgrName* command, and the restart the instances using the **strmqm** -x *QmgrName* command.

IBM i Applying and removing maintenance on IBM i

Maintenance tasks associated with IBM i platforms are grouped in this section.

Procedure

- To apply maintenance level updates, see "Applying maintenance level updates on IBM i" on page 21.
- To restore a queue manager to the previous version of the product from the latest version, see "Restoring a queue manager to a previous release on IBM i" on page 24.
- For information on how to use use multi-instance queue managers to reduce the outage caused by applying maintenance updates, see "Applying maintenance updates to multi-instance queue managers on IBM i" on page 25.

Related tasks

"Applying maintenance to IBM MQ" on page 8
Maintenance is the application of a reversible fix. Any changes to queue manager data are compatible with the previous code level.

IBM i Applying maintenance level updates on IBM i

You apply maintenance level updates on the latest release by stopping IBM MQ and using the IBM i standard maintenance procedure.

Before you begin

To find out what version you have currently installed, use the following commands:

Table 1. IBM MQ commands to display the installed versions

IBM MQ Product	Version command
IBM MQ Server	`DSPMQMVER`

IBM MQ Java	IBM MQ classes for Java:
	`java com.ibm.mq.MQJavaLevel`
	Note: For this command to work, you might need to set your environment classpath to include:
	• `/QIBM/ProdData/mqm/java/lib/com.ibm.mq.jar`
	IBM MQ classes for Java Message Service:
	`java com.ibm.mq.jms.MQJMSLevel`
	Note: For this command to work, you might need to set your environment classpath to include:
	• `/QIBM/ProdData/mqm/java/lib/com.ibm.mqjms.jar`
	See Environment variables relevant to IBM MQ classes for Java and Environment variables relevant to IBM MQ classes for JMS.

Table 1. IBM MQ commands to display the installed versions (continued)

IBM MQ Product	Version command
IBM MQ Client	`DSPMQMVER`

About this task

Maintenance updates for IBM i are supplied as PTFs (Program Temporary Fixes). They are available for download from the web as save files, which are normally stored in the QGPL library. IBM i PTF's can be found in Fix Central.

Procedure

Prepare to quiesce queue managers:

1. Read the cover letter carefully to see if you need to take any special actions.
2. Sign on to a new interactive IBM i session, ensuring that you are not accessing any IBM MQ objects.
3. Ensure that you have the following authorities:
   - `*ALLOBJ` authority, or object management authority for the `QMQM` library.
   - Sufficient authority to use the ENDSBS command.
4. Warn all users that you are going to stop IBM MQ.
5. Stop the mqweb server by entering the following command:

   ```
   ENDMQWEB
   ```

Quiesce all queue managers:

6. Run the ENDMQM command:

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

   Where *15* is a timeout value in seconds.

   If the ENDMQM command has not completed within a reasonable period (at least 10 minutes), run the WRKMQM command. This command identifies the queue managers that are still ending. Then force each one in turn to stop by running the following command:

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

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

   Complete the tidying up of shared memory by running the following command:

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

If the commands in the previous step do not complete, end the subsystem immediately:

7. Run the following command:

   ```
   ENDSBS SBS(QMQM) OPTION(*IMMED)
   ```

If the command in the previous step also does not complete, use the operating system command ENDJOB to end all jobs in the subsystem QMQM:

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

8. If a QMGR must be shut down manually, end the jobs (ENDJOB) in the following order. Wait a few minutes for AMQA* or AMQZ* jobs to tidy up.

   a. `RUNMQLSR` - TCP listener (multi-threaded)
   b. `AMQCLMAA` - TCP listener (single-threaded)
   c. `AMQRMPPA` - Channel process pooling job
   d. `RUNMQCHI` - channel initiator
   e. `AMQCRSTA` - receiving MCA jobs
   f. `RUNMQCHL` - sending MCA jobs
   g. `AMQCRS6B` - LU62 receiver channel
   h. `AMQPCSEA` - command server
   i. `RUNMQTRM` - Application trigger monitor
   j. `RUNMQDLQ` - Dead letter queue handler
   k. `AMQFCXBA` - IBM Integration Bus Worker Job
   l. `AMQFQPUB` - Queued Publish/Subscribe Daemon
   m. `RUNMQBRK` - IBM Integration Bus Control Job
   n. `AMQZMUC0` ('0' is a zero) - Utility Manager
   o. `AMQZMUF0` ('0' is a zero) - Utility Manager
   p. `AMQZMUR0` ('0' is a zero) - Utility Manager
   q. `AMQZMGR0` ('0' is a zero) - Process Controller
   r. `AMQRRMFA` - cluster repository manager
   s. `AMQZDMAA` - deferred message manager
   t. `AMQZFUMA` - object authority manager
   u. `AMQZLSA0` ('0' is a zero) - LQM agents
   v. `AMQZLAA0` ('0' is a zero) - LQM agents
   w. `AMQZXMA0` ('0' is a zero) - Execution Controller

9. Run the following command:

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

10. Run the following command:

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

    Where *05* is a timeout value in seconds.

11. Manually clean up shared memory.

    Run the following command:

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

    then:

    a. Take option 5 for **&SYSTEM** and check that the following directories are empty: `isem, esem, msem, ssem,` and `shmem`.

b. Take option 5 for **QMGRNAME** and check that the following directories are empty:- `isem, esem, msem, ssem,` and `shmem`.

c. Take option 5 for **&ipcc** in the QMGRNAME directory and check that the following directories are empty:- `isem, esem, msem, ssem,` and `shmem`.

d. Take option 5 for **&qmpersist** in the QMGRNAME directory and check that the following directories are empty:- `isem, esem, msem, ssem,` and `shmem`.

e. Take option 5 for **&app** and check that the following directories are empty: `isem, esem, msem, ssem,` and `shmem`.

Apply a PTF:

12. Load and apply a PTF.

IBM i Restoring a queue manager to a previous release on IBM i

On IBM i, you can restore a queue manager to the previous version of the product from the latest version, 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 the later version. For more information see Backing up and restoring IBM 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 revert the queue manager to the earlier version of the product 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.

About this task

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

Important: If the queue manager is a member of one or more IBM MQ clusters, you should also review and follow the steps described in Recovering a cluster queue manager.

Procedure

1. Stop the queue manager.
2. If you performed a slip installation, you must reinstall IBM MQ.
   a) Uninstall the earlier installation.
   b) Reinstall the product from a manufacturing refresh.
   c) Apply the fix pack and interim fixes that restore IBM MQ to its previous level.
   d) Restore the queue manager data from the backup taken before installing the later version.
3. Restart the earlier version queue manager.

Related tasks

Backing up and restoring a queue manager

IBM i Applying maintenance updates to multi-instance queue managers on IBM i

On IBM i, you can use multi-instance queue managers to reduce the outage caused by applying maintenance updates.

Before you begin

Before starting this task, read through the prerequisites described in *Before you begin* in "Applying maintenance level server updates on Windows" on page 47

Before starting this task, see that the maintenance is applied to the IBM MQ installation on a server and not to individual queue managers. Before you apply maintenance, you must stop all the queue managers, and any IBM MQ service, on a server.

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 the queue manager that you want to keep running is an existing single instance queue manager, you must convert it to a multi-instance queue manager. For prerequisites and guidance how to create a multi-instance queue manager, see Multi-instance queue managers.

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 update. 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 an IBM MQ server installation.

Procedure

1. Where the maintenance update procedure instructs you to stop all running queue managers, or quiesce IBM MQ do the following instead:

   See: "Applying and removing maintenance on IBM i" on page 21.

   a) If the queue manager is running as standby:

      End the standby by adding the `INSTANCE(*STANDBY)` option to the **ENDMQM** command.

   b) If the queue manager is running as the active instance:

      End the instance and transfer control to the standby instance by adding the `ALWSWITCH(*YES)` option to the **ENDMQM** command.

      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 IBM MQ and apply maintenance to the IBM MQ server.
3. When you have completed the maintenance update, restart all the queue managers on the IBM MQ server, permitting standby instances:

   Add the `STANDBY(*YES)` option to the **STRMQM** 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:

   Use the **ENDMQM** command with the `ALWSWITCH(*YES)` option, and then restart the instances using the **STRMQM** command with the `STANDBY(*YES)` option.

Linux Applying and removing maintenance on Linux

Maintenance tasks associated with Linux are grouped in this section.

Related tasks

"Applying maintenance to IBM MQ" on page 8

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

Linux Applying maintenance level updates on Linux

You can apply maintenance level updates to IBM MQ on Linux by using RPM. The same procedure applies to all Linux platforms, including Ubuntu.

Before you begin

If you are running on a server with multiple IBM MQ installations, you must identify the installation. Make sure that the commands you enter run against the correct installation; see **setmqenv**.

About this task

Maintenance level updates are delivered in the form of RPM Package Manager (RPM) update images, which are applied using the RPM installation tool.

You can apply and remove maintenance from an IBM 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 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.

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 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 MQ.

Procedure

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

   If you use the 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. Stop the mqweb server that is associated with the IBM MQ installation:

   a) Check whether the mqweb server is running by entering the following command:

   ```
   dspmqweb status
   ```

   b) Stop the mqweb server by entering the following command:

   ```
   endmqweb
   ```
4. End all the activity of queue managers associated with the IBM 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
   ```

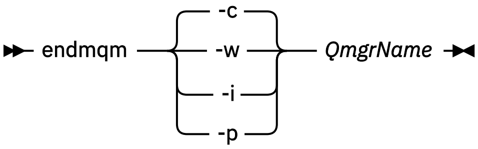
   `dspmq -o installation -o status` displays the installation name and status of queue managers associated with all installations of IBM MQ.

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

   b) Use the MQSC command **`DISPLAY LSSTATUS`** to list the status of listeners associated with a queue manager, as shown in the following example:

   ```
   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 MQ libraries they have loaded. If they do not, you must find another way to force applications to release IBM 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 MQ. The application is not informed about queue managers in the current installation being shut down.

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

Note: “Applying maintenance level updates to multi-instance queue managers on Linux” on page 35 describes how to apply maintenance to a multi-instance queue manager. A multi-instance queue manager can continue to run on one server, while maintenance is applied to another server.

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

```
endmqlsr -m QMgrName
```

5. Log in as root, or switch to the superuser using the **su** command.
6. Change into the directory containing the maintenance packages.
7. 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-V.R.0-1.i386.rpm
   MQSeriesSDK-Uxxxx-V.R.0-1.i386.rpm
   MQSeriesServer-Uxxxx-V.R.0-1.i386.rpm
   ```

   where V is the version number and R is the number of the Release.
8. Run the **rpm** command to find out which packages are installed on your server.

   Enter the following command:

   ```
   rpm -qa | grep MQSeries
   ```

   Note: If you are using rpm on Ubuntu, add the **--force-debian** attribute.

   ```
   rpm --force-debian -qa | grep MQSeries
   ```

   For example, if you have a minimum IBM MQ installation and SDK component, at level 0, the **rpm** command returns:

   ```
   MQSeriesRuntime-V.R.0-0
   MQSeriesSDK-V.R.0-0
   MQSeriesServer-V.R.0-0
   ```

   where V is the version number and R is the number of the Release.
9. 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 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 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 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-V.R.0-1.xxxx.rpm` to `MQSeriesRuntime_1-V.R.0-1.xxxx.rpm`. Where V is the version number and R is the number of the Release.
10. 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-V.R.0-1.i386.rpm
    MQSeriesSDK-Uxxxx-V.R.0-1.i386.rpm
    MQSeriesServer-Uxxxx-V.R.0-1.i386.rpm
  ```

  where V is the version number and 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-V.R.0-1.i386.rpm
    MQSeriesSDK-Uxxxx-V.R.0-1.i386.rpm
    MQSeriesServer-Uxxxx-V.R.0-1.i386.rpm
  ```

  where V is the version number and 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.

11. Repeat step “8” on page 28 to list the packages that are now available.

    The Runtime, SDK, and Server packages are now at level 1:

    ```
    MQSeriesRuntime-V.R.0-0
    MQSeriesSDK-V.R.0-0
    MQSeriesServer-V.R.0-0
    MQSeriesRuntime-Uxxxx-V.R.0-1
    MQSeriesSDK-Uxxxx-V.R.0-1
    MQSeriesServer-Uxxxx-V.R.0-1
    ```

    where V is the version number and R is the number of the Release.

    Note:

    After the installation of IBM 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 MQ fix pack installation code. For further information about this error, see IBM MQ Fix Pack installation errors - Linux reports errors

What to do next

For further information about using RPM to install software packages, see your Linux documentation.

Related tasks

Stopping a queue manager

“Applying maintenance level updates to multi-instance queue managers on AIX” on page 19

On UNIX and Linux, you can use multi-instance queue managers to reduce the outage caused by applying maintenance updates.

Related reference

dspmq

DISPLAY LSSTATUS

endmqm (end queue manager)

endmqlsr (end listener)

Linux Reverting to the previous maintenance level on Linux

You can remove updates and revert to the previous maintenance level of IBM MQ by using **RPM**. The same procedure applies to all Linux platforms, including Ubuntu.

Before you begin

If you are running on a server with multiple IBM MQ installations, you must identify the installation. Make sure that the commands you enter run against the correct installation; see **setmqenv**.

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 RPM 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 MQ installation.

   If you use the 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 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
   ```

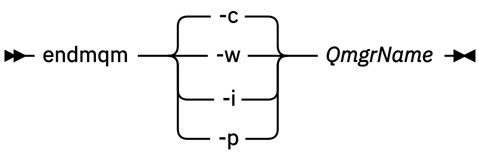
   dspmq -o installation -o status displays the installation name and status of queue managers associated with all installations of IBM MQ.

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

   b) Use the MQSC command **DISPLAY LSSTATUS** to list the status of listeners associated with a queue manager, as shown in the following example:

   ```
   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 MQ libraries they have loaded. If they do not, you must find another way to force applications to release IBM 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 MQ. The application is not informed about queue managers in the current installation being shut down.

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

Note: “Applying maintenance level updates to multi-instance queue managers on Linux” on page 35 describes how to apply maintenance to a multi-instance queue manager. A multi-instance queue manager can continue to run on one server, while maintenance is applied to another server.

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
   ```

   Note: If you are using Ubuntu, add the **--force-debian** attribute.

   ```
   rpm --force-debian -qa | grep MQSeries
   ```

   Using the example given in “Applying maintenance level updates on Linux” on page 26, returns:

   ```
   MQSeriesRuntime-V.R.0-0
   MQSeriesSDK-V.R.0-0
   MQSeriesServer-V.R.0-0
   MQSeriesRuntime-Uxxxx-V.R.0-1
   MQSeriesSDK-Uxxxx-V.R.0-1
   MQSeriesServer-Uxxxx-V.R.0-1
   ```

   where V is the version number and R is the number of the Release.

   Note: If you are using the RDQM facility you might need to use a different process to revert to a prior level. This involves removing the RDQM support using the supplied script and reinstalling from the media supplied with the earlier version.

   See Installing RDQM (replicated data queue managers) for more information.
6. Run the **rpm** command to remove all the updates applied at level 1.

   Enter the following commands:

   ```
   rpm -ev MQSeriesRuntime-Uxxxx-V.R.0-1 MQSeriesSDK-Uxxxx-V.R.0-1
   MQSeriesServer-Uxxxx-V.R.0-1
   ```

   where V is the version number and R is the number of the Release.
7. Repeat step 5 to check that the ptf packages have been removed, leaving only the original installation packages:

   ```
   MQSeriesRuntime-V.R.0-0
   MQSeriesSDK-V.R.0-0
   MQSeriesServer-V.R.0-0
   ```

   where V is the version number and 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 tasks

Stopping a queue manager

“Applying maintenance level updates to multi-instance queue managers on AIX” on page 19

On UNIX and Linux, you can use multi-instance queue managers to reduce the outage caused by applying maintenance updates.

Related reference

dspmq

DISPLAY LSSTATUS

endmqm (end queue manager)

endmqlsr (end listener)

Linux Staging maintenance level updates on Linux

On Linux, you can use multiple installations of IBM MQ on the same server to control the release of maintenance level updates.

Before you begin

The steps in this task are based on an example scenario in which it is assumed that you have two copies of IBM MQ named `Inst_1` and `Inst_2`, and a number of applications and two queue managers, `QM1` and `QM2`, running on a server. To set up your configuration for this scenario, complete the following steps:

1. Install two copies of IBM MQ. 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 MQ Explorer.
6. Set up remote connections to all the queue managers in each instance of IBM MQ Explorer.

Note: From 1Q 2023, for Multiplatforms, there are two types of maintenance:

- Fix packs, which contain roll-ups of all defects fixed since the previous fix pack delivery (or GA). Fix packs are produced exclusively for Long Term Support (LTS) releases during their normal support lifecycle.
- Cumulative security updates (CSUs), which are smaller updates and contain security patches released since the previous maintenance (GA). CSUs are produced for LTS releases (including releases in extended support), and also for the latest IBM MQ Continuous Delivery (CD) release, as required to deliver relevant security patches.

For maintenance releases in or after 1Q 2023, the fourth digit in the VRMF represents either a fix pack number or a CSU number. Both types of maintenance are mutually cumulative (that is, they contain everything included in older CSUs and fix packs), and both are installed using the same mechanisms for applying maintenance. Both types of maintenance update the F-digit of the VRMF to a higher number than any previous maintenance: fix packs use "F" values divisible by 5, CSUs use "F" values not divisible by 5.

For maintenance releases before 1Q 2023, the fourth digit in the VRMF always represents the fix pack level. For example, the first fix pack of the IBM MQ 9.1.0 LTS release is numbered 9.1.0.1.

For more information, see Changes to IBM MQ's maintenance delivery model.

About this task

You can install multiple copies of IBM MQ on a server to stage the release of maintenance level updates. For example, as in the scenario that is described in the task steps, by using two installations to roll out maintenance level updates, you maintain two maintenance levels on a server, with the aim of getting all queue managers and applications to the production maintenance level before replacing the previous level of maintenance 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 re-associate a queue manager with either of the installations running any of the fix packs or cumulative security updates (CSUs).

In the example, an application links to the primary installation. When it connects to a queue manager, IBM 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 98.

For applications built with the link options described in the product documentation, the simplest way to configure the link library search path for IBM MQ applications is to make an installation primary. Only if it is important to pick up a fix in the IBM MQ link library itself, must you review the search path. Either you must make the installation with the IBM 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 MQ Explorer and you have two installations, you also have two IBM MQ Explorer instances. One linked to one installation, and one to the other. Each IBM MQ Explorer shows locally connected queue managers that are associated with the same installation as the instance of IBM MQ Explorer. To monitor all the queue managers on a server, set up remote connections to the queue managers associated with the other installations.

Procedure

Apply the first maintenance level update to Inst_2.

1. Download the first fix pack or cumulative security update (CSU) when it is released.

   See Where to find downloadable installation images.
2. Apply the fix pack or cumulative security update (CSU) you downloaded to Inst_2.
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.

      Note: "Applying maintenance level updates to multi-instance queue managers on Linux" on page 35 describes how to apply maintenance to a multi-instance queue manager. A multi-instance queue manager can continue to run on one server, while maintenance is applied to another server.

   b) Set up the local environment to the installation Inst_2.

      ```
      . 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 MQ Explorer for Inst_2.

i) Start the `Inst_2` instance of IBM MQ Explorer

ii) Click **IBM 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.

```
Inst_2_INSTALLATION_PATH/bin/setmqinst -i -n Inst_2
```

Apply the second maintenance level update to `Inst_1`.

6. Download the next fix pack or cumulative security update (CSU) for the version of your product when it is released.

   See Where to find downloadable installation images.
7. Apply the fix pack or cumulative security update (CSU) that you have just downloaded to `Inst_1`.
8. Verify `Inst_1`.
9. Transfer queue managers to `Inst_1` one at a time.

   a) Follow the procedure in step “4” on page 33

   Replacing `Inst_2` by `Inst_1` in the instructions.
10. Set `Inst_1` primary.

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

For subsequent maintenance fixes, alternate between `Inst_2` and `Inst_1`.

11. Alternate between repeating steps “1” on page 33 to “5” on page 34 for `Inst_2` and steps “6” on page 34 to “10” on page 34 for `Inst_1`.

Related concepts

“Queue manager coexistence” on page 95
Queue managers, with different names, can coexist on any server as long as they use the same IBM MQ installation. On z/OS, 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 98
You can install multiple copies of IBM MQ for UNIX, Linux, and Windows on the same server. These IBM MQ copies can be at the same or different version levels. This is called a multi-installation. Multi-installation is particularly useful when you upgrade from one IBM MQ version to a later version, because it allows you to run the earlier version alongside the later version.

Related tasks

Installing IBM MQ server on Windows
Associating a queue manager with an installation
Changing the primary installation

Related reference

setmqenv
setmqinst
setmqm

Linux Applying maintenance level updates to multi-instance queue managers on Linux

On Linux, you can use multi-instance queue managers to reduce the outage caused by applying maintenance updates.

Before you begin

Before starting this task, read through the prerequisites described in *Before you begin* in "Applying maintenance level updates on Linux" on page 26.

Before starting this task, see that the maintenance is applied to the IBM MQ installation on a server and not to individual queue managers. Before you apply maintenance, you must stop all the queue managers, and any IBM MQ service, on a server.

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 the queue manager that you want to keep running is an existing single instance queue manager, you must convert it to a multi-instance queue manager. For prerequisites and guidance how to create a multi-instance queue manager, see Multi-instance queue managers.

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 update. 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 an IBM MQ server installation.

Procedure

1. Where the maintenance update procedure instructs you to stop all running queue managers, or quiesce IBM MQ do the following instead:

   a) If the queue manager is running as standby:

      - End the standby with the **endmqm** `-x` *`QMgrName`* command.

   b) If the queue manager is running as the active instance:

      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 IBM MQ and apply maintenance to the IBM MQ server.
3. When you have completed the maintenance update, restart all the queue managers on the IBM MQ server, permitting standby instances:

   Use the following command:

   ```
   strmqm -x QmgrName
   ```
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:

   Use the **endmqm** *-shutdown_option* -s *QMgrName* command, and the restart the instances using the **strmqm** -x *QmgrName* command.

Solaris Applying and removing maintenance on Solaris

Maintenance tasks associated with Solaris are grouped in this section.

Related tasks

"Applying maintenance to IBM MQ" on page 8

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

Solaris Applying maintenance level updates on IBM MQ on Solaris

You can apply maintenance level updates to IBM MQ for Solaris using **pkgadd**.

Before you begin

1. Ensure that 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 MQ installations, 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 an IBM 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 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 MQ.

Procedure

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

If you use the 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. Stop the mqweb server that is associated with the IBM MQ installation:

   a) Check whether the mqweb server is running by entering the following command:

   ```
   dspmqweb status
   ```

   b) Stop the mqweb server by entering the following command:

   ```
   endmqweb
   ```

4. End all the activity of queue managers associated with the IBM 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
   ```

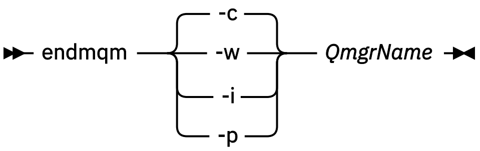
   `dspmq -o installation -o status` displays the installation name and status of queue managers associated with all installations of IBM MQ.

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

   b) Use the MQSC command **`DISPLAY LSSTATUS`** to list the status of listeners associated with a queue manager, as shown in the following example:

   ```
   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 MQ libraries they have loaded. If they do not, you must find another way to force applications to release IBM 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 MQ. The application is not informed about queue managers in the current installation being shut down.

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

   Note: "Applying maintenance level updates to multi-instance queue managers on Linux" on page 35 describes how to apply maintenance to a multi-instance queue manager. A multi-instance queue manager can continue to run on one server, while maintenance is applied to another server.

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

```
endmqlsr -m QMgrName
```

5. Log in as root, or switch to the superuser using the **su** command.
6. Change into the directory containing the maintenance packages.
7. 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 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.

8. Proceed with installation using the following command:

   Enter the following command to start the installation process if this fix pack is to be upgraded on an installation that is

   a) The first installation on the system:

      ```
      pkgadd -d packagename
      ```

      where `packagename` corresponds to the image file name. For example:

      ```
      mqm-U1234.img
      ```

   b) Not the first installation on the system:

      ```
      pkgadd mqm-suffixname
      ```

      where `suffixname` is the name of the directory created in `/var/spool/pkg`.

      For example, if you install IBM WebSphere MQ 7.0 as a package called `mqm-main7` and create a package to upgrade to IBM WebSphere MQ 7.0.0 Fix Pack 1, using the command **crtmqfp** `mqm-main7`, package *mqm-main7-07-00-00-01* is created in `/var/spool/pkg`.

      To install package *mqm-main7-07-00-00-01*, issue the command **pkgadd** `mqm-main7-07-00-00-01`.

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

9. Follow the on-screen instructions.

Related tasks

Stopping a queue manager

"Applying maintenance level updates to multi-instance queue managers on AIX" on page 19
On UNIX and Linux, you can use multi-instance queue managers to reduce the outage caused by applying maintenance updates.

Related reference

dspmq

DISPLAY LSSTATUS

endmqm (end queue manager)

endmqlsr (end listener)

Applying maintenance level updates in non-interactive mode on Solaris

You can install IBM MQ for Solaris non-interactively by creating a response file and an admin file.

Before you begin

1. Ensure that 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 MQ installations, 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 an IBM 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 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 MQ.

Procedure

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

   If you use the 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. Stop the mqweb server that is associated with the IBM MQ installation:
   a) Check whether the mqweb server is running by entering the following command:

   ```
   dspmqweb status
   ```

   b) Stop the mqweb server by entering the following command:

   ```
   endmqweb
   ```
4. End all the activity of queue managers associated with the IBM 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
   ```

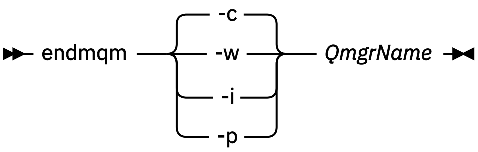
   `dspmq -o installation -o status` displays the installation name and status of queue managers associated with all installations of IBM MQ.

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

   b) Use the MQSC command **`DISPLAY LSSTATUS`** to list the status of listeners associated with a queue manager, as shown in the following example:

```
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 MQ libraries they have loaded. If they do not, you must find another way to force applications to release IBM 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 MQ. The application is not informed about queue managers in the current installation being shut down.

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

Note: "Applying maintenance level updates to multi-instance queue managers on Linux" on page 35 describes how to apply maintenance to a multi-instance queue manager. A multi-instance queue manager can continue to run on one server, while maintenance is applied to another server.

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

```
endmqlsr -m QMgrName
```

5. Log in as root, or switch to the superuser using the **su** command.
6. Change into the directory containing the maintenance packages.
7. 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 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.
8. Create the non-interactive installation response file using the **pkgask** command. Enter the following command to create the response file if this fix pack is to be upgraded on an installation that is:

   a) The first installation on the system:

   ```
   pkgask -d location_to_image/imagefile -r response.txt packagename
   ```

   where *imagefile* corresponds to the image file name, for example `mqm-U200403.img`, `response.txt` is the name of the response file to create, and *packagename* is the fix pack package name, for example `mqm-07-05-00-02`.

   b) Not the first installation on the system:

```
pkgask -d /var/spool/pkg -r response.txt mqm-suffixname
```

where `/var/spool/pkg` is the location of the new package, `response.txt` is the name of the response file to create, and *suffixname* is the name of the directory created in `/var/spool/pkg`.

9. 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
   ```

10. Run the **pkgadd** command to apply the maintenance level update IBM MQ for Solaris in non-interactive mode. Enter the following command to start the installation process if this fix pack is to be upgraded on an installation that is:

    a) The first installation on the system:

    ```
    pkgadd -v -n -r response.txt -a admin_file -d location_to_image/imagefile packagename
    ```

    where *admin_file* is a path qualified name of the admin file you created, and *packagename* corresponds to the fix pack package being installed.

    b) Not the first installation on the system:

    ```
    pkgadd -v -n -r response.txt -a admin_file -d /var/spool/pkg mqm-suffixname
    ```

11. Follow the on-screen instructions.

Related tasks

Stopping a queue manager

"Applying maintenance level updates to multi-instance queue managers on AIX" on page 19
On UNIX and Linux, you can use multi-instance queue managers to reduce the outage caused by applying maintenance updates.

Related reference

dspmq

DISPLAY LSSTATUS

endmqm (end queue manager)

endmqlsr (end listener)

Solaris Reverting to the previous maintenance level on Solaris

You can revert to a previous maintenance level of IBM MQ by stopping IBM MQ and using **pkgrm**.

Before you begin

If you are running on a server with multiple IBM MQ installations, you must identify the installation. Make sure that the commands you enter run against the correct installation; see **setmqenv**.

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 MQ installation.

   If you use the 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 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
   ```

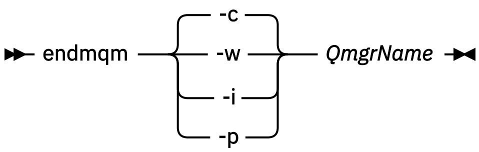
   dspmq -o installation -o status displays the installation name and status of queue managers associated with all installations of IBM MQ.

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

   b) Use the MQSC command **DISPLAY LSSTATUS** to list the status of listeners associated with a queue manager, as shown in the following example:

   ```
   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 MQ libraries they have loaded. If they do not, you must find another way to force applications to release IBM 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 MQ. The application is not informed about queue managers in the current installation being shut down.

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

   Note: "Applying maintenance level updates to multi-instance queue managers on Linux" on page 35 describes how to apply maintenance to a multi-instance queue manager. A multi-instance queue manager can continue to run on one server, while maintenance is applied to another server.

   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`, 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 an IBM MQ MQI client, and the client was updated after installing the maintenance level that is being removed, you must specifically update your IBM MQ MQI client installation again, after the maintenance level has been removed

Related tasks

Stopping a queue manager

"Applying maintenance level updates to multi-instance queue managers on AIX" on page 19
On UNIX and Linux, you can use multi-instance queue managers to reduce the outage caused by applying maintenance updates.

Related reference

dspmq

DISPLAY LSSTATUS

endmqm (end queue manager)

endmqlsr (end listener)

Solaris Staging maintenance level updates on Solaris

On Solaris, you can use multiple installations of IBM MQ on the same server to control the release of maintenance level updates.

Before you begin

The steps in this task are based on an example scenario in which it is assumed that you have two copies of IBM MQ named `Inst_1` and `Inst_2`, and a number of applications and two queue managers, `QM1` and `QM2`, running on a server. To set up your configuration for this scenario, complete the following steps:

1. Install two copies of IBM MQ. 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.

Note: From 1Q 2023, for Multiplatforms, there are two types of maintenance:

- Fix packs, which contain roll-ups of all defects fixed since the previous fix pack delivery (or GA). Fix packs are produced exclusively for Long Term Support (LTS) releases during their normal support lifecycle.
- Cumulative security updates (CSUs), which are smaller updates and contain security patches released since the previous maintenance (GA). CSUs are produced for LTS releases (including releases in

extended support), and also for the latest IBM MQ Continuous Delivery (CD) release, as required to deliver relevant security patches.

For maintenance releases in or after 1Q 2023, the fourth digit in the VRMF represents either a fix pack number or a CSU number. Both types of maintenance are mutually cumulative (that is, they contain everything included in older CSUs and fix packs), and both are installed using the same mechanisms for applying maintenance. Both types of maintenance update the F-digit of the VRMF to a higher number than any previous maintenance: fix packs use "F" values divisible by 5, CSUs use "F" values not divisible by 5.

For maintenance releases before 1Q 2023, the fourth digit in the VRMF always represents the fix pack level. For example, the first fix pack of the IBM MQ 9.1.0 LTS release is numbered 9.1.0.1.

For more information, see Changes to IBM MQ's maintenance delivery model.

About this task

You can install multiple copies of IBM MQ on a server to stage the release of maintenance level updates. For example, as in the scenario that is described in the task steps, by using two installations to roll out maintenance level updates, you maintain two maintenance levels on a server, with the aim of getting all queue managers and applications to the production maintenance level before replacing the previous level of maintenance 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 re-associate a queue manager with either of the installations running any of the fix packs or cumulative security updates (CSUs).

In the example, an application links to the primary installation. When it connects to a queue manager, IBM 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 98.

For applications built with the link options described in the product documentation, the simplest way to configure the link library search path for IBM MQ applications is to make an installation primary. Only if it is important to pick up a fix in the IBM MQ link library itself, must you review the search path. Either you must make the installation with the IBM 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.

Procedure

Apply the first maintenance level update to `Inst_2`.

1. Download the first fix pack or cumulative security update (CSU) when it is released.

   See Where to find downloadable installation images.
2. Apply the fix pack or cumulative security update (CSU) you downloaded to `Inst_2`.
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.

      Note: "Applying maintenance level updates to multi-instance queue managers on Solaris" on page 46 describes how to apply maintenance to a multi-instance queue manager. A multi-instance queue manager can continue to run on one server, while maintenance is applied to another server.

b) Set up the local environment to the installation Inst_2.

```
. 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.

5. Set Inst_2 primary.

```
Inst_2_INSTALLATION_PATH/bin/setmqinst -i -n Inst_2
```

Apply the second maintenance level update to Inst_1.

6. Download the next fix pack or cumulative security update (CSU) for the version of your product, for example, IBM MQ 9.1.0 Fix Pack 2, when it is released.

   See Where to find downloadable installation images.
7. Apply the fix pack or cumulative security update (CSU) that you have just downloaded to Inst_1.
8. Verify Inst_1.
9. Transfer queue managers to Inst_1 one at a time.

   a) Follow the procedure in step “4” on page 44

   Replacing Inst_2 by Inst_1 in the instructions.
10. Set Inst_1 primary.

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

For subsequent maintenance fixes, alternate between Inst_2 and Inst_1.

11. Alternate between repeating steps “1” on page 44 to “5” on page 45 for Inst_2 and steps “6” on page 45 to “10” on page 45 for Inst_1.

Related concepts

“Queue manager coexistence” on page 95
Queue managers, with different names, can coexist on any server as long as they use the same IBM MQ installation. On z/OS, 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 98
You can install multiple copies of IBM MQ for UNIX, Linux, and Windows on the same server. These IBM MQ copies can be at the same or different version levels. This is called a multi-installation. Multi-installation is particularly useful when you upgrade from one IBM MQ version to a later version, because it allows you to run the earlier version alongside the later version.

Related tasks

Installing IBM MQ server on Windows

Associating a queue manager with an installation

Changing the primary installation

Related reference

setmqenv

setmqinst

setmqm

Solaris Applying maintenance level updates to multi-instance queue managers on Solaris

On Solaris, you can use multi-instance queue managers to reduce the outage caused by applying maintenance updates.

Before you begin

Before starting this task, read through the prerequisites described in *Before you begin* in "Applying maintenance level updates on IBM MQ on Solaris" on page 36,

Before starting this task, see that the maintenance is applied to the IBM MQ installation on a server and not to individual queue managers. Before you apply maintenance, you must stop all the queue managers, and any IBM MQ service, on a server.

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 the queue manager that you want to keep running is an existing single instance queue manager, you must convert it to a multi-instance queue manager. For prerequisites and guidance how to create a multi-instance queue manager, see Multi-instance queue managers.

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 update. 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 an IBM MQ server installation.

Procedure

1. Where the maintenance update procedure instructs you to stop all running queue managers, or quiesce IBM MQ do the following instead:

   a) If the queue manager is running as standby:

      - End the standby with the **endmqm** `-x` *`QMgrName`* command.

   b) If the queue manager is running as the active instance:

      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 IBM MQ and apply maintenance to the IBM MQ server.
3. When you have completed the maintenance update, restart all the queue managers on the IBM MQ server, permitting standby instances:

   Use the following command:

   ```
   strmqm -x QmgrName
   ```
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:

   Use the **endmqm** `-shutdown_option -s QMgrName` command, and the restart the instances using the **strmqm** `-x QmgrName` command.

Windows Applying and removing maintenance on Windows

Maintenance tasks associated with IBM MQ on Windows are grouped in this section.

Procedure

- To apply maintenance level server updates, see "Applying maintenance level server updates on Windows" on page 47.
- To apply maintenance level client updates, see "Applying maintenance level client updates on Windows" on page 58.
- To remove updates and revert to the previous maintenance level using the Windows installer, see "Removing maintenance level server updates on Windows" on page 54.
- For information on how to use multiple installations of IBM MQ on the same server to control the release of maintenance fixes, see "Staging maintenance level updates on Windows" on page 59.
- For information on how to use use multi-instance queue managers to reduce the outage caused by applying maintenance updates, see "Applying maintenance level updates to multi-instance queue managers on Windows" on page 52.

Related tasks

"Applying maintenance to IBM MQ" on page 8
Maintenance is the application of a reversible fix. Any changes to queue manager data are compatible with the previous code level.

Windows Applying and removing maintenance level server updates on Windows

On Windows, you can apply maintenance level server updates to IBM MQ either interactively or silently. You can also remove updates from a server and revert to the previous maintenance level of IBM MQ. You can use multi-instance queue managers to reduce the outage caused by applying maintenance updates.

Windows Applying maintenance level server updates on Windows

You can apply maintenance level server updates to IBM MQ for Windows either interactively or silently.

Before you begin

1. If you are running on a server with multiple IBM MQ installations, you must identify the installation. Make sure that the commands you enter run against the correct installation; see **setmqenv**.
2. Download the fix pack from Fix Central. For more information and download links, go to Downloading IBM MQ 9.1 and select the tab for the Long Term Support version that you require.

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.

Procedure

1. Log on as an Administrator.
2. Stop all applications using the IBM MQ installation.

   If you use the 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. Stop the mqweb server that is associated with the IBM MQ installation:

   a) Check whether the mqweb server is running by entering the following command:

   ```
   dspmqweb status
   ```

   b) Stop the mqweb server by entering the following command:

   ```
   endmqweb
   ```
4. End all the activity of queue managers associated with the IBM 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
   ```

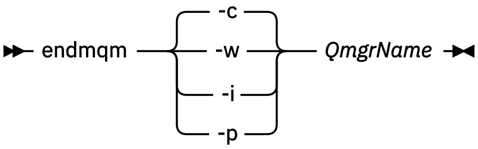
   `dspmq -o installation -o status` displays the installation name and status of queue managers associated with all installations of IBM MQ.

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

   b) Use the MQSC command **`DISPLAY LSSTATUS`** to list the status of listeners associated with a queue manager, as shown in the following example:

   ```
   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 MQ libraries they have loaded. If they do not, you must find another way to force applications to release IBM 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 MQ. The application is not informed about queue managers in the current installation being shut down.

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

Note: "Applying maintenance level updates to multi-instance queue managers on Windows" on page 52 describes how to apply maintenance to a multi-instance queue manager. A multi-instance queue manager can continue to run on one server, while maintenance is applied to another server.

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

```
endmqlsr -m QMgrName
```

5. Stop the IBM MQ service for the installation.

   a) Right-click the **IBM MQ** icon in the taskbar > click **Stop IBM MQ**.

6. Load and apply the maintenance files for server 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 "7" on page 50

- 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 2 on page 49.

Table 2. 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"` If `MQPLOG` is not specified (which is the case if you start maintenance by clicking the **Apply fix pack n.n.n.n** icon in the IBM MQ program group) the log name used by default will be `amqicsdn.txt` in your TEMP directory (`%TEMP%`).
`MQPINSTALLATIONNAME`	*`Installation name`*	The name of the installation to maintain. If there is only one installation (of any level) on the machine, this argument can be safely omitted. If there is more than one 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 is applicable, then `amqicsdn.exe` issues error message AMQ4781 and ends.

Table 2. Properties used to install or uninstall a maintenance update (continued)

Property	Value	Description
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 installation 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 IBM MQ 9.0.0 Fix Pack 1: `9.0.0-IBM-MQ-Windows-FP0001.exe`.
- *responseFile* is the full path and name of the response file.

7. Optional: Apply the maintenance to other server installations on the system:

- Interactively:

  a. From the Windows start menu, select **Start> Programs > IBM 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 2 on page 49.

8. Optional: Uninstall the fix pack files from your machine.

   After installing the fix pack files and applying the maintenance to all the server installations that you want to update, you can either uninstall the fix pack files from your machine or leave them installed for future use.

   Note: Uninstalling these files does NOT remove them from the installations to which you have already applied the maintenance. If that is what you intend, you should instead follow the instructions in "Removing maintenance level server updates on Windows" on page 54.

   If you add any installable features at a later time, you must reapply the maintenance to update the added feature(s).

   - Interactively:
     - To uninstall the fix pack files, on the **Control Panel... Programs and Features** panel, select **IBM MQ (fix pack 9.1.0.x) files**.
     - Click **Uninstall**.
   - Silently:

     To silently remove the fix pack files, enter the following command from an admin command prompt:

     ```
     fixpack_files_installdir\_IBM MQ (fix pack V.R.M.F files)_installation\Change IBM MQ (fix
     pack V.R.M.F files) Installation.exe" -i silent
     ```

     where

     - *fixpack_files_installdir* is the location in which you chose to install the files. For IBM MQ 9.1.0 Fix Pack 7, for example, this might be: `C:\Program Files\IBM\source\MQ 9.1.0.7`.
     - *V.R.M.F* is the fix pack number. For IBM MQ 9.1.0 Fix Pack 7, for example, this is `9.1.0.7`.

   Note: The fix pack files contain a JRE, so if you chose not to install a JRE in your IBM MQ installation for local policy reasons, you may want to uninstall the fix pack files as soon as you have finished applying the update to your installation(s).

What to do next

On a server installation, you must restart the IBM MQ taskbar application manually after the maintenance application completes.

The IBM 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\bin64\amqmtbrn.exe -Startup
```

Related tasks

Stopping a queue manager

“Applying maintenance level updates to multi-instance queue managers on Windows” on page 52
On Windows platforms, you can use multi-instance queue managers to reduce the outage caused by applying maintenance updates.

Related reference

dspmq (display queue managers)

DISPLAY LSSTATUS

endmqm (end queue manager)

endmqlsr (end listener)

Windows *Applying maintenance level updates to multi-instance queue managers on Windows*

On Windows platforms, you can use multi-instance queue managers to reduce the outage caused by applying maintenance updates.

Before you begin

Before starting this task, read through the prerequisites described in *Before you begin* in “Applying maintenance level server updates on Windows” on page 47

Before starting this task, see that the maintenance is applied to the IBM MQ installation on a server and not to individual queue managers. Before you apply maintenance, you must stop all the queue managers, and any IBM MQ service, on a server.

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 the queue manager that you want to keep running is an existing single instance queue manager, you must convert it to a multi-instance queue manager. For prerequisites and guidance how to create a multi-instance queue manager, see Multi-instance queue managers.

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 update. 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

Follow these steps to apply maintenance to a multi-instance queue manager on Windows.

Procedure

1. Log on as an Administrator.
2. Stop all applications using the IBM MQ installation.

   If you use the 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. Find out the current state of the queue managers and their associated listeners associated with the IBM MQ installation.

   a) From the installation that you are updating, use the **dspmq** command to list the state of the queue managers:

   - To display the installation name and status of queue managers associated with all installations of IBM MQ, run the following command:

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

   - To display the status of active queue managers associated with the installation from which you are running the command, run the following command:

     ```
     dspmq -a
     ```

   b) Use the MQSC command **DISPLAY LSSTATUS** to list the status of listeners associated with a queue manager, as shown in the following example:

   ```
   echo "DISPLAY LSSTATUS(*) STATUS" | runmqsc QmgrName
   ```
4. Use the **endmqm** command to stop each running queue manager associated with this installation.

   - If the queue manager is running as standby, run the **endmqm** command to end the standby as shown in the following example:

     ```
     endmqm -x QMgrName
     ```

   - If the queue manager is running as the active instance, run the **endmqm** command to end the active instance and transfer control to the standby instance as shown in the following example:

     ```
     endmqm -shutdown_option -s QMgrName
     ```

     where *-shutdown_option* is an optional parameter specifying the type of shutdown. For more information about optional parameters for the **endmqm** command, see endmqm.

     If there is no standby instance running, and the command fails, start a standby instance on a different server.
   - If a queue manager is running as a single instance queue manager, stop the queue manager. In the case of a single queue manager you have no alternative but to stop the queue manager before applying the maintenance update. For more information about how to stop a queue manager, see Stopping a queue manager.

   Stop any listeners associated with the queue managers by using the **endmqlsr** command as shown in the following example:

   ```
   endmqlsr -m QMgrName
   ```

   After you complete this step, no queue manager instances are left running on the server that you intend to update.
5. Apply maintenance to the IBM MQ server.

   Follow the instructions in “Applying maintenance level server updates on Windows” on page 47, starting from Step 4.

6. When you have completed the maintenance update, use the **strmqm** command to restart all the queue managers on the IBM MQ server, permitting standby instances, as shown in the following example:

   ```
   strmqm -x QmgrName
   ```

7. Repeat the procedure on the standby server, to update its maintenance level.
8. If necessary, switch the active instances back to the primary servers:
   a) Stop the instances by using the **endmqm** command as shown in the following example:

      ```
      endmqm -shutdown_option -s QMgrName
      ```

   b) Restart the instances by using the **strmqm** command as shown in the following example:

      ```
      strmqm -x QmgrName
      ```

Related tasks

Stopping a queue manager

Related reference

dspmq (display queue managers)

DISPLAY LSSTATUS

endmqm (end queue manager)

endmqlsr (end listener)

strmqm (start queue manager)

Windows *Removing maintenance level server updates on Windows*

You can remove updates from a server and revert to the previous maintenance level of IBM MQ by using the Windows installer.

Before you begin

Important: This task describes how to remove maintenance level updates for an IBM MQ server. For information about how to remove maintenance level updates for an IBM MQ client, see “Reverting a client to an earlier maintenance level on Windows” on page 59.

The server fix pack is for updating installations installed from the server DVD or eImage only and is neither tested nor supported as an update mechanism for updating client installations from the MQC91 IBM MQ Clients SupportPac.

1. If you are running on a server with multiple IBM MQ installations, you must identify the installation. Make sure that the commands you enter run against the correct installation; see **setmqenv**.
2. 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 an IBM MQ server, you can restore the server to a previous level of maintenance.

Important: If you installed the server at a particular maintenance level, that is, a *Manufacturing Refresh*, you cannot restore IBM MQ to an earlier maintenance level.

Procedure

1. Log on as an Administrator.
2. Stop all applications using the IBM MQ installation.

If you use the 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 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
      ```

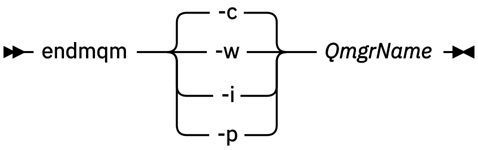
      dspmq -o installation -o status displays the installation name and status of queue managers associated with all installations of IBM MQ.

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

   b) Use the MQSC command **DISPLAY LSSTATUS** to list the status of listeners associated with a queue manager, as shown in the following example:

      ```
      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 MQ libraries they have loaded. If they do not, you must find another way to force applications to release IBM 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 MQ. The application is not informed about queue managers in the current installation being shut down.

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

      Note: "Applying maintenance level updates to multi-instance queue managers on Linux" on page 35 describes how to apply maintenance to a multi-instance queue manager. A multi-instance queue manager can continue to run on one server, while maintenance is applied to another server.

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

      ```
      endmqlsr -m QMgrName
      ```

4. Stop the IBM MQ service for the installation.

   a) Right-click the **IBM MQ** icon in the taskbar > click **Stop IBM MQ**.

5. Remove the maintenance interactively, or silently using a command.

   - Interactively:

a. For each installation of IBM MQ that has had maintenance applied, you are presented with one of the following icons in the Windows start menu:

i) **Start > Programs > IBM MQ > Remove Refresh Pack *V.R.M.L* (*installation_name*)**

ii) **Start > Programs > IBM MQ > Remove Fix Pack *V.R.M.L* (*installation name*)**

where

V is the version number
R is the release number
M is the modification number
L is the level of modification

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 3 on page 56.

Table 3. 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" If MQPLOG is not specified (which is the case if you start maintenance by clicking the **Apply fix pack n.n.n.n** icon in the IBM MQ program group) the log name used by default will be amqicsdn.txt in your TEMP directory (%TEMP%).
MQPINSTALLATIONNAME	*Installation name*	The name of the installation to maintain. If there is only one installation (of any level) on the machine, this argument can be safely omitted. If there is more than one 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 is 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 installation fails.

Table 3. Properties used to install or uninstall a maintenance update (continued)

Property	Value	Description
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 MQ (fix pack V.R.M.L files)_installation\Change IBM MQ
   (fix pack V.R.M.L 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 V.R.M.L`

   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 MQ taskbar application manually after the maintenance application completes.

The IBM 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\bin64\amqmtbrn.exe -Startup
   ```

Related tasks

Stopping a queue manager

"Applying maintenance level server updates on Windows" on page 47

You can apply maintenance level server updates to IBM MQ for Windows either interactively or silently.

Related reference

dspmq (display queue managers)

DISPLAY LSSTATUS

endmqm (end queue manager)

endmqlsr (end listener)

Windows Updating the maintenance level of a client on Windows

On Windows, you can apply maintenance level client updates to IBM MQ either interactively or silently. You can also revert to the previous maintenance level of IBM MQ.

Before you begin

V 9.1.0 From IBM MQ 9.1, separate client eImages are not available for downloading from Passport Advantage. A separate client DVD is also not available. Instead, you can either obtain the client eImage from inside the main IBM MQ server eImage, or server DVD, which include the server and client. Alternatively, you can download the IBM MQ client components from Fix Central. Follow the links in Resource adapter, clients and other resources.

Windows *Applying maintenance level client updates on Windows*

You can apply maintenance level updates to IBM MQ for Windows clients either interactively or by performing a silent MSI update.

About this task

You can apply maintenance to an IBM MQ client interactively by running the `setup.exe` program, or you can perform a silent MSI update by running the **`msiexec`** command with file `IBM MQ.msi`. The `setup.exe` and `IBM MQ.msi` files are both located in the `\Windows\MSI\` directory of the client installation image.

Important: The server fix pack is for updating installations installed from the server DVD or eImage only and is neither tested nor supported as an update mechanism for updating client installations from the IBM MQ client components downloaded from Fix Central.

Procedure

- Access the IBM MQ installation image.

  See Where to find downloadable installation images.

Interactive client update

- On the client installation image, navigate to the `\Windows\MSI\` directory, then run the `setup.exe` program.

Silent client update

- From the command line, use the command **`msiexec`** to perform a silent MSI update.
  - To update a computer with only a single installation, use a command similar to the following example:

    ```
    msiexec /i "InstallationImage\Windows\MSI\IBM MQ.msi" /l*v install_log_path
    /q TRANSFORMS="1033.mst" REINSTALL=ALL REINSTALLMODE=vomus
    ```

  - To update a single client on a multi installation computer with multiple clients, use a command similar to the following example:

```
msiexec /i "InstallationImage\Windows\MSI\IBM MQ.msi" /l*v install_log_path
/q TRANSFORMS=":InstanceId2.mst;1033.mst" REINSTALL=ALL REINSTALLMODE=vomus
```

Results

When the maintenance completes you can query the maintenance level by running the **dspmqver** command. For more details, see “Applying maintenance to IBM MQ” on page 8.

Windows *Reverting a client to an earlier maintenance level on Windows*

For a client, the way in which you revert to an earlier maintenance level depends on whether your client installation was installed from the server DVD or eImage, or from the IBM MQ client components downloaded from Fix Central.

About this task

If your client installation was installed from the IBM MQ server DVD or eImage, you can revert to an earlier maintenance level by rolling back the fixpack installation.

Important: Client only installations installed from the IBM MQ client components downloaded from Fix Central cannot be restored to an earlier maintenance level on Windows. If you need to do this, you must first uninstall the currently installed version of the client and then reinstall client at the appropriate earlier maintenance level.

Procedure

To revert to an earlier maintenance level for a client installation that you installed from the server DVD or eImage:

- Log on as an Administrator.
- Remove the maintenance either interactively or silently (see “Removing maintenance level server updates on Windows” on page 54).

To revert to an earlier maintenance level for a client only installation that you installed from the IBM MQ client components downloaded from Fix Central:

- Uninstall the current version of the client as described in Uninstalling IBM MQ on Windows.
- Reinstall the client at the earlier maintenance level.

Windows Staging maintenance level updates on Windows

On Windows systems, you can use multiple installations of IBM MQ on the same server to control the release of maintenance level updates.

Before you begin

The steps in this task are based on an example scenario in which it is assumed that you have two copies of IBM MQ named `Inst_1` and `Inst_2`, and a number of applications and two queue managers, `QM1` and `QM2`, running on a server. To set up your configuration for this scenario, complete the following steps:

1. Install two copies of IBM MQ. 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 MQ Explorer.
6. Set up remote connections to all the queue managers in each instance of IBM MQ Explorer.

Note: From 1Q 2023, for Multiplatforms, there are two types of maintenance:

- Fix packs, which contain roll-ups of all defects fixed since the previous fix pack delivery (or GA). Fix packs are produced exclusively for Long Term Support (LTS) releases during their normal support lifecycle.
- Cumulative security updates (CSUs), which are smaller updates and contain security patches released since the previous maintenance (GA). CSUs are produced for LTS releases (including releases in extended support), and also for the latest IBM MQ Continuous Delivery (CD) release, as required to deliver relevant security patches.

For maintenance releases in or after 1Q 2023, the fourth digit in the VRMF represents either a fix pack number or a CSU number. Both types of maintenance are mutually cumulative (that is, they contain everything included in older CSUs and fix packs), and both are installed using the same mechanisms for applying maintenance. Both types of maintenance update the F-digit of the VRMF to a higher number than any previous maintenance: fix packs use "F" values divisible by 5, CSUs use "F" values not divisible by 5.

For maintenance releases before 1Q 2023, the fourth digit in the VRMF always represents the fix pack level. For example, the first fix pack of the IBM MQ 9.1.0 LTS release is numbered 9.1.0.1.

For more information, see Changes to IBM MQ's maintenance delivery model.

About this task

You can install multiple copies of IBM MQ on a server to stage the release of maintenance level updates. For example, as in the scenario that is described in the task steps, by using two installations to roll out maintenance level updates, you maintain two maintenance levels on a server, with the aim of getting all queue managers and applications to the production maintenance level before replacing the previous level of maintenance 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 or cumulative security updates (CSUs).

In the example, an application links to the primary installation. When it connects to a queue manager, IBM 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 98.

For applications built with the link options described in the product documentation, the simplest way to configure the link library search path for IBM MQ applications is to make an installation primary. Only if it is important to pick up a fix in the IBM MQ link library itself, must you review the search path. Either you must make the installation with the IBM MQ link library fix primary, or make a local adjustment for the application, perhaps by running the **`setmqenv`** command. See “Migrating IBM MQ library loading to a later version on Windows” on page 134.

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 MQ Explorer and you have two installations, you also have two IBM MQ Explorer instances. One linked to one installation, and one to the other. Each IBM MQ Explorer shows locally connected queue managers that are associated with the same installation as the instance of IBM MQ Explorer. To monitor all the queue managers on a server, set up remote connections to the queue managers associated with the other installations.

Procedure

Apply the first maintenance level update to `Inst_2`.

1. Download the first fix pack or cumulative security update (CSU) when it is released.

   See Where to find downloadable installation images.

2. Apply the fix pack or cumulative security update (CSU) you downloaded to `Inst_2`.

   See "Applying maintenance level server updates on Windows" on page 47.

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.

   Note: "Applying maintenance level updates to multi-instance queue managers on Windows" on page 52 describes how to apply maintenance to a multi-instance queue manager. A multi-instance queue manager can continue to run on one server, while maintenance is applied to another server.

   b) Set up the local environment to the installation `Inst_2` by using the **setmqenv** command:

   ```
   "Inst_2_INSTALLATION_PATH\bin\setmqenv" -s
   ```

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

   c) Associate the queue manager with `Inst_2` by using the **setmqm** command:

   ```
   setmqm -m QM1 -n Inst_2
   ```

   d) Start `QM1` by using the **strmqm** command:

   ```
   strmqm QM1
   ```

   e) Repeat substeps c and d for `QM2`.

   f) Set up IBM MQ Explorer for `Inst_2`.

      i) Start the `Inst_2` instance of IBM MQ Explorer

      Tip: On Windows, hover over the IBM MQ icon in the system tray. The hover help shows the installation name associated with the IBM MQ Explorer instance.

      ii) Click **IBM 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 by using the **setmqinst** command:

   ```
   "Inst_2_INSTALLATION_PATH\bin\setmqinst" -i -n Inst_2
   ```

Apply the second maintenance level update to `Inst_1`.

6. Download the next fix pack or cumulative security update (CSU) for the version of your product when it is released.

   See Where to find downloadable installation images.

7. Apply the fix pack or cumulative security update (CSU) that you have just downloaded to `Inst_1`.

   See "Applying maintenance level server updates on Windows" on page 47.

8. Verify `Inst_1`.

9. Transfer queue managers to `Inst_1` one at a time.

   a) Follow the procedure in step "4" on page 61.

   Replacing `Inst_2` by `Inst_1` in the instructions.

10. Set `Inst_1` primary by using the **setmqinst** command:

   ```

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

For subsequent maintenance fixes, alternate between `Inst_2` and `Inst_1`.

11. Alternate between repeating steps “1” on page 60 to “5” on page 61 for `Inst_2` and steps “6” on page 61 to “10” on page 61 for `Inst_1`.

Related concepts

“Queue manager coexistence” on page 95
Queue managers, with different names, can coexist on any server as long as they use the same IBM MQ installation. On z/OS, 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 98
You can install multiple copies of IBM MQ for UNIX, Linux, and Windows on the same server. These IBM MQ copies can be at the same or different version levels. This is called a multi-installation. Multi-installation is particularly useful when you upgrade from one IBM MQ version to a later version, because it allows you to run the earlier version alongside the later version.

Related tasks

Installing IBM MQ server on Windows
Associating a queue manager with an installation
Changing the primary installation

z/OS Applying and removing maintenance on z/OS

You can install maintenance to update IBM MQ for z/OS to a new maintenance level.

Before you begin

This task assumes that you understand the difference between Long Term Support and Continuous Delivery releases, and the maintenance delivery model that applies in each case. For more information, see IBM MQ Release Types and versioning.

The command level of a queue manager is the three-digit VRM level. An IBM MQ program can call `MQINQ`, passing the `MQIA_COMMAND_LEVEL` selector, to get the command level of the queue manager it is connected to.

About this task

IBM MQ for z/OS uses PTFs for Maintenance.

LTS PTFs are specific to a particular set of libraries corresponding to specific release level. For UNIX System Services features (that is, JMS and WEB UI, Connector Pack, and Managed File Transfer) the z/OS PTFs are aligned directly with the Multiplatforms fix packs and cumulative security updates (CSUs). These fixes are cumulative, and are made available at the same time as the equivalent Multiplatforms fix pack or CSU.

CD CD CSUs are not usually made available between CD releases, but are included in the next IBM MQ for z/OS CD release. You can also contact support to request a ++USERMOD.

Other fixes on IBM MQ for z/OS are distinct fixes on particular parts. These fixes resolve specific issues, are not cumulative, and are made available as and when they are produced.

PTFs are applied to and removed from the installed code using SMP/E. For more information about SMP/E, see Installing IBM MQ for z/OS and z/OS installation overview and the appropriate Program Directory. For download links for the Program Directories, see IBM MQ for z/OS Program Directory PDF files.

Apart from any exceptions documented with the PTFs, PTFs do not change the correct operation of IBM MQ, and are reversible. Nonetheless, you must check that the fixes have not changed the operation of critical programs unexpectedly.

Applying PTFs does not change the version, release, or maintenance level of the code. No queue manager migration is required after applying maintenance. PTFs are grouped into Recommended Service Updates

(RSUs) that have been tested together in a Consolidated Service Test (CST); see Consolidated Service Test and the RSU.

PTFs that apply to a category of software fixes might be grouped together and identified using a fix category. For more information, see IBM Fix category values and descriptions.

Procedure

1. Open Downloading IBM MQ 9.1.
2. Click the CD releases tab.

   Note that 9.1.5 was the final CD release for IBM MQ 9.1, and you are instructed to upgrade to CD releases in IBM MQ 9.2.
3. To access the latest 9.1 LTS downloads, click the "Latest LTS fix" tab.

   From this tab you can download either an LTS fix pack or an LTS CSU, whichever is the latest.

   a) Find the section **Downloading the fix pack/CSU for z/OS**.

   b) Get the specified fix pack/CSU PTFs from Shopz.

   Fix packs and CSUs are cumulative. Therefore you are only offered the latest fix, which might be either a fix pack or a CSU.

Related concepts

IBM MQ release types and versioning

“Queue manager coexistence” on page 95

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

Applying WebSphere Liberty interim fixes to the mqweb server

You can apply WebSphere Liberty interim fixes to the WebSphere Liberty that runs the mqweb server.

About this task

This task applies to both IBM MQ Long Term Support and IBM MQ Continuous Delivery.

Important: If an interim fix is not available for the WebSphere Liberty version that is installed in the IBM MQ installation, then you should contact IBM Support. You should NOT look to change the WebSphere Liberty version that is used by the IBM MQ Console and REST API and should only look to apply a WebSphere Liberty interim fix.

Note: This information does NOT apply to IBM MQ Appliance.

Procedure

1. Check the Liberty version.

   To do this, run the following command:

   ```
   <MQ_INSTALLATION_PATH>/web/bin/productInfo version --ifixes
   ```
2. Use the security link or the information on the page for the Liberty APAR to locate the correct archive interim fix (iFix) for the version that is installed.

   Liberty archive interim fixes come in a JAR format, and have an associated readme file that you can refer to for installation instructions. Download both files into a temporary directory.
3. After the interim fix has been downloaded, start a console and navigate to the directory that contains the interim fix JAR file.
4. Stop the mqweb server by using the command:

```
<MQ_INSTALLATION_PATH>/bin/endmqweb
```

5. As an administrative user, run the following command to install the interim fix:

   ```
   java -jar <iFix JAR> -installLocation <MQ_INSTALLATION_PATH>/web
   ```

6. Run the following command and check the output to confirm that the interim fix has been installed correctly:

   ```
   <MQ_INSTALLATION_PATH>/web/bin/productInfo version --ifixes
   ```

7. Restart the mqweb server by using the command:

   ```
   <MQ_INSTALLATION_PATH>/bin/strmqweb
   ```

Results

When the mqweb server restarts, the interim fix should be loaded.

Example

The following example shows how to apply a WebSphere Liberty interim fix for APAR PH31442 to an IBM MQ 9.1.0.8 installation on Linux.

1. Run the following command to check the version of Liberty installed with IBM MQ 9.1.0.8:

   ```
   /opt/mqm/web/bin/productInfo version --ifixes
   ```

   This command generates the following output, which indicates that the Liberty version is 21.0.0.3:

   ```
   Product name: WebSphere Application Server
   Product version: 21.0.0.3
   Product edition: BASE
   ```

2. Go to the web page for APAR PH31442.
3. In the **Download Package** section of the web page, click the download link for the archive **21003-wlp-archive-IFPH34122**.
4. After you have been redirected to Fix Central, download the following files into a temporary directory:
   - `21003-wlp-archive-IFPH34122-ReadMe.txt`
   - `21003-wlp-archive-ifph34122.jar`
5. Start a console, and navigate to the temporary directory.
6. Stop the mqweb server by using the command:

   ```
   /opt/mqm/bin/endmqweb
   ```

7. Next, run the following command as a root user to install the interim fix:

   ```
   java -jar 21003-wlp-archive-ifph34122.jar --installLocation /opt/mqm/web
   ```

   All being well, you should see the following output:

   ```
   Applying fix to Liberty install directory at /opt/mqm/web now.
   lib/com.ibm.ws.ui.tool.javaBatch_1.0.50.cl210320210319-1444.jar
   lib/com.ibm.ws.ui.tool.explore_1.0.50.cl210320210319-1444.jar
   lib/com.ibm.ws.ui_1.0.50.cl210320210319-1444.jar
   Fix has been applied successfully.
   Successfully extracted all product files.
   ```

8. Run the following command to check that the interim fix has been installed correctly:

   ```
   /opt/mqm/web/bin/productInfo version --ifixes
   ```

   You should see the following output:

   ```
   Product name: WebSphere Application Server
   Product version: 21.0.0.3
   ```

```
Product edition: BASE

PH34122 in the iFix(es): [21003-wlp-archive-IFPH34122]
```

9. Restart the mqweb server by using the command:

   ```
   /opt/mqm/bin/strmqweb
   ```

Related tasks
Contacting IBM Support
Related reference
endmqweb (end mqweb server)
strmqweb (start mqweb server)

Upgrading IBM MQ

Upgrading is the process of taking an existing IBM MQ installation and upgrading to a new level of code.

Before you begin

This task assumes that you understand the difference between Long Term Support and Continuous Delivery releases, and the maintenance delivery model that applies in each case. For more information, see IBM MQ Release Types and versioning.

About this task

When you upgrade from one release to another, or apply maintenance, including fix packs, cumulative security updates (CSUs), or interim fixes, the impact of the change depends on the extent of the change in the VRMF level:

- The term *upgrade* applies to increasing the version V, release R, or modification M level.
- The term *fix* applies to increasing the fix F level.

At each change of the V, R, or M level, the command level on the queue manager changes. On a change to the F level, the command level does not change.

Multi On Multiplatforms, after an upgrade has been applied, the only way to back out a VRM change is by taking one of the following actions:

- Uninstalling the product code and reinstalling the code.
- Installing the old level of code alongside the existing code and using the setmqm command to associate the queue manager with the other installation.

The general rule, is that if you have carried out an install that causes the command level of the new installation to be updated, and started the queue manager, you cannot back out the changes.

z/OS On z/OS, you can install new releases of IBM MQ to upgrade IBM MQ to a new VRM level. For more information, see “Upgrade and migration of IBM MQ on z/OS ” on page 197.

Procedure

1. Open Downloading IBM MQ 9.1.
2. Click the "CD releases" tab.

   Note that 9.1.5 was the final CD release for IBM MQ 9.1, and you are instructed to upgrade to CD releases in IBM MQ 9.2.
3. To access the latest 9.1 LTS downloads, click the "Latest LTS fix" tab.

   From this tab you can download the latest LTS base install level, and either an LTS fix pack or an LTS CSU, whichever is the latest.

The format of the download is platform-specific. For Multiplatforms you download one or more parts from Passport Advantage or Fix Central; For z/OS you download PTFs from Shopz.

Note: Fix Pack 9.1.0.9 and earlier also included fixes for the IBM MQ Appliance. However, support for the IBM MQ 9.1 Appliance firmware ended in September 2021, and subsequent fix packs do not include firmware updates for IBM MQ 9.1 on the Appliance. For the IBM MQ Appliance support schedule, see https://www.ibm.com/support/pages/lifecycle/search/?q=mq%20appliance.

a) To get the latest LTS base install level for Multiplatforms, find the section **Downloading the release from Passport Advantage (IBM MQ for Multiplatforms)**.

b) To get the latest fix pack or CSU:

- Multi Find the section **Downloading the fix pack/CSU from Fix Central (IBM MQ for Multiplatforms)**
- z/OS Find the section **Downloading the fix pack/CSU for z/OS**

Fix packs and CSUs are cumulative. Therefore you are only offered the latest fix, which might be either a fix pack or a CSU.

Related tasks

"Applying maintenance to IBM MQ" on page 8
Maintenance is the application of a reversible fix. Any changes to queue manager data are compatible with the previous code level.

Related reference

"OPMODE on z/OS" on page 196
In earlier versions of IBM MQ for z/OS the availability of new functions and backward migration was controlled by the **OPMODE** parameter in the CSQ6SYSP macro. In IBM MQ 9.1.0 this no longer applies, and the **OPMODE** parameter is ignored.

Windows Upgrading an IBM MQ installation on Windows

To upgrade an IBM MQ server installation on Windows, from one version, release, and modification level to a later one, you can use either the Launchpad or msiexec. To upgrade a client installation, you can use either the GUI installer or msiexec.

About this task

Before you begin, ensure that you have backed up your data.

Important: If you want to apply maintenance instead, for example, from IBM MQ 9.0.0.0 to IBM MQ 9.0.0 Fix Pack 1, see "Applying and removing maintenance on Windows" on page 47.

Procedure

- To upgrade a server installation, see "Upgrading an IBM MQ server installation using the Launchpad" on page 66 or "Upgrading an IBM MQ server installation using msiexec" on page 68.
- To upgrade a client installation, see "Upgrading an IBM MQ client installation using the GUI installer" on page 69 or "Upgrading an IBM MQ client installation using msiexec" on page 70.

Windows Upgrading an IBM MQ server installation using the Launchpad

How you upgrade an IBM MQ server installation on Windows to a newer version, release, or modification, using the Launchpad.

Before you begin

Ensure that you have:

1. Stopped all your IBM MQ applications

2. Shut down your listeners
3. Stopped all your queue managers
4. Backed up your data

Important: If you want to apply maintenance instead, for example, from IBM MQ 9.1.0 to IBM MQ 9.1.0 Fix Pack 1, see "Applying and removing maintenance on Windows" on page 47.

Procedure

1. Access the IBM MQ installation image.

   The location might be the mount point of the server DVD, a network location, or a local file system directory. See Where to find downloadable installation images.
2. Locate `setup.exe` in the base directory of the IBM MQ installation image.
   - From a DVD, this location might be *E:*`\setup.exe`
   - From a network location, this location might be *m:*`\`*instmqs*`\setup.exe`
   - From a local file system directory, this location might be *C:*`\`*instmqs*`\setup.exe`
3. Start the installation process.

   Either run `setup.exe` from a command prompt, or double-click `setup.exe` from Windows Explorer.

   Note: If you are installing on a Windows system with UAC enabled, accept the Windows prompt to allow the launchpad to run as elevated. During installation, you might also see **Open File - Security Warning** dialog boxes that list International Business Machines Limited as the publisher. Click **Run** to allow the installation to continue.

   The IBM MQ Installation window is displayed.
4. Follow the instructions on screen.
5. Select **Installing a new instance**, if you see a panel asking you to choose between installing a new instance, or maintaining or upgrading an existing instance, when you click the **Launch IBM MQ Installer** button.

   You use the other option when adding or removing features from an already installed IBM MQ.
6. On the next panel, choose between **Install leaving the existing installation(s) untouched** or **Upgrade an existing named installation already on the machine**, and click **Next**.

   ⚠ **Attention:** If you do not see this screen, it means that there was no IBM MQ server installation on the machine that could be upgraded by this installer.
7. Follow the installer prompts to upgrade your IBM MQ server installation.

Related tasks

"Upgrading an IBM MQ server installation using msiexec" on page 68
How you upgrade an IBM MQ server installation on Windows to a newer version, release, or modification, using msiexec.

"Upgrading an IBM MQ client installation using the GUI installer" on page 69
How you upgrade an IBM MQ client installation on Windows to a newer version, release, or modification, using the GUI installer.

"Upgrading an IBM MQ client installation using msiexec" on page 70

How you upgrade an IBM MQ client installation on Windows to a newer version, release, or modification, using msiexec.

Windows Upgrading an IBM MQ server installation using msiexec

How you upgrade an IBM MQ server installation on Windows to a newer version, release, or modification, using msiexec.

Before you begin

Ensure that you have:

1. Stopped all your IBM MQ applications
2. Shut down your listeners
3. Stopped all your queue managers
4. Backed up your data

Important: If you want to apply maintenance instead, for example, from IBM MQ 9.1.0 to IBM MQ 9.1.0 Fix Pack 1, see "Applying and removing maintenance on Windows" on page 47.

Procedure

1. Access the IBM MQ installation image.

   The location might be the mount point of the server DVD, a network location, or a local file system directory. See Where to find downloadable installation images.
2. Locate `IBM MQ.msi` in the MSI directory of the IBM MQ installation image.
   - From a DVD, this location might be *E:*`\MSI\IBM MQ.msi`
   - From a network location, this location might be *m:\instmqs*`\MSI\IBM MQ.msi`
   - From a local file system directory, this location might be *C:\instmqs*`\MSI\IBM MQ.msi`
3. Optional: If you are upgrading the only IBM MQ server installation, where the installation has the default value `Installation1` issue the following command:

   ```
   msiexec /i "InstallationImage\MSI\IBM MQ.msi" /q AGREETOLICENSE=YES
   INSTALLATIONNAME="Installation1"
   ```
4. Optional: If you are upgrading an installation on a machine that already has one or more IBM MQ server installations of the level you are upgrading to, you must provide additional parameters to select a free MSI instance ID.

   See Choosing MSI Instance IDs for multiple server installations for more information.

   In this case, the command might look something like this:

   ```
   msiexec /i "InstallationImage\MSI\IBM MQ.msi" /q AGREETOLICENSE=YES
   INSTALLATIONNAME="Installation2" NEWINSTANCE=1
   TRANSFORMS=":InstanceId2.mst;1033.mst"
   ```

Related tasks

"Upgrading an IBM MQ server installation using the Launchpad" on page 66
How you upgrade an IBM MQ server installation on Windows to a newer version, release, or modification, using the Launchpad.

"Upgrading an IBM MQ client installation using the GUI installer" on page 69
How you upgrade an IBM MQ client installation on Windows to a newer version, release, or modification, using the GUI installer.

"Upgrading an IBM MQ client installation using msiexec" on page 70

How you upgrade an IBM MQ client installation on Windows to a newer version, release, or modification, using msiexec.

Windows Upgrading an IBM MQ client installation using the GUI installer

How you upgrade an IBM MQ client installation on Windows to a newer version, release, or modification, using the GUI installer.

Before you begin

Ensure that you have:

1. Stopped all your IBM MQ applications
2. Shut down your listeners
3. Stopped all your queue managers
4. Backed up your data

Procedure

1. Access the IBM MQ installation image.

   See Where to find downloadable installation images.
2. Locate `setup.exe` in the `Windows` directory of the IBM MQ installation image.
3. Start the installation process.

   Either run `setup.exe` from a command prompt, or double-click `setup.exe` from Windows Explorer.

   Note: If you are installing on a Windows system with UAC enabled, accept the Windows prompt to allow the launchpad to run as elevated. During installation, you might also see **Open File - Security Warning** dialog boxes that list International Business Machines Limited as the publisher. Click **Run** to allow the installation to continue.

   The IBM MQ Installation window is displayed.
4. Follow the instructions on screen. When you click the **Launch IBM MQ Installer** button, if you see a panel asking you to choose between installing a new instance, or maintaining or upgrading an existing instance, select **Installing a new instance**.
5. On the next panel, choose between **Install leaving the existing installation(s) untouched** or **Upgrade an existing named installation already on the machine**, and click **Next**.

   ⚠ **Attention:** If you do not see this screen, it means that there was no IBM MQ client installation on the machine that could be upgraded by this installer.
6. Follow the installer prompts to upgrade your IBM MQ client installation.

Related tasks

“Upgrading an IBM MQ client installation using msiexec” on page 70
How you upgrade an IBM MQ client installation on Windows to a newer version, release, or modification, using msiexec.

“Upgrading an IBM MQ server installation using the Launchpad” on page 66
How you upgrade an IBM MQ server installation on Windows to a newer version, release, or modification, using the Launchpad.

“Upgrading an IBM MQ server installation using msiexec” on page 68

How you upgrade an IBM MQ server installation on Windows to a newer version, release, or modification, using msiexec.

Windows Upgrading an IBM MQ client installation using msiexec

How you upgrade an IBM MQ client installation on Windows to a newer version, release, or modification, using msiexec.

Before you begin

Ensure that you have:

1. Stopped all your IBM MQ applications
2. Shut down your listeners
3. Stopped all your queue managers
4. Backed up your data

Procedure

1. Access the IBM MQ installation image.

   See Where to find downloadable installation images.
2. Locate `IBM MQ.msi` in the `Windows\MSI` directory of the IBM MQ installation image.
3. Optional: If you are upgrading the only IBM MQ client installation, where the installation has the default value `Installation1` issue the following command:

   ```
   msiexec /i "InstallationImage\Windows\MSI\IBM MQ.msi" /l*v install_log_path
   /q TRANSFORMS="1033.mst" REINSTALL=ALL REINSTALLMODE=vomus
   ```
4. Optional: If you are upgrading an installation on a machine that already has one or more IBM MQ client installations of the level you are upgrading to, you must provide additional parameters to select a free MSI instance ID.

   See Choosing MSI Instance IDs for multiple client installations for more information.

   In this case, the command might look something like this:

   ```
   msiexec /i "InstallationImage\MSI\IBM MQ.msi" /q AGREETOLICENSE=YES
   INSTALLATIONNAME="Installation2" NEWINSTANCE=1
   TRANSFORMS=":InstanceId2.mst;1033.mst"
   ```

Related tasks

"Upgrading an IBM MQ client installation using the GUI installer" on page 69
How you upgrade an IBM MQ client installation on Windows to a newer version, release, or modification, using the GUI installer.

"Upgrading an IBM MQ server installation using the Launchpad" on page 66
How you upgrade an IBM MQ server installation on Windows to a newer version, release, or modification, using the Launchpad.

"Upgrading an IBM MQ server installation using msiexec" on page 68

How you upgrade an IBM MQ server installation on Windows to a newer version, release, or modification, using msiexec.

MQ Adv. Linux Upgrading RDQM (replicated data queue managers)

You can upgrade RDQM when applying a fixpack by using the supplied upgrade script.

About this task

An upgrade script is supplied with the fixpack image at `*MQ_FixpackDriver*/Advanced/RDQM/`. You must run the script as `root`:

`updateRDQMsupport`

This script upgrades the latest kmod-DRBD packages and latest pacemaker packages. kmod-DRBD is the kernel module for DRBD. If you apply an OS upgrade such that the version of the Linux kernel that is being used changes, you must upgrade the DRBD kernel module to match the new version by running `updateRDQMsupport`.

The script does not update IBM MQ.

You can run your queue managers while upgrading by failing over to the other nodes in the RDQM configuration in turn and upgrading one at a time. For advice on the order in which to update HA RDQM nodes, see "Migrating HA RDQMs" on page 229, for advice on the order in which to update DR RDQM nodes, see "Migrating DR RDQMs" on page 230, for advice on the order in which to update DR/HA RDQM nodes, see "Migrating DR/HA RDQMs" on page 232.

Procedure

- To upgrade HA RDQM support, on each node:
  a) Suspend the HA group. To suspend the HA group on the node, enter the following command:

     ```
     rdqmadm -s
     ```

  b) Run the script `updateRDQMsupport` to upgrade DRBD and Pacemaker.
  c) Install the MQSeriesRDQM package that was provided with the fix pack.
  d) Resume the HA group. To resume the HA group on the node, enter the following command:

     ```
     rdqmadm -r
     ```

- To upgrade DR RDQM support:
  a) Upgrade the recovery node by running the `updateRDQMsupport` to upgrade DRBD.
  b) Install the MQSeriesRDQM package that was provided with the fix pack.
  c) Stop the queue manager running on the primary node using the **endmqm** command. (If required, start the queue manager on the recovery node.)
  d) Run the script `updateRDQMsupport` to upgrade DRBD.
  e) Install the MQSeriesRDQM package that was provided with the fix pack.
  f) If you started the queue manager on the recovery node, end it.
  g) Start the queue manager on the primary node by using the **`strmqm`** command.
- To upgrade DR/HA RDQM support:
  a) On each node of the HA group on the recovery site, complete the following steps:
     a. Suspend the HA group. To suspend the HA group on the node, enter the following command:

        ```
        rdqmadm -s
        ```

     b. Run the script `updateRDQMsupport` to upgrade DRBD and Pacemaker.
     c. Install the MQSeriesRDQM package that was provided with the fix pack.

d. Resume the HA group. To resume the HA group on the node, enter the following command:

```
rdqmadm -r
```

b) Stop the DR/HA RDQMs on the main site, or perform a managed failover to the recovery site.

c) On each node of the HA group on the main site, complete the following steps:

a. Suspend the HA group. To suspend the HA group on the node, enter the following command:

```
rdqmadm -s
```

b. Run the script `updateRDQMsupport` to upgrade DRBD and Pacemaker.

c. Install the MQSeriesRDQM package that was provided with the fix pack.

d. Resume the HA group. To resume the HA group on the node, enter the following command:

```
rdqmadm -r
```

d) Restart the DR/HA RDQMs, or fail them back to the main site.

Note:

You might see the following warning if kmod-DRBD is installed or upgraded on a system that is using kabi-yum-plugins:

```
WARNING: possible kABI issue with package: kmod-drbd
```

The warning indicates that kmod-DRBD uses some symbols that are not on the allowlist. Provided that the DRBD kernel module matches the Linux kernel that is being used, this warning can be ignored.

Related reference

rdqmadm (administer replicated data queue manager cluster)
endmqm (end queue manager)
strmqm (start queue manager)

Migrating IBM MQ

Migration is the conversion of programs and data to work with a new code level of IBM 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.

Before you begin

Before upgrading your IBM MQ installation or migrating your queue managers, you must read “Changes that affect migration” on page 73 to identify what migration tasks you must plan for.

About this task

Whenever you upgrade IBM MQ to a new release that changes its command level, migration is performed by the queue manager. Whenever you upgrade IBM 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.

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 a queue manager on Windows” on page 116; see also “Changes that affect migration” on page 73.

Multi On IBM MQ for Multiplatforms, you cannot easily revert to a previous level of IBM MQ after installation. If you install a copy of IBM MQ obtained from Passport Advantage or from physical media, the installer uninstalls IBM MQ, if it is present. It then installs the new level of IBM MQ. To revert to the previous level of IBM 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 MQ. (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.) To restore IBM 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 MQ.

LTS z/OS On IBM MQ for z/OS it is possible to backwards migrate from a Long Term Support (LTS) release. For more information, see "Backward migration to earlier supported releases of IBM MQ for z/OS" on page 203.

z/OS CD Backwards migration is not supported for a Continuous Delivery (CD) release on z/OS.

Related concepts

"Multi-installation queue manager coexistence on UNIX, Linux, and Windows" on page 98
You can install multiple copies of IBM MQ for UNIX, Linux, and Windows on the same server. These IBM MQ copies can be at the same or different version levels. This is called a multi-installation. Multi-installation is particularly useful when you upgrade from one IBM MQ version to a later version, because it allows you to run the earlier version alongside the later version.

"Queue manager coexistence" on page 95
Queue managers, with different names, can coexist on any server as long as they use the same IBM MQ installation. On z/OS, UNIX, Linux, and Windows, different queue managers can coexist on the same server and be associated with different installations.

Related tasks

Backing up and restoring a queue manager

Changes that affect migration

Changes to the product might affect the migration of a queue manager from an earlier release to the current release of IBM MQ, or affect existing applications or configurations. Review these changes before upgrading queue managers to the latest product version and decide whether you must plan to make changes to existing applications, scripts, and procedures before starting to migrate your systems.

Changes in the current release

For details of changes in the current release, including those that affect migration, see the following information:

- V 9.1.0 What's new and changed in IBM MQ 9.1
- CD What's new and changed in IBM MQ 9.1.x Continuous Delivery
- LTS What's changed in IBM MQ 9.1.0 Long Term Support
- LTS CD IBM MQ, WebSphere MQ, and MQSeries product readmes

Changes in earlier versions

For information about what changed in an earlier release of the product, see the *What's changed* section in the product documentation for that version of the product:

- IBM MQ 9.0

    - What's new and changed in IBM MQ 9.0.0
    - What's new and changed in IBM MQ 9.0.x Continuous Delivery
    - What's changed in IBM MQ 9.0.0.x Long Term Support
- What's changed in IBM MQ 8.0
- What's changed in IBM WebSphere MQ 7.5
- What's changed in IBM WebSphere MQ 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 IBM MQ.

Restrictions on reversing queue manager migration

⚠ **Attention:**

Multi On IBM MQ for Multiplatforms, you cannot reverse queue manager migration to remove the effect of changes. This restriction applies whether your enterprise uses the Long Term Support (LTS) release or Continuous Delivery (CD) release model.

z/OS LTS On IBM MQ for z/OS, you can reverse queue manager migration as long as you have migrated from an earlier LTS release to a more recent LTS release, and the earlier LTS release had the backwards migration PTFs for the more recent LTS release installed before the migration started.

For further information, see IBM MQ release types.

Related concepts

"Migration concepts and methods" on page 76
An overview of the various concepts and methods for migrating from one release of the product to another.

"Migration considerations for IBM MQ 8.0 or later on Windows" on page 113

"Program and data directory locations on Windows" on page 115
The installation location for IBM MQ program binary and data files on Windows depends on the IBM MQ version you are installing, and whether this is the first time IBM MQ is being installed.

Considerations when migrating from Advanced Message Security 7.0.1

From IBM WebSphere MQ 7.5 onwards, Advanced Message Security became a component of IBM MQ. Some aspects of AMS functionality changed, that might affect applications, administrative scripts, or management procedures.

Important:

If you are migrating to IBM MQ 9.1 from IBM WebSphere MQ 7.5 or earlier, you must first migrate to an interim version. See Migration paths.

Review the following list of changes carefully before upgrading queue managers to IBM WebSphere MQ 7.5 or later. Decide whether you must plan to make changes to existing applications, scripts, and procedures before starting to migrate systems:

- From IBM WebSphere MQ 7.5, AMS installation is a part of the IBM MQ installation process.
- The AMS security capabilities are enabled with its installation and controlled with security policies. You do not need to enable interceptors to allow AMS to start intercepting data.
- AMS in IBM MQ does not require the use of the **`cfgmqs`** command as in the stand-alone version of Advanced Message Security.

Migration paths

An overview of the migration paths between different IBM MQ versions. For some IBM MQ for z/OS migration paths, you can revert to the version you were using prior to migration. For IBM MQ for Multiplatforms, you cannot easily revert to a previous version.

Note: Before each new version of the product is released, it is tested for migration from earlier versions that are in support at that time. Direct migration from a version that is out of support might also work, but is neither tested nor supported. Therefore, to migrate to the latest version from a version that is out of support, you should first migrate to an interim version that was released before the earlier version went out of support.

- “Migration paths: IBM MQ for Multiplatforms” on page 75
- “Migration paths: IBM MQ for z/OS” on page 75
- “Detailed migration information” on page 76

Migration paths: IBM MQ for Multiplatforms

Multi

You can migrate from IBM MQ 8.0 or later direct to IBM MQ 9.1. To migrate from IBM WebSphere MQ 7.5, you must first migrate to IBM MQ 8.0 or IBM MQ 9.0.

IBM i For IBM MQ on IBM i there was no IBM WebSphere MQ 7.5 so the appropriate version to use is IBM WebSphere MQ 7.1.

For IBM MQ for UNIX, Linux, and Windows, Side-by-side and Multi-stage migration can be used to migrate from one release to a later release.

Table 4. Migration paths: IBM MQ for Multiplatforms

Migration from	To IBM MQ 9.1	To IBM MQ 9.0	To IBM MQ 8.0
IBM MQ 9.0	Yes	-	-
IBM MQ 8.0	Yes	Yes	-
IBM WebSphere MQ 7.5	No	Yes	Yes

For IBM MQ for Multiplatforms, you cannot easily revert to a previous version. However, if a queue manager has not been started, you can uninstall the current version and reinstall a different version of IBM MQ. It does not matter what versions of IBM MQ are installed between when a queue manager was last started and when it is next started.

Migration paths: IBM MQ for z/OS

z/OS

You can migrate from IBM MQ 8.0 or later direct to IBM MQ 9.1.

If you are migrating to IBM MQ 9.1 from IBM WebSphere MQ 7.1, you must first migrate to IBM MQ 8.0 or IBM MQ 9.0 and restart each queue manager with OPMODE=(NEWFUNC,800) or OPMODE=(NEWFUNC,900). For more information, see “Migrating from earlier unsupported releases of IBM MQ for z/OS ” on page 198.

Table 5. Migration paths: IBM MQ for z/OS

Migration from	To IBM MQ 9.1	To IBM MQ 9.0	To IBM MQ 8.0
IBM MQ 9.0	Yes	-	-
IBM MQ 8.0	Yes	Yes	-
IBM WebSphere MQ 7.1	No	Yes	Yes

To revert to a previous version, see "Reverting a queue manager to a previous version on z/OS" on page 215.

Detailed migration information

For detailed information about migrating to the current version, see the following links:

- Windows "Planning to migrate IBM MQ to a later version on Windows" on page 112
- Linux UNIX "Planning to migrate IBM MQ to a later version on UNIX and Linux" on page 146
- IBM i "Planning to migrate IBM MQ to a later version on IBM i" on page 174
- z/OS "Planning to migrate IBM MQ to a later version on z/OS" on page 200

For migrating to an IBM MQ version other than the current version, see the documentation for the target version:

- Migrating IBM MQ (IBM MQ 9.0)
- IBM MQ migration (IBM MQ 8.0)
- Multi IBM WebSphere MQ migration (IBM WebSphere MQ 7.5)

For out of service IBM MQ versions, where the documentation is no longer available in the online IBM Documentation, you can instead download the documentation for offline use. See Documentation for older versions of IBM MQ.

Related concepts

"Migration concepts and methods" on page 76
An overview of the various concepts and methods for migrating from one release of the product to another.

Migration concepts and methods

An overview of the various concepts and methods for migrating from one release of the product to another.

Objects to consider during migration

It is important to consider four types of object during migration:

Operating environment migration

Upgrading the operating environment, or components in the environment such as installing a new level of JRE; see " IBM MQ operating environment migration" on page 79.

Queue manager migration

Migrating a queue manager following an upgrade of the IBM MQ installation to a new command level; see "Queue manager migration" on page 80.

When migrating queue managers that are members of a cluster, do full repositories before partial repositories. This is because an older repository cannot store newer attributes introduced in a newer release. It tolerates them, but does not store them.

IBM MQ MQI client migration

Migrating a client configuration following installation of a new version or release of the IBM MQ MQI client ; see "IBM MQ MQI client migration" on page 81.

It is better to migrate the clients after the queue managers that they communicate with have been migrated.

Application migration

Relinking, recompiling, or recoding an IBM MQ server or client application; see “Application migration and interoperation” on page 82. Application migration also includes migrating any API or channel exits.

Use the new version of the libraries to build the applications, once the queue managers have been upgraded.

Impact of migration on other queue managers or clients

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

Compatibility, coexistence, and interoperability

See “Coexistence, compatibility, and interoperability” on page 93 for information about the compatibility of IBM MQ applications connected to queue managers and IBM MQ MQI client clients on different command levels. The section also explains the concept of queue manager coexistence, and the interoperability of IBM MQ JMS applications with WebSphere Application Server.

Queue manager clusters

Can a queue manager cluster contain queue managers at different command levels? See “Migrating a queue manager cluster” on page 219 to answer this question, and how to migrate a cluster of queue managers.

z/OS **Queue sharing groups**

Queue sharing groups involve multiple queue managers running on z/OS. How do you migrate queue managers that are part of a queue sharing group to a new command level; see “Queue sharing group migration” on page 218.

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 “Migrating a queue manager in a high-availability configuration” on page 225, which covers both migration of multi-instance queue managers, and the migration of queue managers operating in high-availability clusters.

IBM MQ application migration model

Figure 1 on page 78 shows the various components of the application migration model.

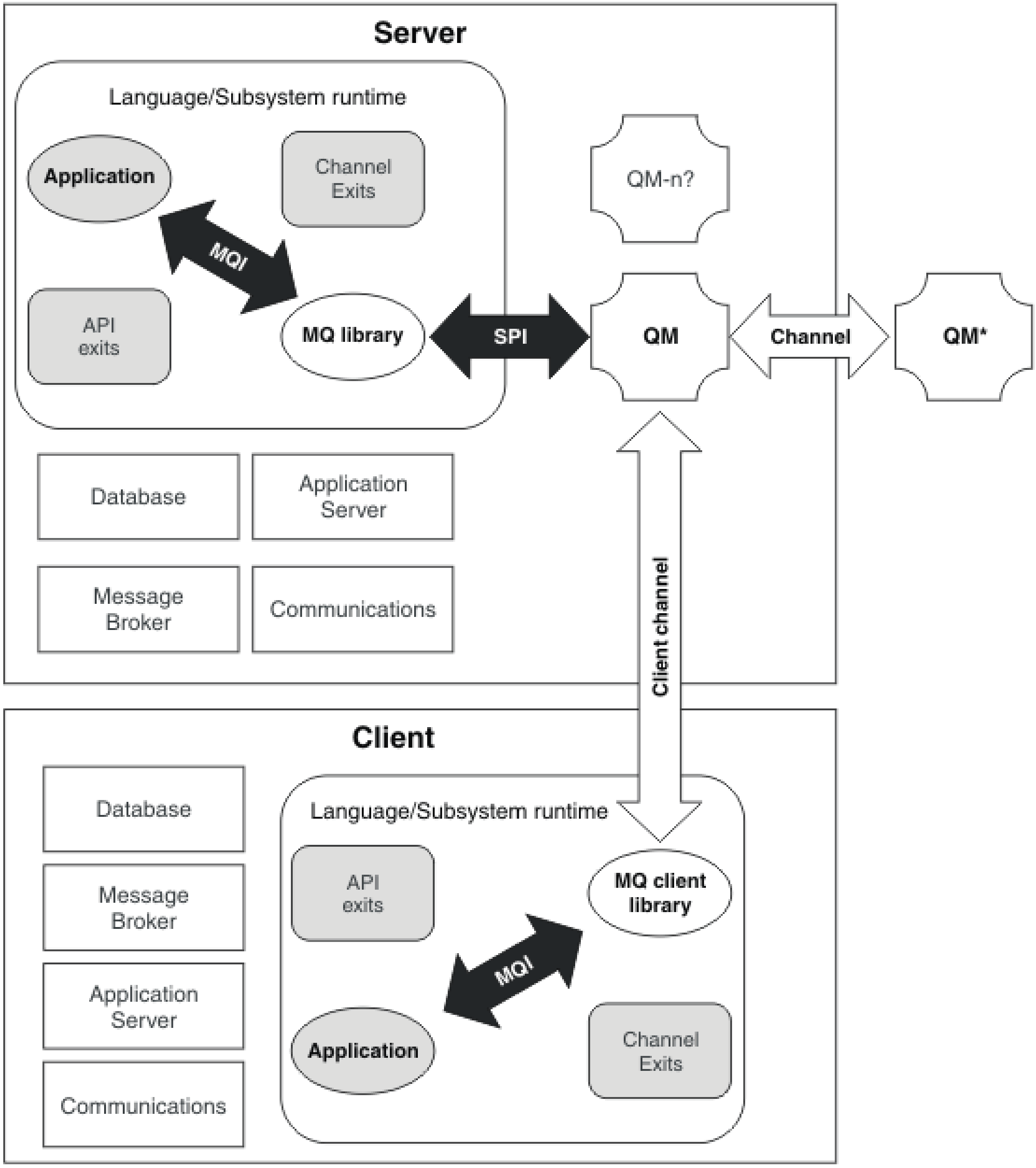


Figure 1. IBM MQ application migration model

This diagram shows two runtime operating system environments, each of which contains a number of software components, such as databases, application servers, and the language or subsystem run time environment. One environment is called `Server`, and contains an IBM MQ server and server application. The other environment is called `Client`, and contains an IBM MQ MQI client application.

The language or subsystem run time environment contains an IBM MQ application, the IBM MQ MQI client or server library, and IBM MQ channel and API exit programs.

The server environment has one or more queue managers, represented in the diagram by `QM`, that are using the installation of IBM MQ installed on the server. The components of the language or subsystem run time environment are connected to queue manager `QM`, either locally in the server, or remotely from the client.

The application is linked to the IBM MQ library by the MQI. The libraries are shown linked to the queue manager QM either by an SPI, which describes the connection between the process running the MQI and the queue manager processes, or by an IBM MQ MQI client connection.

The diagram also shows two more queue managers:

- The queue manager labeled QM*, which represents queue managers of various levels installed on other servers.
- The queue manager labeled QM-n?, which represents a number of queue managers that coexist on the same server as queue manager QM, but are running at a different release level. The installations for these different release levels are not shown in the diagram. The question-mark in the queue manager name QM-n? indicates that this capability might not be present in your environment.

Multiple releases of IBM MQ installed in the same operating environment are called *coexistent*. It is not necessary, but it is usual, for coexistent installations to be at different release levels. 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. You must consider any configuration implications of running some applications or clusters on the same server with queue managers at different command levels.

For more information, see "Queue manager coexistence" on page 95.

IBM MQ operating environment migration

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

To find out what operating environment upgrades you must make before upgrading IBM MQ, compare the requirements for different releases. For more information about system requirements, see System Requirements for IBM MQ. By selecting the appropriate link on the web page, the SPCR tool enables you to go directly to the following information for the specific operating system, or systems, that your enterprise uses.

- Supported operating systems
- Prerequisites
- System requirements
- Optional supported software

For details about operating environment changes in the latest release that directly affect the migration to a new version of IBM MQ, see the following information:

- V 9.1.0 What's new and changed in IBM MQ 9.1.0
- CD What's new and changed in IBM MQ 9.1.x Continuous Delivery
- LTS What's changed in IBM MQ 9.1.0 Long Term Support
- LTS CD IBM MQ, WebSphere MQ, and MQSeries product readmes

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:

- What's changed in IBM MQ 8.0
- What's changed in IBM WebSphere MQ 7.5
- What's changed in IBM WebSphere MQ 7.1
- For older versions of products, including IBM WebSphere MQ 7.0.1, where documentation is provided outside of IBM Documentation, see Documentation for older versions of IBM MQ.

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

Queue manager migration

After upgrading an installation, queue manager migration might be required. Migration takes place when you start a queue manager. You can remove an upgrade before you have started a queue manager. However, if you remove the upgrade after a queue manager has been started, the queue manager will not work.

Migrating a queue manager to a later release

z/OS On IBM MQ for z/OS, queue manager migration is required after upgrading to a different version, release, or maintenance level of the product. The upgrade changes the command level. The current command, or VRM, level is shown in the z/OS console log.

Multi On IBM MQ for Multiplatforms, 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. 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.

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 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.

When migrating queue managers that are members of a cluster, migrate full repositories before partial repositories. This is because an older repository cannot store newer attributes introduced in a newer release. It tolerates them, but does not store them.

Restoring a queue manager to an earlier release

Multi For IBM MQ for Multiplatforms, you cannot restore a queue manager to an earlier release level after you have migrated it to a new release. You must back up your system before starting backwards migration. You can either back up queue manager data, or use a backup queue manager; see Backing up and restoring IBM MQ. Before backing up, you must stop the queue manager.

z/OS V 9.1.0 For IBM MQ for z/OS, the following considerations apply to migration:

- LTS You can backwards migrate queue managers only if you are using the Long Term Support (LTS) release model. For more information, see IBM MQ release types.
- From IBM MQ 9.1.0 the OPMODE parameter has been removed. An IBM MQ 9.1.0 queue manager can always be backwards migrated to the release it was previously running at, if the release was one of IBM MQ 8.0.0 or IBM MQ 9.0.0. You must have applied compatibility PTFs to the earlier release before starting a queue manager at the new release.

Related concepts

“Upgrade and migration of IBM MQ on z/OS ” on page 197
You can install new releases of IBM MQ to upgrade IBM MQ to a new release, or version level. Multiple installations at the same or different levels can coexist on the same z/OS instance. Running a queue manager at a higher level requires migration.

Related tasks

“Migrating a queue manager on UNIX and Linux” on page 147
The procedures for migrating a queue manager to a later version of the product, and for restoring a queue manager to an earlier version of the product are detailed in this section.

“Migrating a queue manager on Windows” on page 116
The procedures for migrating a queue manager to a later version of the product, and for restoring a queue manager to an earlier version of the product are detailed in this section.

“Migrating a queue manager to the latest version on IBM i” on page 176
Follow these instructions to migrate a queue manager on IBM i to the latest MQ version.

“Moving a queue manager to a different operating system” on page 232
Follow these instructions to move a queue manager from one operating system to another. Note that this is **not** a migration of a queue manager.

Related reference

“OPMODE on z/OS” on page 196
In earlier versions of IBM MQ for z/OS the availability of new functions and backward migration was controlled by the **OPMODE** parameter in the CSQ6SYSP macro. In IBM MQ 9.1.0 this no longer applies, and the **OPMODE** parameter is ignored.

IBM MQ MQI client migration

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

It is better to migrate the clients after the queue managers that they communicate with have been migrated.

Client migration on the client workstation is optional and manual. Client migration on the server is required and automatic. You must upgrade an IBM 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 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.

An IBM MQ MQI client can interoperate with earlier and later versions of IBM MQ. Upgrading the IBM MQ MQI client makes new function available to client applications, and is important to maintain the service level. Migrating an IBM MQ MQI client gives it access to new configuration options.

The IBM 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 MQ 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 MQ" on page 104

"Application compatibility and interoperability with later versions of IBM MQ" on page 105
IBM 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 IBM MQ to a queue manager running at a later version of IBM MQ.

Related tasks

"Migrating an IBM MQ MQI client to the latest version on IBM i" on page 190
Before migrating an IBM MQ MQI client, create a migration plan. Stop all IBM MQ activity on the client workstation. Upgrade the IBM MQ MQI client installation. Make any essential configuration and application changes.

"Migrating an IBM MQ MQI client on UNIX and Linux" on page 161
Before migrating an IBM MQ MQI client, create a migration plan. Stop all IBM MQ activity on the client workstation. Upgrade the IBM MQ MQI client installation. Make any essential configuration and application changes.

"Migrating an IBM MQ MQI client on Windows" on page 132
Before migrating an IBM MQ MQI client, create a migration plan. Stop all IBM MQ activity on the client workstation. Upgrade the IBM MQ MQI client installation. Make any essential configuration and application changes.

Related reference

"MQ clients: Client Channel Definition Table (CCDT)" on page 107
You can connect an IBM MQ client application to any level of queue manager. If a 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.

"Supported IBM MQ client: Default behavior of client-connection and server-connection channels" on page 108

Application migration and interoperation

IBM MQ supports running applications compiled and linked against previous versions of IBM MQ, with later levels of IBM MQ. Use the new version of the libraries to build the applications, once the queue managers have been upgraded.

To migrate an application to run with a new level of IBM 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 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 MQ. The suggested extent of testing varies depending on the extent of the changes in IBM MQ; see "Characteristics of different types of upgrade on z/OS" on page 197 or "Characteristics of different types of upgrade" on page 7.

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 73 to see if any changes might affect your applications.

3. Changes that are not required, but that you might want to make in future, perhaps because you have a 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 MQ.

Do not load IBM MQ libraries from an earlier level. IBM 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 Windows platforms, the application load path must be set up to the location of the IBM MQ server libraries. You do not have to recompile and relink an application. Applications compiled and linked against an earlier version of IBM MQ can load libraries from a later version.

Multi On Multiplatforms, IBM 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 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 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 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 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 IBM MQ 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 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.

Connection authentication

For a new IBM MQ 8.0, or later, installation, the **CONNAUTH CHCKLOCL** attribute will be set to `OPTIONAL`. This means that user IDs and passwords are not required, but if they are provided they must be a valid pair, or they will be rejected.

When you are migrating between a previous version of IBM MQ and the latest version, the **CONNAUTH CHCKLOCL** attribute on each queue manager is set to `NONE`, ensuring version to version continuity, but switching connection authentication off.

For more information see Connection authentication: Configuration.

Related concepts

"Application compatibility and interoperability with earlier versions of IBM MQ" on page 104

"Application compatibility and interoperability with later versions of IBM MQ" on page 105

IBM 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 IBM MQ to a queue manager running at a later version of IBM MQ.

Related tasks

"Migrating IBM MQ library loading to a later version on UNIX and Linux" on page 164
On UNIX and Linux, no change in the way IBM MQ libraries are loaded is normally required if you upgrade from an earlier version of the product to a later version by replacing an earlier version of the product with the later version, based on the single stage scenario. However, if you choose to take advantage of multi-installation in the later version of the product, based on the side-by-side or multi-stage migration scenarios, you might have to configure the runtime environment differently, for the operating system to load the later version of the IBM MQ library.

"Migrating IBM MQ library loading to a later version on Windows" on page 134
On Windows, no change in the way IBM MQ libraries are loaded is normally required if you upgrade from an earlier version of the product to the later version by replacing an earlier version of the product with the later version, based on the single stage scenario. However, if you choose to take advantage of multi-installation in the later version of the product, based on the side-by-side or multi-stage migration scenarios, you might have to configure the runtime environment differently, for the operating system to load the later version of the IBM MQ library.

Multi Migration methods on IBM MQ for Multiplatforms

There are three main methods of migrating from one release to another: Single-stage migration (called a slip installation on IBM i), side-by-side migration, and multi-stage migration. Multi-stage migration is not an option for IBM i.

Important:

If you are migrating to IBM MQ 9.1 from IBM WebSphere MQ 7.5 or earlier, you must first migrate to an interim version. See Migration paths.

Single-stage migration

Single-stage migration is the term that is used to describe replacing the only installation of IBM MQ on a server, with a later release.

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 later version, automatically. Queue managers are automatically associated with the installation on the later 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 later version to be the primary installation, commands such as **`strmqm`** work without providing an explicit path to the command.

Of the three approaches, single-stage migration preserves the greatest number of existing scripts and procedures for running IBM MQ. However, the other migration approaches support a gentler transition to the new version, which can reduce the overall impact on users.

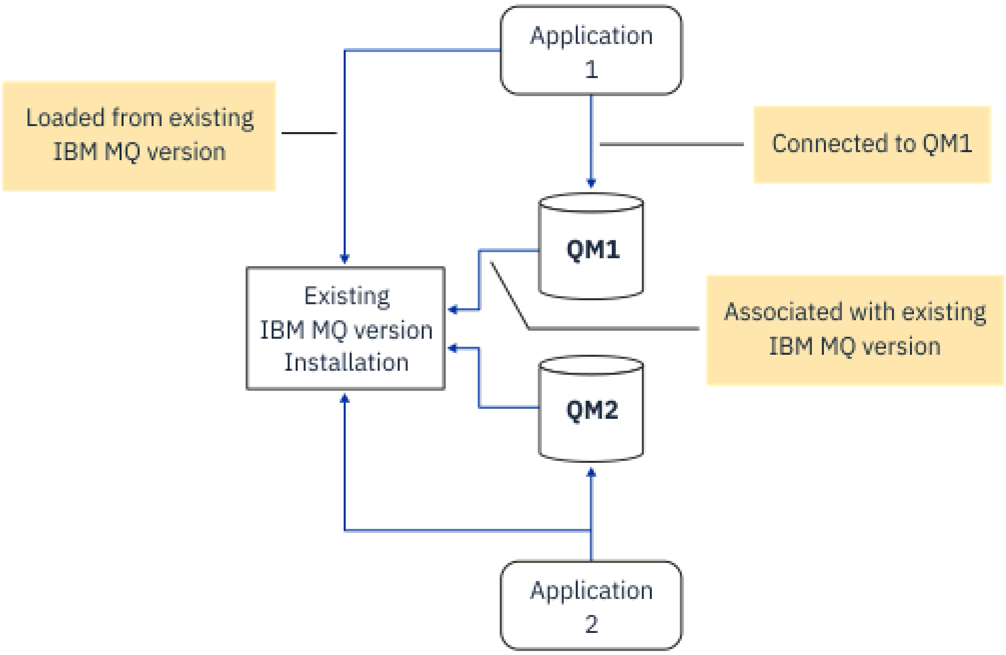

Application 1
Loaded from existing IBM MQ version
Connected to QM1
QM1
Existing IBM MQ version Installation
Associated with existing IBM MQ version
QM2
Application 2

Figure 2. Single_stage migration: earlier version installed with connected queue managers and associated applications

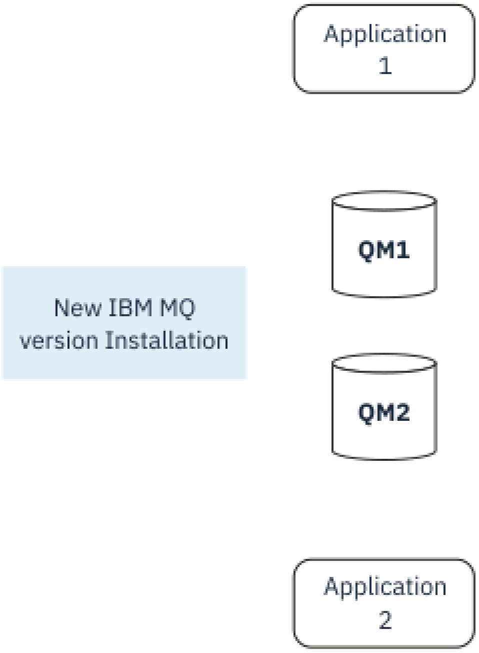


Figure 3. Single_stage migration: later version installed but queue managers not yet connected and applications not yet associated

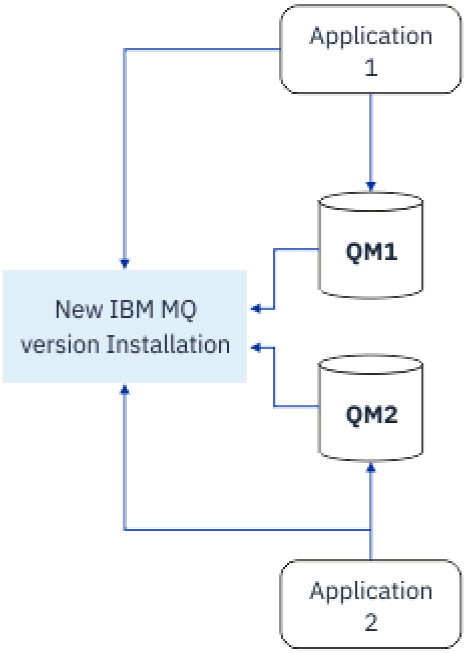


Figure 4. Single_stage migration: migrated queue managers connected to and applications associated with later version

For more information about single-stage migration, see:

- Linux UNIX "Migrating on UNIX and Linux: single-stage" on page 150
- Windows "Migrating on Windows: single stage" on page 120
- IBM i "Installation methods on IBM i" on page 177 (on IBM i, a single-stage migration is called a slip installation)

Side-by-side migration

On UNIX, Linux and Windows, side-by-side migration is the term that is used to describe installing a later version of IBM MQ alongside an older version on the same server. The side-by-side migration scenario sits half-way between the single-stage and multi-stage migration scenarios and is based on the following premise:

- Install additional IBM MQ code alongside existing installation while queue managers are still running.
- Move queue managers one at a time to the new installation.
- Migrate and test applications one at a time.

During the installation and verification of the later version of IBM MQ, queue managers continue running, and remain associated with the older version of IBM MQ.

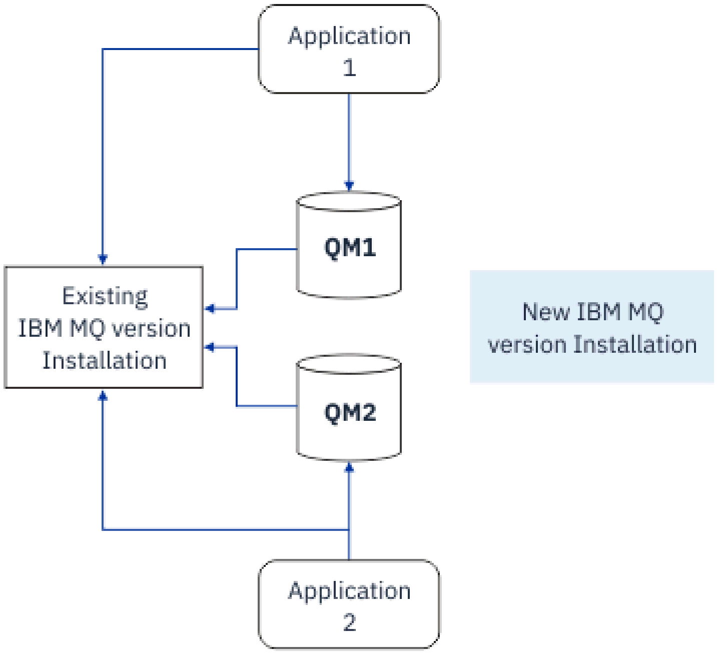


Figure 5. Side-by-side migration: later version installed but queue managers still connected to and applications still associated with earlier version

When you decide to migrate queue managers to the later version of IBM MQ, you stop all queue managers, migrate them all to the later version, and uninstall the earlier version of IBM MQ.

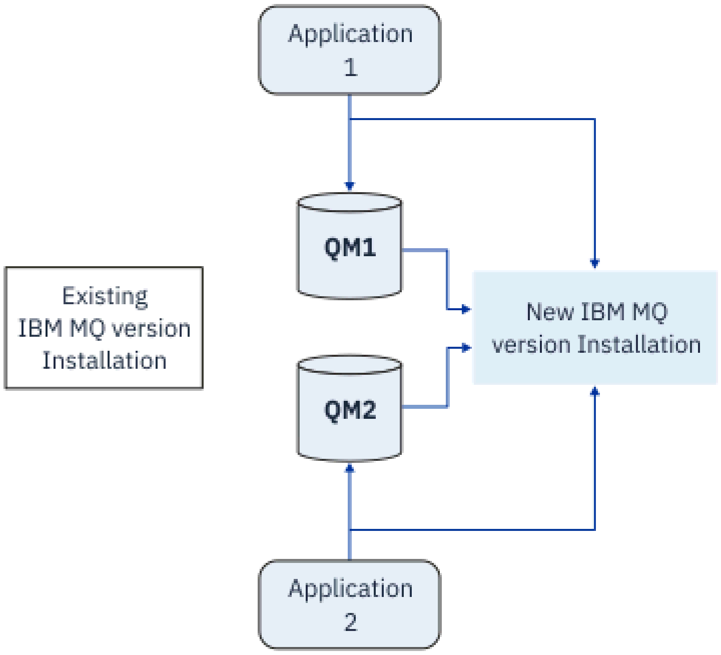


Figure 6. Side-by-side migration: migrated queue managers connected to and applications associated with later version

The advantage that the side-by-side migration has over the single-stage migration is that you can install and verify the later IBM MQ installation on the server before you switch over to it.

Although side-by-side migration is less flexible than multi-stage migration, it does have some advantages over the multi-stage approach. With the side-by-side approach, you can assign a later version of IBM MQ to be the primary installation. With the multistage approach, and one version of IBM MQ set as the primary installation, many applications restart without having to reconfigure their environment, as IBM MQ commands work without providing a local search path.

For more information about side-by-side migration, see:

- Linux UNIX “Migrating on UNIX and Linux: side-by-side” on page 153
- Windows “Migrating on Windows: side-by-side” on page 124

Note: IBM i Side-by-side migration has a different meaning on IBM i. A side-by-side installation upgrades IBM MQ on a different computer. For more information, see “Installation methods on IBM i” on page 177. Multiple installations are not applicable to IBM i.

Multi-stage migration

ULW

Multi-stage migration is the term that is used to describe running a later version of IBM MQ alongside an older version on the same server. Multi-stage migration is the most flexible approach.

After you install the later version alongside the earlier version, you can create new queue managers to verify the installation of the later version, and develop new applications. At the same time, you can migrate queue managers and their associated applications from the earlier version to the later version. By migrating queue managers and applications one by one, you can reduce the peak workload on your staff who are managing the migration.

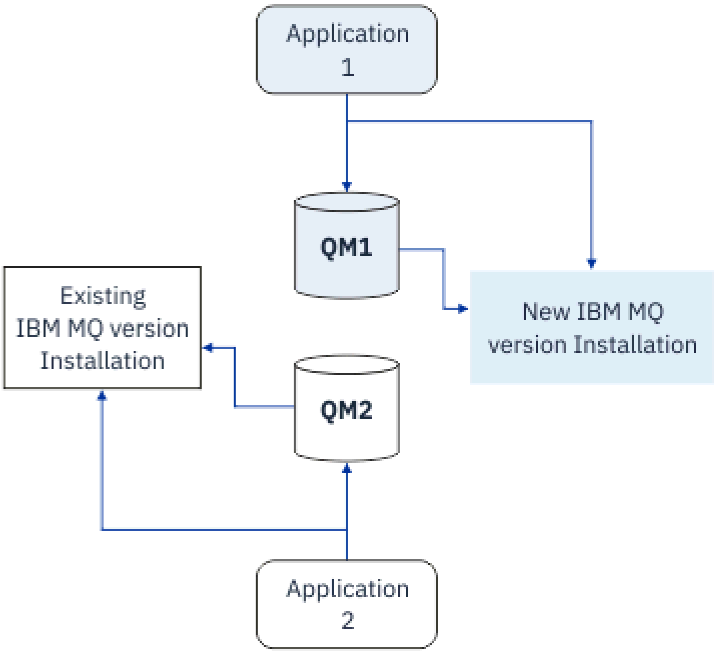

Application 1
QM1
Existing IBM MQ version Installation
New IBM MQ version Installation
QM2
Application 2

Figure 7. Multi-stage migration: one queue manager and application migrated to later version, and another queue manager and application still at earlier version

For more information about multi-stage migration, see:

- Linux UNIX “Migrating on UNIX and Linux: multi-stage” on page 156
- Windows “Migrating on Windows: multi-stage” on page 127

Related concepts

“Upgrade and migration of IBM MQ on z/OS ” on page 197

You can install new releases of IBM MQ to upgrade IBM MQ to a new release, or version level. Multiple installations at the same or different levels can coexist on the same z/OS instance. Running a queue manager at a higher level requires migration.

ULW Primary installation on UNIX, Linux and Windows

On UNIX, Linux, and Windows, which support multiple installations of IBM MQ, you can optionally define one installation as the primary installation. The primary installation is the one to which IBM MQ system-wide locations refer.

Overview

You can install multiple versions of the product on UNIX, Linux, and Windows, and configure one of these installations as the primary installation. The primary installation is:

- The installation to which system-wide locations refer
- Optional, but convenient

UNIX and Linux

Linux UNIX

The primary installation:

- Has symbolic links in `/usr/lib` and `/usr/bin`

  If you have not set a primary installation there are no symbolic links.
- Must be configured manually using the following command:

```
$ MQ_INSTALLATION_PATH/bin/setmqinst -i -p MQ_INSTALLATION_PATH
```

To locate your various installations, you can:

- Use the platform installation tools to query what is installed and where on the system
- Use the dspmqver command to display IBM MQ version and build information.
- Use the dspmqinst command to display installation entries from `mqinst.ini`.
- Use the following command to list the installations:

```
cat /etc/opt/mqm/mqinst.ini
```

Windows

Windows

The primary installation is:

- By default the first installation.
- Pointed to by global environment variables.
- Used by some operating system features that require central registration of interface libraries.

  For example, .NET monitor (transactional mode) and COM/ActiveX interface classes.

To locate your various installations, you can use the:

- Use the platform installation tools to query what is installed and where on the system
- Use the dspmqver command to display IBM MQ version and build information.
- Use the dspmqinst command to display installation entries from `mqinst.ini`.
- Use the following command to query the registry:

```
reg.exe query "HKLM\Software\[Wow6432Node\]IBM\WebSphere MQ\Installation" /s
```

Migration of Windows registry information

Windows

IBM WebSphere MQ 7.1 onwards uses `mqs.ini` and `qm.ini`.

Related concepts

Primary installation

ULW z/OS Multiple IBM MQ installations

Multiple IBM MQ installations are supported on UNIX, Linux, and Windows. This gives you the option to install and select between one or more IBM MQ installations.

Overview

You can select between:

- Simplicity of maintaining a single IBM MQ installation.
- Flexibility, by allowing up to a maximum of 128 IBM MQ installations on a system.

You can install multiple copies of the same code level; this is especially convenient for maintenance purposes.

LTS For example, if you want to upgrade IBM MQ 9.0.0.0 to IBM MQ 9.0.0 Fix Pack 1, you can install a second copy of IBM MQ 9.0.0.0, apply the maintenance to bring it to IBM MQ 9.0.0 Fix Pack 1, and then move the queue managers across to the new installation. You still have the original installation, so it is a simple matter to move the queue managers back if you encounter any problems.

Note that you can only move the queue manager to an installation at the same or a higher version. That is, you can move a queue manager in the following ways:

- From an earlier version to a later version, but not back. For example, from IBM MQ 9.0.0 to IBM MQ 9.1.0, but not from IBM MQ 9.1.0 to IBM MQ 9.0.0.
- From one fix pack level to another fix pack level at the same version, and back. For example, from IBM MQ 9.0.0.0 to IBM MQ 9.0.0 Fix Pack 1, and back to IBM MQ 9.0.0.0.

Notes:

1. Solaris Linux On Linux and Solaris only, you must ensure that each package installed has a unique name.

   You need to use a tool to create a unique set of packages:

   - `$ crtmqpkg PACKAGE_SUFFIX`
   - This takes the IBM MQ installation packages, and repackages them with a new name of your choice. You then install as usual.
2. All installations share a data directory; this is where `mqs.ini` is located for example.
3. All installations share the same namespace for queue managers. This means that you cannot create several queue managers of the same name in different installations.
4. IBM MQ installations are fully relocatable; each installation has a separate installation path. You can choose where you would like to install IBM MQ.
5. IBM MQ resources have installation-scope resource isolation, so operations on one installation do not affect the others.

This means that the resources created by one installation are isolated from those created by other installations. It enables actions, such as removing an installation of IBM MQ, while queue managers are running under another installation.

6. Queue managers are "associated" with an installation You can move them, but you cannot migrate data back to earlier releases.

Working with multiple installations

To work with a queue manager, you need to use the commands from its installation. If you select the wrong installation, you see:

```
AMQ5691: Queue manager 'MYQM' is associated with a different installation (Inst1)
```

To work with a queue manager, you have to use the control commands from its associated installation. You have a choice of:

- Using the full path to the control commands, for example:

  ```
  $ MQ_INSTALLATION_PATH\bin\strmqm MYQM
  ```

  or

- Setting the environment variables for an installation with one of:

  ```
  $ MQ_INSTALLATION_PATH/bin/setmqenv 's
  $ setmqenv -m MYQM
  $ setmqenv -n InstallationName
  $ setmqenv -p MQ_INSTALLATION_PATH
  ```

You might consider using a shell script or batch file to set up the environment for each IBM MQ installation. You can use the **setmqenv** or **crtmqenv** commands to help with this.

- setmqenv sets the values of the environment variables, such as PATH, CLASSPATH and LD_LIBRARY_PATH, for use with an IBM MQ installation.
- crtmqenv creates a list of the environment variables and their values for use with a particular IBM MQ installation. You can then use this list to incorporate into a shell script or batch file.

Commands

To run a command, the operating system must find the command in an IBM MQ installation. In general, you must run a command from the installation that is associated with the correct queue manager. IBM MQ does not switch commands to the correct installation. However, there are some exceptions, such as the **setmqinst** command, where you can run the command from any installation that has the latest version of the product installed.

Commands that work across installations

- dspmq (display queue managers)
- dspmqinst (display IBM MQ installation)
- dspmqver (display version information)
- setmqinst (set IBM MQ installation)

Other control commands for multiple installations

- crtmqenv (create IBM MQ environment)
- dspmqinst (display IBM MQ installation)
- setmqenv (set IBM MQ environment)
- setmqinst (set IBM MQ installation)

- setmqm (set queue manager)

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**. 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.

Multiple releases on z/OS

z/OS

Multiple releases can exist on z/OS. You use STEPLIBs to control which level of IBM MQ is used. For more information, see “Coexistence” on page 94.

Related concepts
Multiple installations

ULW *Multiple installations and application programs*

When a local application connects to a queue manager, the application needs to load the libraries from the installation associated with the queue manager. Multiple installations introduce some complexity.

Using the `setmqm` command

When you use setmqm to change the installation associated with a queue manager, the libraries that need to be loaded change.

When an application connects to multiple queue managers owned by different installations, multiple sets of libraries need to be loaded.

Note: If you link your applications to IBM MQ libraries, the applications automatically load the appropriate libraries when the application connects to a queue manager.

Loading IBM MQ libraries in a multi-version environment

How libraries are located depends upon your environment.

If IBM MQ is installed in the default location, existing applications continue to work as before. Otherwise, you might need to rebuild the application or change your configuration.

The order in which libraries are searched, depends upon the platform you are using:

- Windows
  - The application's directory
  - The current directory
  - The global and your PATH variables
- Other platforms
  - LD_LIBRARY_PATH (or LIBPATH/SHLIB_PATH)
  - An embedded search path (RPath)
  - The default library path

Table 6. Options for loading libraries

Platform	Option	Benefits	Drawbacks
UNIX	Set/change the embedded runtime search path (RPath)	The path is explicit in the way the application is built	You need to recompile and link If you move IBM MQ, you must change RPath

Table 6. Options for loading libraries (continued)

Platform	Option	Benefits	Drawbacks
UNIX	Set LD_LIBRARY_PATH or equivalent using setmqenv	Overrides RPath No changes to existing applications Easy to change if you move IBM MQ	Depends on environment variables Possible impacts on other applications
Windows	Set PATH using setmqenv	No changes to existing applications Easy to change if you move IBM MQ	Depends on environment variables Possible impacts on other applications
All	Set the primary installation to IBM WebSphere MQ 7.1 or later	No changes to existing applications Easy to change the primary installation Similar behavior to previous versions of IBM MQ	UNIX: Relies on `/usr/lib` in the default search path

Related concepts

Multiple installations

Coexistence, compatibility, and interoperability

The definitions of the IBM MQ terms 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 IBM MQ, it normally means installing and running multiple versions of IBM 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.

If you are using a message channel agent (MCA) channel, any version and release of an IBM MQ queue manager can connect, using an MCA channel, to any version and release of another IBM MQ queue manager.

The MCA channel is automatically configured to the latest version of protocol that is supported by both ends of the channel.

Compatibility is also the ability to run client applications with different versions of the IBM MQ MQI client, and different levels of the queue manager.

Interoperability

Is mainly the ability to exchange messages between different versions of IBM MQ. It can also mean the interoperability between others things, such as publish/subscribe brokers, or between components such as the IBM MQ classes for JMS and WebSphere Application Server.

Maintaining the compatibility, coexistence, and interoperability of IBM 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 IBM MQ Explorer.
- Information for service, such as FFST files and traces.

- Error messages. The text in an error message might change, to make the wording clearer or more accurate.

Coexistence

Queue managers, with different names, can coexist on any server as long as they use the same IBM MQ installation. On z/OS, 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.

Multiple queue manager versions in z/OS

z/OS

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

This means that you can run one queue manager at the latest version and another in the same image with an earlier version, provided that the early code is that of the latest version.

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" on page 95
Queue managers, with different names, can coexist on any server as long as they use the same IBM MQ installation. On z/OS, 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 98
You can install multiple copies of IBM MQ for UNIX, Linux, and Windows on the same server. These IBM MQ copies can be at the same or different version levels. This is called a multi-installation. Multi-installation is particularly useful when you upgrade from one IBM MQ version to a later version, because it allows you to run the earlier version alongside the later version.

Related tasks

"Migrating IBM MQ library loading to a later version on UNIX and Linux" on page 164
On UNIX and Linux, no change in the way IBM MQ libraries are loaded is normally required if you upgrade from an earlier version of the product to a later version by replacing an earlier version of the product with the later version, based on the single stage scenario. However, if you choose to take advantage of multi-installation in the later version of the product, based on the side-by-side or multi-stage migration scenarios, you might have to configure the runtime environment differently, for the operating system to load the later version of the IBM MQ library.

"Migrating IBM MQ library loading to a later version on Windows" on page 134
On Windows, no change in the way IBM MQ libraries are loaded is normally required if you upgrade from an earlier version of the product to the later version by replacing an earlier version of the product with the later version, based on the single stage scenario. However, if you choose to take advantage of multi-installation in the later version of the product, based on the side-by-side or multi-stage migration scenarios, you might have to configure the runtime environment differently, for the operating system to load the later version of the IBM MQ library.

Queue manager coexistence

Queue managers, with different names, can coexist on any server as long as they use the same IBM MQ installation. On z/OS, 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 managers 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.

Queue manager coexistence in a multi-installation

ULW z/OS Multi-installation queue manager coexistence is supported on UNIX, Linux, and Windows, and z/OS.[1]

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 MQ.

The biggest benefit of multi-installation identified by early users, is in upgrading from one version of IBM 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 8 on page 96, which is extracted from “Migrating on UNIX and Linux: side-by-side” on page 153 or “Migrating on Windows: side-by-side” on page 124.

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

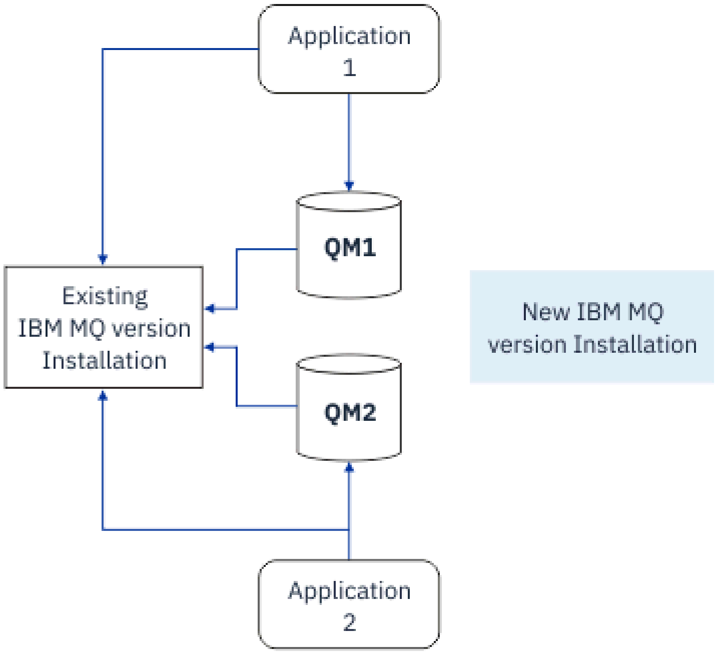


Figure 8. Side-by-side installation - step 2

When the installation is complete, and verified, migrate queue managers and applications to the new installation; see Figure 9 on page 96. When migration is complete, uninstall the old installation.

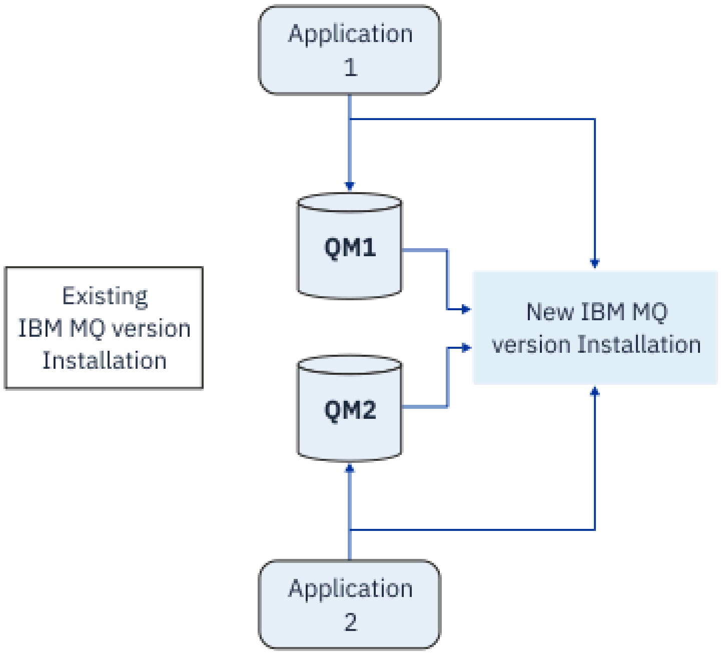


Figure 9. Side-by-side installation - step 4

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 the following tasks:

1. "Migrating on UNIX and Linux: single-stage" on page 150 or "Migrating on Windows: single stage" on page 120

2. “Migrating on UNIX and Linux: side-by-side” on page 153 or “Migrating on Windows: side-by-side” on page 124
3. “Migrating on UNIX and Linux: multi-stage” on page 156 or or “Migrating on Windows: multi-stage” on page 127

Migration of queue managers to a new maintenance level

LTS

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

Related concepts

“Multi-installation queue manager coexistence on UNIX, Linux, and Windows” on page 98
You can install multiple copies of IBM MQ for UNIX, Linux, and Windows on the same server. These IBM MQ copies can be at the same or different version levels. This is called a multi-installation. Multi-installation is particularly useful when you upgrade from one IBM MQ version to a later version, because it allows you to run the earlier version alongside the later version.

“Upgrade and migration of IBM MQ on z/OS ” on page 197
You can install new releases of IBM MQ to upgrade IBM MQ to a new release, or version level. Multiple installations at the same or different levels can coexist on the same z/OS instance. Running a queue manager at a higher level requires migration.

Related tasks

“Migrating IBM MQ library loading to a later version on UNIX and Linux” on page 164
On UNIX and Linux, no change in the way IBM MQ libraries are loaded is normally required if you upgrade from an earlier version of the product to a later version by replacing an earlier version of the product with the later version, based on the single stage scenario. However, if you choose to take advantage of multi-installation in the later version of the product, based on the side-by-side or multi-stage migration scenarios, you might have to configure the runtime environment differently, for the operating system to load the later version of the IBM MQ library.

“Staging maintenance level updates on AIX” on page 17
On AIX, you can use multiple installations of IBM MQ on the same server to control the release of maintenance level updates.

“Staging maintenance level updates on Linux” on page 32
On Linux, you can use multiple installations of IBM MQ on the same server to control the release of maintenance level updates.

“Staging maintenance level updates on Solaris” on page 43
On Solaris, you can use multiple installations of IBM MQ on the same server to control the release of maintenance level updates.

“Staging maintenance level updates on Windows” on page 59
On Windows systems, you can use multiple installations of IBM MQ on the same server to control the release of maintenance level updates.

“Migrating IBM MQ library loading to a later version on Windows” on page 134
On Windows, no change in the way IBM MQ libraries are loaded is normally required if you upgrade from an earlier version of the product to the later version by replacing an earlier version of the product with the later version, based on the single stage scenario. However, if you choose to take advantage of multi-installation in the later version of the product, based on the side-by-side or multi-stage migration scenarios, you might have to configure the runtime environment differently, for the operating system to load the later version of the IBM MQ library.

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

You can install multiple copies of IBM MQ for UNIX, Linux, and Windows on the same server. These IBM MQ copies can be at the same or different version levels. This is called a multi-installation. Multi-installation is particularly useful when you upgrade from one IBM MQ version to a later version, because it allows you to run the earlier version alongside the later version.

Note: On Linux, a multi-installation can only be created when using the RPM installation media. Multi-installation is not supported on Ubuntu.

When you plan a multi-installation, you need only consider the major IBM MQ version number (for example, IBM MQ 9.0). CD releases and fix pack levels are not a significant additional factor.

The following figure shows two IBM MQ installations at different versions (for example versions 9.0 and 8.0), two queue managers, and three applications. In this figure, note that application 3 is configured to load libraries from the `Inst_1` (IBM MQ 9.0) installation, even though it is connected to queue manager QM2 (IBM MQ 8.0).

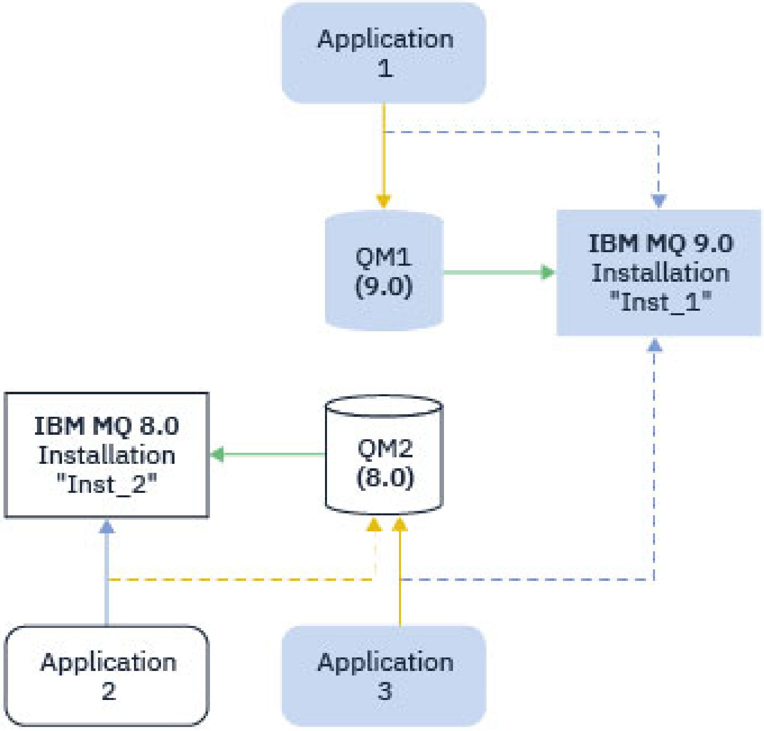

Application 1
QM1 (9.0)
IBM MQ 9.0 Installation "Inst_1"
IBM MQ 8.0 Installation "Inst_2"
QM2 (8.0)
Application 2
Application 3

Figure 10. Coexistence of two queue managers running at different IBM MQ versions

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

1. Which installation is a queue manager associated with? See "Queue manager association" on page 98.
2. Which installation does an application load? See "Loading IBM MQ libraries" on page 99.
3. Which installation is an IBM MQ command run from? See "Command association" on page 100.

Queue manager association

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 10 on page 98, 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 is running IBM MQ 8.0 it is migrated to the later version. QM2 is associated with IBM MQ 8.0 because the association has not been changed.

Loading IBM MQ libraries

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

Which IBM MQ library an application loads depends on the configuration of the operating system loader, and on the IBM MQ installation the queue manager is associated with. In Figure 10 on page 98, the operating system loads the IBM MQ library from the Inst_1 installation for applications 1 and 3. It loads the IBM MQ 8.0 library for application 2. The operating system has loaded the wrong library for application 3. Application 3 requires the IBM MQ 8.0 libraries.

Figure 11 on page 99 shows what happens to application 3. Application 3 is connecting to QM2, and QM2 is associated with the IBM MQ 8.0 installation. IBM MQ detects that the operating system has loaded the wrong library to process calls from application 3 to QM2. IBM MQ loads the correct library from the IBM MQ 8.0 installation. It transfers the MQCONN or MQCONNX call to the IBM MQ 8.0 library. Subsequent MQI calls that use the connection handle returned by MQCONN or MQCONNX, call entry points in the IBM MQ 8.0 library.

If you attempt a connection to QM1 with application 2, IBM MQ returns an error; see 2059 (080B) (RC2059): MQRC_Q_MGR_NOT_AVAILABLE.

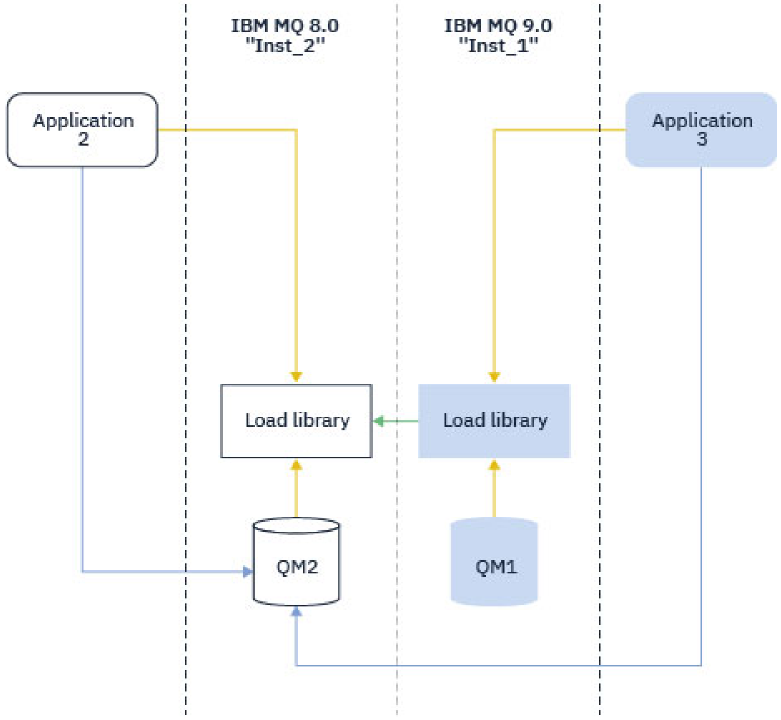


Figure 11. Loading calls in a different library

From IBM WebSphere MQ 7.1, IBM MQ libraries include a routing capability that is based on the installation a queue manager is associated with. The operating system can load a library from any IBM WebSphere MQ 7.1 or later installation, and IBM MQ transfers MQI calls to the correct library.

The loading capability of IBM MQ libraries does not relax the restriction that an application compiled and linked at a later release level must not directly load an IBM MQ library at an earlier release level. In practice, 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 MQ can call any other level of IBM MQ on the same server.

For example, suppose you recompile and link an application that is to connect to an IBM MQ 8.0 queue manager using the libraries shipped with IBM MQ 9.0. At run time the operating system must load the IBM MQ 9.0 libraries for the application, even though the application connects to an IBM MQ 8.0 queue manager. IBM MQ 9.0 detects the inconsistency and loads the IBM MQ 8.0 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 MQ library that matches the later release, even if it continues to connect to an IBM MQ 9.0 queue manager.

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

Special migration considerations involving loading IBM MQ libraries

You might have modified the installation of an early IBM MQ release to satisfy the requirements of a build environment, or the IT standards in your organization. If you copied IBM MQ libraries to other directories, or created symbolic links, you ended up with an unsupported configuration. A common IT standard or build environment requirement is to include IBM MQ libraries in the default load path on UNIX and Linux systems. From IBM WebSphere MQ 7.5, you can install IBM MQ into a directory of your own choosing, and IBM MQ can create symbolic links in `/usr` and its subdirectories. If you make an IBM MQ installation primary by using the **setmqinst** command, IBM MQ inserts symbolic links to the IBM MQ libraries into `/usr/lib`. As a result, the operating system finds the IBM MQ libraries in the default load path, if that includes `/usr/lib`.

For more information, see 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 an IBM 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. For more information, see Running **setmqenv**.

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`. For more information, 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.

The following code shows two examples of running **setmqenv** to set up the command environment for the copy of IBM MQ that is associated with queue manager `QM1`.

IBM MQ for Windows.

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

IBM MQ for UNIX or Linux.

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

Figure 12. Running ***setmqenv***

Related concepts

Connecting applications in a multiple installation environment

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

Related tasks

“Migrating on UNIX and Linux: single-stage” on page 150
Single-stage migration is the term used to describe replacing the only installation of IBM MQ on a server, with a later release. Single stage migration is also known as *upgrading in place* or *in place upgrade*. Single-stage migration preserves existing scripts and procedures for running IBM 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.

“Migrating on Windows: single stage” on page 120
Single-stage migration is the term used to describe replacing the only installation of IBM MQ on a server, with a later version of the product. Single stage migration is also known as *upgrading in place* or *in place upgrade*. Single-stage migration preserves existing scripts and procedures for running IBM 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.

Changing the primary installation

“Staging maintenance level updates on AIX” on page 17
On AIX, you can use multiple installations of IBM MQ on the same server to control the release of maintenance level updates.

“Staging maintenance level updates on Linux” on page 32
On Linux, you can use multiple installations of IBM MQ on the same server to control the release of maintenance level updates.

“Staging maintenance level updates on Windows” on page 59
On Windows systems, you can use multiple installations of IBM MQ on the same server to control the release of maintenance level updates.

“Migrating IBM MQ library loading to a later version on UNIX and Linux” on page 164
On UNIX and Linux, no change in the way IBM MQ libraries are loaded is normally required if you upgrade from an earlier version of the product to a later version by replacing an earlier version of the product with the later version, based on the single stage scenario. However, if you choose to take advantage of multi-installation in the later version of the product, based on the side-by-side or multi-stage migration scenarios, you might have to configure the runtime environment differently, for the operating system to load the later version of the IBM MQ library.

“Migrating IBM MQ library loading to a later version on Windows” on page 134
On Windows, no change in the way IBM MQ libraries are loaded is normally required if you upgrade from an earlier version of the product to the later version by replacing an earlier version of the product with the later version, based on the single stage scenario. However, if you choose to take advantage of multi-installation in the later version of the product, based on the side-by-side or multi-stage migration scenarios, you might have to configure the runtime environment differently, for the operating system to load the later version of the IBM MQ library.

Related reference

"Coexistence" on page 94
Queue managers, with different names, can coexist on any server as long as they use the same IBM MQ installation. On z/OS, 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

Dynamic-Link Library Search Order

Mixed version cluster coexistence

A cluster can contain queue managers running at IBM MQ IBM MQ 9.1, and any currently supported earlier level of the product. However new features cannot be exploited from queue managers at an earlier level.

Routing behavior in a mixed version publish/subscribe cluster

From IBM MQ 8.0, topic host routing is available for publish/subscribe clusters. The queue manager where the object is defined, and the full repository queue managers, must be at a level that supports the topic route hosting feature, that is IBM MQ 8.0 or later. Any queue manager in the cluster that is at an earlier level does not adhere to the topic route hosting behavior.

When a clustered topic is defined for topic host routing (by setting the topic **CLROUTE** parameter to TOPICHOST), only queue managers at the new level are aware of the clustered topic. Older queue managers do not receive the clustered topic definition and therefore behave as if the topic is not clustered. This means that all queue managers that need to work in a routed publish/subscribe manner must be at a version that supports this feature, not just the queue managers that host the routed topics.

Important notes:

- All full repositories must be at IBM MQ 8.0 or later to use this feature. If a full repository queue manager is at an earlier version, the **CLROUTE** of TOPICHOST is not recognized by the full repository, and the full repository propagates the topic definition to all queue managers in the cluster. Any pre-IBM MQ 8.0 queue managers then use the topic as if it is defined for DIRECT routing. This behavior is unsupported.
- If an older queue manager defines a direct routed clustered topic with the same name as an existing topic host routed clustered topic, the full repositories notice the conflicting definition and do not propagate the definition.

To find out the version of each queue manager in the cluster, specify the VERSION parameter with the DISPLAY CLUSQMGR command. If you issue this command from a queue manager with a full repository, the information returned applies to every queue manager in the cluster. Otherwise the information returned applies only to the queue managers in which it has an interest. That is, every queue manager to which it has tried to send a message and every queue manager that holds a full repository.

z/OS *ISPF operations and control panels on z/OS*

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

Compatibility of queue manager versions with operations and control panel versions on z/OS shows which versions of the operations and controls panels you use in ISPF are compatible with which levels of queue manager.

V 9.1.0

Table 7. Compatibility of queue manager versions with operations and control panel versions on z/OS

Version	Queue sharing group containing a mixture of IBM MQ 8.0 IBM MQ 9.0, and IBM MQ 9.1 queue managers.	IBM MQ 9.1 queue manager	IBM MQ 9.0 queue manager	IBM MQ 8.0 queue manager
IBM MQ 9.1 panel	Compatible	Compatible	Compatible	Compatible
IBM MQ 9.0 panel	Not compatible	Not compatible	Compatible	Compatible
IBM MQ 8.0 panel	Not compatible			Compatible

z/OS *Queue sharing group coexistence on z/OS*

V 9.1.0 A queue sharing group can contain queue managers running on IBM MQ 8.0.0, and on later releases. The queue managers can access the same shared queues and other shared objects.

Notes: V 9.1.0

1. Unless stated otherwise, Long Term Support (LTS) and Continuous Delivery (CD) queue managers, with the same version and release numbers, can coexist in a queue sharing group without the need for a backwards migration PTF.
2. LTS version 9.1.0, and CD version 9.0.n queue managers can also coexist in a queue sharing group without the need for a backwards migration PTF.

Only run queue managers in a mixed-version queue sharing group for the time it takes to migrate all queue managers to the later version.

V 9.1.4 IBM MQ for z/OS 9.1.3, or later, supports z/OS data set encryption for active log data sets. Earlier releases of IBM MQ for z/OS do not.

V 9.1.4 This means that queue managers with encrypted archive logs cannot coexist with IBM MQ 9.1.3 or earlier release queue managers in a queue sharing group.

V 9.1.4 See the section, confidentiality for data at rest on IBM MQ for z/OS with data set encryption. for more information.

Notes:

1. The rules for coexistence of IBM MQ for z/OS 9.1.3 or later queue managers without encrypted active logs are the same as those for IBM MQ for z/OS 9.1.3 queue managers. This includes queue managers with page sets encrypted using z/OS data set encryption.
2. You can convert queue manager encrypted active logs to unencrypted active logs, to allow coexistence with earlier versions of IBM MQ for z/OS.

z/OS *Properties of objects in a mixed queue sharing group on z/OS*

Attributes that did not exist in earlier versions can be created and altered on queue managers of a later version in a mixed queue sharing group. The attributes are not available to queue managers in the group that are at an earlier level.

z/OS *MQSC commands in a mixed queue sharing group on z/OS*

Existing **MQSC** commands using new keywords and attribute values can be entered for routing to a migrated queue manager. You can enter the commands on any queue manager. Route the commands using **CMDSCOPE**. Commands with new keywords and attribute values, or new commands, routed to a previous version of queue manager, fail.

Application compatibility and interoperability with earlier versions of IBM MQ

Connecting an application that is built against libraries shipped with a later version of IBM MQ to an earlier version IBM 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 MQ applications do interoperate with applications running on earlier versions of IBM MQ, as long as they use no new function. IBM 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 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 MQ an application is built on and connected to. It can exchange messages with an application connected to an earlier version of IBM 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 MQ. The last two cases rely on the interoperability between all versions of IBM MQ

1. Running an IBM MQ server application, built with a later version of IBM MQ, connecting to a queue manager running on a server with an earlier version of IBM MQ installed.
2. Running an IBM MQ client application, built with a later version of IBM 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 MQ installed.
3. Running an IBM MQ client application, built with a later version of IBM 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 MQ installed.
4. Exchanging messages between an IBM MQ client or server application, connected to a queue manager running on a server with a later version of IBM MQ installed, with applications connected to a queue manager running on a server with an earlier version of IBM 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 MQ. You can then continue with problem diagnosis.

Multi-installation and application loading

The loading capability of IBM MQ libraries does not relax the restriction that an application compiled and linked at a later release level must not directly load an IBM MQ library at an earlier release level. In practice, 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 MQ can call any other level of IBM MQ on the same server.

For example, suppose you recompile and link an application that is to connect to an IBM MQ 8.0 queue manager using the libraries shipped with IBM MQ 9.0. At run time the operating system must load the IBM MQ 9.0 libraries for the application, even though the application connects to an IBM MQ 8.0 queue manager. IBM MQ 9.0 detects the inconsistency and loads the IBM MQ 8.0 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 MQ library that matches the later release, even if it continues to connect to an IBM MQ 9.0 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 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 MQ client.
2. You deploy some IBM MQ queue managers at a new version level. You have an existing IBM 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 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 MQ does not support compatibility with earlier versions.

z/OS application stubs

z/OS

The stub modules that are listed are link-edited with applications and exits.

- CSQASTUB
- CSQBRSSI
- CSQBRSTB
- CSQBSTUB
- CSQCSTUB
- CSQQSTUB
- CSQXSTUB

Application compatibility and interoperability with later versions of IBM MQ

IBM 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 IBM MQ to a queue manager running at a later version of IBM 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 73. Likewise applications connected to the IBM 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 MQ Client libraries to connect using later versions of the libraries.

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 rely on the compatibility of a later version of IBM MQ with applications built against earlier versions. The last two cases rely on the interoperability between all versions of IBM MQ.

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

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 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.

z/OS application stubs

z/OS

The stub modules that are listed are link-edited with applications and exits.

- `CSQASTUB`
- `CSQBRSSI`
- `CSQBRSTB`
- `CSQBSTUB`
- `CSQCSTUB`
- `CSQQSTUB`
- `CSQXSTUB`

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

Any supported version and release of an IBM MQ client can connect to any supported version and release of an IBM MQ queue manager. Supported IBM MQ clients are all clients included with the main MQ product. 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, upgrade to a more recent fix pack for the same `Version.Release`.

When you upgrade a queue manager to a different version, you automatically upgrade IBM MQ libraries. The libraries are used by IBM MQ 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 MQ client installation on remote workstations. The IBM MQ client includes the IBM MQ client libraries.

Remote clients that have not been upgraded continue to work with an upgraded queue manager. In rare cases, the behavior of the client application might change. See "Changes that affect migration" on page 73.

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 73.

You can generally assume that upgrading the IBM MQ 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 a queue manager on Windows" on page 116, 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 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 MQ 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 IBM 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 IBM MQ client installation to the new version.

MQ clients: Client Channel Definition Table (CCDT)

You can connect an IBM MQ client application to any level of queue manager. If a 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.

When a client uses a CCDT file that was generated using a newer version of IBM MQ, only channel attributes within the CCDT that were available at the client's IBM MQ version are considered during negotiation with the queue manager. Channel attributes present in the CCDT that were added in newer versions of IBM MQ will be ignored by older clients.

Version of originating queue manager for a CCDT

In earlier releases before IBM MQ 9.0, clients can use a CCDT built by the same or earlier version queue manager, but there was previously a restriction on clients using a CCDT built by a later version queue manager. However, this restriction is removed in IBM MQ 9.0.

From IBM MQ 9.0, if a client uses a CCDT, it can use a CCDT built by a later version queue manager, as well as a CCDT built by the same, or earlier version of queue manager.

The same restriction on the use of CCDTs originating from later version queue managers is also removed in IBM MQ 8.0, IBM WebSphere MQ 7.5, and IBM WebSphere MQ 7.1 by APARs IT10863 and IT11547. for more information, see the technote MQ 7.x, MQ 8.0, MQ 9.0 and MQ 9.1 compatibility with previous versions - including usage of CCDT files, JMS .bindings, SSL/TLS.

Common migration scenarios

If, for example, you upgrade a queue manager from an earlier release to a later release, and you do not create new CCDTs for its clients, the clients connect to the later release queue manager without any changes being required. Client behavior might change as a result of changes to the queue manager.

Another common migration scenario is to update some queue managers and some clients to a later release, leaving other queue managers and clients at the earlier release. In this scenario, you want to update the CCDT for the IBM MQ clients that are connected to the later release queue managers to that later release so that those clients can fully use the function in the later release. The new clients can also connect to the earlier release queue managers. Existing clients connect to queue managers in both releases. In order that the clients in the later release can use the new function in that release, you must deploy a CCDT that has been generated by a queue manager in that new release. Clients in the earlier release can continue to use the CCDT for that earlier release. Both sets of clients can connect to both sets of queue managers, regardless of the CCDT they are using.

Related concepts

Client channel definition table

Web addressable access to the client channel definition table

Related tasks

Accessing client-connection channel definitions

MQ clients: Client configuration stanzas moved into a new configuration file

Client configuration information is moved from existing configuration stanzas into a new configuration file, `mqclient.ini`.

Moving client configuration information affects existing settings; for example:

- Set the TCP `KeepAlive` attribute for client connections in `mqclient.ini` ; for example:

  ```
  TCP:
  KeepAlive = Yes
  ```

  An existing setting in `qm.ini` is ignored.
- Set `ClientExitPath` in `mqclient.ini` ; for example:

  ```
  ClientExitPath:
  ExitsDefaultPath=/var/mqm/exits
  ExitsDefaultPath64=/var/mqm/exits64
  ```

  An existing setting in `mqs.ini` is moved to the client configuration file when you upgrade the client. If you add values to `mqs.ini`, they are ignored.
- Set `JavaExitsClasspath` in `mqclient.ini`.

  Do not continue to use the Java system property `com.ibm.mq.exitClasspath`. Existing settings continue to work, but they are deprecated. The setting in `mqclient.ini` has precedence over the Java system property.

Related tasks

Assigning channel exits for IBM MQ classes for JMS

Related reference

The IBM MQ classes for JMS configuration file

IBM MQ client configuration file

Supported IBM MQ client: Default behavior of client-connection and server-connection channels

The default settings for client and server connection channels use shared conversations.

The default for client and server connections is to share an MQI channel. You use the **SHARECNV** (sharing conversations) parameter to specify the maximum number of conversations that can be shared over a particular TCP/IP client channel instance. The possible values are as follows:

SHARECNV(0)

This value specifies no sharing of conversations over a TCP/IP socket. Only use a value of `0` if you have existing client applications that do not run correctly when you set **SHARECNV** to 1 or greater.

SHARECNV(1)

This value specifies no sharing of conversations over a TCP/IP socket. Performance on distributed servers is similar to that for a value of `0`. Client heartbeating (whether in an MQGET call or not) and read ahead are available, and channel quiescing is more controllable.

SHARECNV(2) to SHARECNV(999999999)

Each of these values specifies the number of shared conversations. If the client-connection **SHARECNV** value does not match the server-connection **SHARECNV** value, then the lowest value is used. The default value is SHARECNV(10), which specifies 10 threads to run up to 10 client conversations per channel instance. However, on distributed servers there are performance issues with SHARECNV channels that can be eased by using SHARECNV(1) wherever possible.

For all **SHARECNV** values of 1 or greater, the channel supports the following features:

- Bi-directional heartbeats
- Administrator stop-quiesce
- Read-ahead
- Asynchronous-consume by client applications

You can also set the MQCONNX option, MQCNO_NO_CONV_SHARING and connect the application to a channel with **SHARECNV** set to a value greater than 1. The result is the same as connecting the application to a channel with **SHARECNV** set to 1.

Performance

The use of shared conversations, and further enhancements introduced in IBM MQ 8.0, can impact performance on distributed servers. See Tuning client and server connection channels.

Heartbeats

Heartbeats can flow across the channel at any time in either direction. SHARECNV(0) is for heartbeats to flow only when an MQGET call is waiting.

Channel exits

The behavior of a client or server connection channel exit changes when the channel is sharing conversations (that is, when you set **SHARECNV** to a value greater than 1). It is unlikely, but possible, that the change affects the behavior of existing exits. The change is as follows:

- Send or receive exits can alter the MQCD structure on an MQXR_INIT call. The effect of these exits differs, depending on whether the conversation is shared with other conversations on the same channel:
  - If the MQCXP SharingConversations field passed to the exit instance is set to FALSE, this exit instance is the first, or only, conversation on the channel instance. No other exit can be changing the MQCD at the same time, and changes that are made to the MQCD can affect the way that the channel runs.
  - If the MQCXP SharingConversations field passed to the exit instance is set to TRUE, this exit instance is a subsequent conversation. It is sharing the channel instance with other conversations. Changes made to the MQCD in the exit instance are retained in the MQCD but do not affect the way the channel runs.
- Send, receive, and security exit instances can alter the MQCD, when the MQCXP SharingConversations field is set to TRUE. Exit instances on other conversations might be changing the MQCD at the same time. Updates written by one exit instance can be overwritten by another instance. It might be necessary to serialize access to the MQCD across these different exit instances to maintain the consistency of fields in MQCD.

Updating MQCD when the SharingConversations field is set to TRUE does not affect the way the channel runs. Only alterations made when the MQCXP SharingConversations field is set to FALSE, on an MQXR_INIT call, change channel behavior.

Related concepts

Channel-exit programs for MQI channels

Related tasks

Using sharing conversations

Using read ahead

Stopping MQI channels

Tuning client and server connection channels

Related reference

HeartbeatInterval (MQLONG)

SharingConversations (MQLONG)

ALTER CHANNEL

The Asynchronous consume sample program

GSKit version compatibility

The stash files that you generate with IBM MQ 9.0.0 Fix Pack 1, or later must be compatible with any applications and other IBM MQ installations.

For the version of GSKit for IBM MQ 9.0.0 Fix Pack 1 or later, the stash file format that is used when you generate an `.sth` file to stash the key database password is different from earlier versions of GSKit. Stash files that are generated with this version of GSKit are not readable by earlier versions of GSKit. To ensure that stash files that are generated with IBM MQ 9.0.0 Fix Pack 1, or later, are compatible with your applications and other IBM MQ installations, you must update to a version of IBM MQ that contains a compatible version of GSKit. The following fix packs contain a compatible version of GSKit:

- V7.1.0.8
- V7.5.0.8
- V8.0.0.6
- V9.0.0.1

If you cannot update your applications or other IBM MQ installations, you can request a stash file format that is compatible with an earlier version. When you use the **`runmqakm`** or **`runmqckm`** commands with the `-stash` or `-stashpw` option, include the `-v1stash` command line parameter. You cannot use the **`strmqikm`** GUI to generate a stash file that is compatible with an earlier version.

CD Migrating from one Continuous Delivery release to another

An overview of how you migrate from one Continuous Delivery (CD) release to another.

Before you begin

If you want to migrate replicated data queue managers, follow the instructions in “Migrating replicated data queue managers” on page 228.

1. Back up your existing IBM MQ environment. This is required in case you need to revert to your current CD release of IBM MQ.

   Important: Once the new installation is started, all of the existing IBM MQ objects will be migrated to the new modification level. If you do not back up your system, you cannot revert the objects to their previous level without completely uninstalling, and restoring a backup you made before you carried out the migration.

   Windows Copy the `qm.ini` file and the registry entries.

   For more information about backing up your system, see Backing up and restoring IBM MQ queue manager data.
2. Use the **`dmpmqcfg`** command to save the current configuration details to a text file.

Procedure

1. Stop all of the IBM MQ processes for the installation being migrated.
2. Uninstall the existing CD installation.

   Note: This does not remove the object definitions from the system; they remain in place.
3. Install the new CD modification level, for example, `9.1.2` onto the same system.
4. Start the queue manager.

   ```
   strmqm QmgrName
   ```

   When you first start a queue manager after migration to the new CD level:

   - Any new attributes for existing objects are set to their default values.
   - Any new default objects are created.
   - Queue manager objects are migrated to the new modification level.

   Note: If you have saved your current configuration details in a text file, that file can be used to duplicate these objects in the newly created queue manager after it has been created, if you installed the new version onto a different system.

   See the **`runmqsc`** command for instructions on how you can do this.

Related concepts
IBM MQ release types

Windows Migrating IBM MQ on Windows

IBM MQ migration tasks associated with Windows platforms are grouped in this section.

About this task

CD

This topic lists the various steps you need to take to migrate to, or migrate from, the latest version of the IBM MQ product.

See also, "Migrating from one Continuous Delivery release to another" on page 110, if you are migrating a Continuous Delivery release of the product.

Procedure

- For information about creating a migration plan, see "Planning to migrate IBM MQ to a later version on Windows" on page 112.
- For information about migrating a queue manager from an earlier version to the latest version, see "Migrating a queue manager to a later version on Windows" on page 117.
- For information about reverting a queue manager to an earlier version, see "Reverting a queue manager to an earlier version on Windows" on page 130.
- For information about migrating an IBM MQ MQI client to the latest version, see "Migrating an IBM MQ MQI client to a later version on Windows" on page 133.
- For information about converting a single instance queue manager to a multi-instance queue manager, see "Migrating from a single instance to a multi-instance queue manager on Windows" on page 138.
- For information about reverting a multi-instance queue manager to a single instance queue manager, see "Reverting to a single-instance queue manager on Windows " on page 141.
- For information about migrating IBM MQ library loading to the latest version, see "Migrating IBM MQ library loading to a later version on Windows" on page 134.

- For information about migrating MQ Telemetry to the latest version, see "Migrating MQ Telemetry on Windows" on page 142.
- For information about migrating an MSCS configuration to the latest version, see "Migrating an MSCS configuration on Windows" on page 142.
- V 9.1.0

  For information about Migrating logs to an Advanced Format disk, see "Migrating logs to an Advanced Format disk on Windows" on page 145.

Related concepts

"Migration concepts and methods" on page 76
An overview of the various concepts and methods for migrating from one release of the product to another.

Related tasks

"Migrating IBM MQ on UNIX and Linux" on page 145
Migration tasks associated with UNIX and Linux platforms are grouped in this section.

"Migrating IBM MQ on IBM i" on page 174
IBM MQ migration tasks associated with IBM i are grouped in this section.

"Migrating IBM MQ on z/OS" on page 195
Migration tasks associated with z/OS are grouped in this section.

Related reference

"Changes that affect migration" on page 73

Windows Planning to migrate IBM MQ to a later version on Windows

Before migrating IBM MQ to a later version on Windows, review the system requirements information, and the information about any changes that might affect migration, then create a migration plan.

Before you begin

If there are concepts about migration you do not understand, see "Migration concepts and methods" on page 76.

If you are migrating to IBM MQ 9.1 from IBM WebSphere MQ 7.5 or earlier, you must first migrate to an interim version. See Migration paths.

Before you use side-by-side or multi-stage migration to migrate from IBM WebSphere MQ 7.0.1, check that your IBM WebSphere MQ 7.0.1 installation is at Fix Pack 6 or later.

About this task

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

Procedure

1. Review the IBM MQ system requirements for the later version of the product.

   See System Requirements for IBM MQ.
2. Decide whether to run the earlier version and the later version of the product on the same server, and also which migration method you want to use.

   Choices are single-stage migration, side-by-side migration, or multi-stage migration. See "Migration methods on IBM MQ for Multiplatforms" on page 84.
3. Review all the changes in IBM MQ that affect you.

   See "Changes that affect migration" on page 73.
4. Review performance changes.

See MQ Performance documents.

5. Review the readme file for the later version of IBM MQ.

   See IBM MQ, WebSphere MQ, and MQSeries product readmes.
6. Plan the sequence and timing of queue manager migrations.
   - 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 “Migrating a queue manager in a high-availability configuration” on page 225.
7. Plan to migrate your queue manager to the later version.

   See “Migrating a queue manager to a later version on Windows” on page 117.

   Backing up queue manager data is part of the queue manager migration task. An alternative approach is to install and configure a new server, then test the later version with a new queue manager on the new server. When you are ready to go into production on the later version, 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.

   From IBM MQ 9.1, a suffix letter, indicating the severity of a message (I, W, E, S or T) is appended to IBM MQ diagnostic (AMQ) messages. Existing scripts looking for error codes without the severity will fail. For example, existing scripts looking for error matching to AMQ7468 will fail. You must update the scripts to look for error codes with the severity suffix added (for example, AMQ7468I). For more information, see IBM MQ messages on Multiplatforms.
9. Decide on what regression tests to perform before putting the queue manager into production on the later version. Include in your regression tests the procedures and applications you identified in previous steps.
10. Plan to migrate your IBM MQ MQI client installations to the later version.
11. Plan to migrate your client and server applications to use new functions in the later version.
12. Decide which downloadable images you require for the migration.

    For more information, see Where to find downloadable installation images.

Windows *Migration considerations for IBM MQ 8.0 or later on Windows*

From IBM MQ 8.0, a number of changes were made for IBM MQ for Windows. You must understand these changes before planning any migration tasks for IBM MQ 8.0 or later on Windows.

Installing a single copy of the product

If you have an existing previous version of the product on your system, and want to upgrade to the latest version, you have various options. You can either:

- Uninstall the previous version and then install the latest version,
- Install the new copy alongside the currently installed one and uninstall the original at a later time. See “Installing the product alongside an existing version” on page 113, or
- Perform a migration installation, electing to replace the currently installed version when prompted.

After you have installed the product, start each queue manager and its data migration takes place. This includes the migration of queue managers from 32-bit to 64-bit.

Installing the product alongside an existing version

If you want to install another version of the product alongside your existing product you can do so. See “Multiple IBM MQ installations” on page 90 and “Migrating on Windows: side-by-side” on page 124 for further information.

When you install the new version of the product, run the setmqm command to associate the queue managers with the new installation.

Start each queue manager in turn and its data migration takes place.

Upgrading one of a pair of (or more) installations

If you already have, for example, an IBM MQ 8.0 installation and an IBM MQ 9.0 installation on a machine, upgrading the IBM MQ 8.0 installation to IBM MQ 9.0 requires the following additional step.

When you start the IBM MQ 9.0 installer, you are asked whether you want to **Install a new instance** or **Maintain or upgrade an existing instance**.

However, only the other IBM MQ 9.0 installation, or installations, are displayed; not the IBM MQ 8.0 installation in the selection box. At this point, select **Install a new instance**.

After the splash screen has been displayed, a second panel appears, which lists any older installations that you can upgrade to IBM MQ 9.0 using the IBM MQ 9.0 installer.

On this panel, select **Upgrade 8.0.0.n Installation 'Installation m'**, and then click **Next**.

Change of digital signature algorithm

The IBM MQ programs and installation image are digitally signed on Windows to confirm that they are genuine and unmodified.

In older releases before IBM MQ 8.0, the product was signed using the SHA-1 with RSA algorithm.

From IBM MQ 8.0, the SHA-256 with RSA algorithm is used. Some older versions of Windows do not support the new digital signature algorithm, but those versions are not supported by IBM MQ 8.0 or later.

See Hardware and software requirements on Windows systems, and ensure that you install IBM MQ 8.0 or later on a supported version of Windows.

Existing applications

All applications that were built with previous versions of the product continue to work in IBM MQ 8.0 or later with a 64 bit queue manager.

All applications using the C++ object interface need to be rebuilt; applications using the C interface are not affected.

Exits

Queue manager exits on Windows 64-bit operating systems must be compiled as 64-bit exits. Any 32-bit queue manager exits must be recompiled before they can be used with a 64-bit queue manager. If you try to use a 32-bit exit with a 64-bit queue manager on IBM MQ 8.0 or later, an AMQ9535 "invalid exit" error message is issued.

Clients

32-bit client applications can connect transparently to queue managers from all supported versions of the product. This includes 64-bit IBM MQ 8.0 or later.

Samples

From IBM MQ 8.0, the samples for the C and C++ languages are compiled as 64-bit.

Related reference

Windows: changes from IBM MQ 8.0

Directory structure on Windows systems

Hardware and software requirements on Windows systems

▶ Windows *Program and data directory locations on Windows*

The installation location for IBM MQ program binary and data files on Windows depends on the IBM MQ version you are installing, and whether this is the first time IBM MQ is being installed.

First-time installations

When you install IBM MQ for the first time, you can accept default installation locations. You can also select the custom installation option by choosing the location for the IBM MQ binary files, and the location for the IBM MQ data and logs.

Before IBM MQ 8.0, if the default option was chosen, both the IBM MQ program binary and the data files were installed in the same directory. From IBM MQ 8.0, the default location for the program binary files is different from the default location for the data files.

Table 8. Default program and data directory locations on different versions of IBM MQ on Windows

IBM MQ version	**IBM MQ program binary files installation location**	**IBM MQ data files location**
IBM WebSphere MQ 7.5	Program and data files are in one location: `C:\Program Files (x86)\IBM\WebSphere MQ`	
IBM MQ 8.0	`C:\Program Files\IBM\WebSphere MQ`	`C:\ProgramData\IBM\MQ`
IBM MQ 9.0 and later	`C:\Program Files\IBM\MQ`	`C:\ProgramData\IBM\MQ`

Subsequent installations and reinstallations

After the data directory is specified, during the installation process of any installation, it cannot be changed for subsequent installations. IBM MQ is only installed as a 64-bit version when it is installed on a 64-bit operating system.

For IBM MQ 9.0 and later, the default data directory is `C:\ProgramData\IBM\MQ`, unless a version of the product was previously installed, in which case the new installation continues to use the existing data directory.

Existing IBM MQ 9.0 installation

The default program and data directory locations are the same for IBM MQ 9.0 and later versions. Therefore you do not need to change the specification of the program and data directories when upgrading from IBM MQ 9.0 to a later version.

Existing IBM MQ 8.0 installation

Three upgrade paths are possible:

- Uninstall IBM MQ 8.0 first and then install IBM MQ 9.0 or later.
- Upgrade IBM MQ 8.0 at the beginning of the IBM MQ 9.0 or later installation process, without first uninstalling the earlier version.
- Install IBM MQ 9.0 or later alongside IBM MQ 8.0 and then uninstall IBM MQ 8.0.

When IBM MQ 8.0 is installed, the product binary files are put by default into `C:\Program Files\IBM\WebSphere MQ` and the product data and logs are put by default into `C:\ProgramData\IBM\MQ`.

When you uninstall IBM MQ 8.0, information about the location of the data directory is left in the registry. After uninstalling IBM MQ 8.0 and before installing IBM MQ 9.0 or later, you can run the `ResetMQ.cmd` script to tidy up files and data left behind by the uninstallation process.

Important: You should use this script with caution. `ResetMQ.cmd` can remove the existing queue manager configuration. For more information, see Clearing IBM MQ installation settings.

Installing IBM MQ 9.0 or later after uninstalling IBM MQ 8.0

After uninstalling IBM MQ 8.0, IBM MQ 9.0 or later is installed using the same installation name but using the IBM MQ 9.0 and later default program binary files location `C:\Program Files\IBM\MQ`. That is, the program files change location after the upgrade.

Optional: You can use the custom installation option to modify the installation path, including modifying it back to `C:\Program Files (x86)\IBM\WebSphere MQ`.

The default data path is `C:\ProgramData\IBM\MQ`.

Upgrading IBM MQ 8.0 at the beginning of the IBM MQ 9.0 or later installation process

If you install IBM MQ 9.0 or later without uninstalling IBM MQ 8.0 and choose to upgrade the IBM MQ 8.0 installation, the new program binary files replace the IBM MQ 8.0 binary files so, by default, the new binary files are in `C:\Program Files (x86)\IBM\WebSphere MQ`. The existing data path is kept so, by default, the data path is `C:\ProgramData\IBM\MQ`.

Installing IBM MQ 9.0 or later to coexist with IBM MQ 8.0

If you install IBM MQ 9.0 or later alongside IBM MQ 8.0, a unique path is chosen, which by default is `C:\Program Files\IBM\MQ`. The existing data path is kept so, by default, the data path is `C:\ProgramData\IBM\MQ`.

Existing IBM WebSphere MQ 7.5, IBM WebSphere MQ 7.1, or IBM WebSphere MQ 7.0.1 installation

If you are migrating to IBM MQ 9.1 from IBM WebSphere MQ 7.5 or earlier, you must first migrate to an interim version. See Migration paths.

For IBM WebSphere MQ 7.5 or IBM WebSphere MQ 7.1, the interim version you use can be either IBM MQ 9.0 or IBM MQ 8.0. For information about specifying program and data directories when upgrading, see either of the following topics:

- Program and data directory locations on Windows in the IBM MQ 9.0 product documentation.
- Windows: Program and data directory locations in the IBM MQ 8.0 product documentation.

For IBM WebSphere MQ 7.0.1, the interim version you use can only be IBM MQ 8.0. For information about specifying program and data directories when upgrading to IBM MQ 8.0, see Windows: Program and data directory locations in the IBM MQ 8.0 product documentation.

Related concepts

"Migration concepts and methods" on page 76
An overview of the various concepts and methods for migrating from one release of the product to another.

Related tasks

Clearing IBM MQ installation settings

Related reference

Hardware and software requirements on Windows systems

Windows Migrating a queue manager on Windows

The procedures for migrating a queue manager to a later version of the product, and for restoring a queue manager to an earlier version of the product are detailed in this section.

Related tasks

"Migrating a queue manager on UNIX and Linux" on page 147
The procedures for migrating a queue manager to a later version of the product, and for restoring a queue manager to an earlier version of the product are detailed in this section.

"Migrating a queue manager to the latest version on IBM i" on page 176

Follow these instructions to migrate a queue manager on IBM i to the latest MQ version.

“Migrating IBM MQ on z/OS” on page 195
Migration tasks associated with z/OS are grouped in this section.

Windows *Migrating a queue manager to a later version on Windows*

On Windows platforms, follow these instructions to migrate a queue manager from an earlier version to a later version of IBM MQ.

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.

1. Create a migration plan; see “Planning to migrate IBM MQ to a later version on Windows” on page 112.
2. Review the IBM MQ system requirements for the latest version, including information about the versions of Windows that IBM MQ supports. See System Requirements for IBM MQ.
3. Back up your system before you install a later version of IBM MQ over an earlier version. Once you have started a queue manager you cannot revert to the previous version. If you must restore the system, you cannot recover any work, such as changes to messages and objects, performed by the later version of IBM MQ. For more information about backing up your system, see Backing up and restoring IBM MQ queue manager data.
4. Review any other installed SupportPacs for their applicability to the later version.
5. If you are running on a server with multiple IBM MQ installations, you must identify the installation. Make sure that the commands you enter run against the correct installation; see **`setmqenv`**.

About this task

To run a command, the operating system must find the command in an IBM MQ installation. For some commands, you must run the command from the installation that is associated with the correct queue manager. IBM 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 later 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 IBM WebSphere MQ 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. Log in as a user in `group mqm`.
2. Stop all applications using the IBM MQ installation.

   If you use the 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 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
      ```

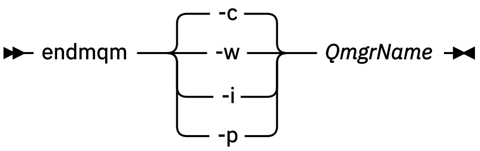
dspmq -o installation -o status displays the installation name and status of queue managers associated with all installations of IBM MQ.

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

b) Use the MQSC command **DISPLAY LSSTATUS** to list the status of listeners associated with a queue manager, as shown in the following example:

```
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 MQ libraries they have loaded. If they do not, you must find another way to force applications to release IBM 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 MQ. The application is not informed about queue managers in the current installation being shut down.

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

Note: “Applying maintenance level updates to multi-instance queue managers on Windows” on page 52 describes how to apply maintenance to a multi-instance queue manager. A multi-instance queue manager can continue to run on one server, while maintenance is applied to another server.

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

```
endmqlsr -m QMgrName
```

4. Back up the queue manager.

   Take copies of all the queue manager's data and log file directories, including all subdirectories, and also the `qm.ini` file and the registry entries. For more information, see Backing up and restoring IBM MQ queue manager data.
5. Stop the IBM WebSphere MQ or IBM MQ Service and exit the Service icon application.
6. Optional: If you are doing a single stage migration and you are migrating from IBM WebSphere MQ 7.0.1 Fix Pack 6 or later, optionally uninstall the current version of the product.

   Note, that you carry out this step only if you are doing a single stage migration; see “Migrating on Windows: single stage” on page 120.
7. Install the later version of IBM MQ.

   On Windows, you can do this either by using the Installation Launchpad or by using the **msiexec** command. For more information, see:

   - Modifying the installation using IBM MQ Installation Launchpad
   - Silently modifying an IBM MQ server installation using **msiexec**
8. Reenter domain, user ID, and password information

When the installation of the latest version completes, the **Prepare IBM MQ Wizard** starts automatically.

Where UAC is enabled: If you rerun the **Prepare IBM MQ Wizard**, ensure that the wizard is run with Administrator privilege, otherwise the wizard might fail.

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 re-create the default system objects.

You must start IBM MQ before you start any listeners.

What to do next

Complete the tasks in your migration plan, such as verifying the new code level and deploying new functions such as automatically restarting client connections.

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, migrate the other members of the cluster.

Important: You must migrate the publish/subscribe broker state before you migrate your IBM MQ system to IBM MQ 8.0, or later, as broker publish/subscribe migration is not supported in IBM MQ 8.0, or later.

Where to find downloadable installation images

“Migrating a queue manager in a high-availability configuration” on page 225
High-availability configurations of queue managers can increase the availability of IBM MQ applications. If a queue manager, or server fails, it is restarted automatically on another server. You can arrange for IBM MQ MQI client applications to automatically reconnect to the queue manager. Server applications can be configured to start when the queue manager starts.

“Migrating a queue manager cluster” on page 219
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. You must consider what the effect is of migrating some queue managers in a cluster, before all the queue managers are migrated.

“Queue manager migration” on page 80
After upgrading an installation, queue manager migration might be required. Migration takes place when you start a queue manager. You can remove an upgrade before you have started a queue manager. However, if you remove the upgrade after a queue manager has been started, the queue manager will not work.

IBM MQ - SupportPacs by Product

“Maintaining and migrating IBM MQ” on page 5
Maintenance, upgrade, and migration have three distinct meanings for IBM MQ. The definitions are described here. The following sections describe the various concepts associated with migration, followed by the various tasks needed; these tasks are platform-specific where needed.

“Migrating IBM MQ” on page 72
Migration is the conversion of programs and data to work with a new code level of IBM 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.

"Upgrading IBM MQ" on page 65
Upgrading is the process of taking an existing IBM MQ installation and upgrading to a new level of code.

Windows *Migrating on Windows: single stage*
Single-stage migration is the term used to describe replacing the only installation of IBM MQ on a server, with a later version of the product. Single stage migration is also known as *upgrading in place* or *in place upgrade*. Single-stage migration preserves existing scripts and procedures for running IBM 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

These topics guide you in deciding what other tasks you must perform to migrate queue managers and applications to the later version. For the precise sequence of commands to upgrade a queue manager to the later 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.

Attention: From IBM MQ 9.0, the `ccsid_part2.tbl` file replaces the existing `ccsid.tbl` file, used in previous versions of the product, to supply additional CCSID information.

The `ccsid_part2.tbl` file takes precedence over the `ccsid.tbl` file and:

- Allows you to add or modify CCSID entries
- Specify default data conversion
- Specify data for different command levels

The `ccsid_part2.tbl` is applicable to the following platforms only:

- Linux Linux - all versions
- Solaris Solaris
- Windows Windows

If you have added any of your own CCSID information into your existing `ccsid.tbl` file, you should copy this information into the new `ccsid_part2.tbl` file, if you want to take advantage of the new formats in your customizations

You should copy the required information, rather than move the information, so that your existing version of IBM MQ continues to work.

About this task

In the single-stage migration scenario, the installation of the later version of the product replaces an earlier version in the same installation location.

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 later version, automatically. Queue managers are automatically associated with the installation on the later 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 later version to be the primary installation, commands such as **`strmqm`** work without providing an explicit path to the command.

When you upgrade the earlier version to the later version, 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 MQ Explorer was not installed in an earlier installation, it is not stored in a later installation. If you want IBM MQ Explorer, select a custom installation, and select the IBM MQ Explorer feature on the **Features** panel. If you do not want IBM MQ Explorer, uninstall the IBM MQ Explorer feature by selecting a custom installation. Then clear the IBM MQ Explorer feature on the **Features** panel. For more information about how to uninstall features, see Modifying the installation using IBM MQ Installation Launchpad.

You can also migrate a queue manager to a later version of the product on a system where an earlier version has been uninstalled. In this case, 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 MQ installation.

   If you use the 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 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
   ```

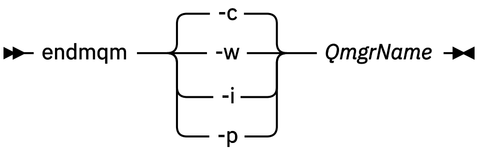
   `dspmq -o installation -o status` displays the installation name and status of queue managers associated with all installations of IBM MQ.

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

   b) Use the MQSC command **`DISPLAY LSSTATUS`** to list the status of listeners associated with a queue manager, as shown in the following example:

   ```
   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 MQ libraries they have loaded. If they do not, you must find another way to force applications to release IBM 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 MQ. The application is not informed about queue managers in the current installation being shut down.

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

Note: “Applying maintenance level updates to multi-instance queue managers on Windows” on page 52 describes how to apply maintenance to a multi-instance queue manager. A multi-instance queue manager can continue to run on one server, while maintenance is applied to another server.

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

```
endmqlsr -m QMgrName
```

4. Back up the queue manager.

   Take copies of all the queue manager's data and log file directories, including all subdirectories, and also the `qm.ini` file and the registry entries. For more information, see Backing up and restoring IBM MQ queue manager data.
5. Stop the IBM WebSphere MQ or IBM MQ Service and exit the Service icon application.
6. Optional: If you are migrating from IBM WebSphere MQ 7.0.1 Fix Pack 6 or later, optionally uninstall the current version of the product.
7. Upgrade the earlier version of the product to the later version in the same installation directory.

   A reason for installing into the same location is to simplify application migration. If you change the installation location, you might remove IBM 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.

   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 later version in place, or uninstall the earlier version, without deleting any queue managers, and install the later version in the same default location.

   On Windows, you can do this either by using the Installation Launchpad or by using the **msiexec** command. For more information, see:

   - Modifying the installation using IBM MQ Installation Launchpad
   - Silently modifying an IBM MQ server installation using **msiexec**

   On Windows, uninstalling the previous version of the product before you install the later version is optional.
8. Reenter domain, user ID, and password information

   When the installation of the latest version completes, the **Prepare IBM MQ Wizard** starts automatically.

   Where UAC is enabled: If you rerun the **Prepare IBM MQ Wizard**, ensure that the wizard is run with Administrator privilege, otherwise the wizard might fail.
9. Optional: Make the later version of the installation the primary installation.

   a) Run the **setmqinst** command

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

   Make the installation primary to avoid specifying a search path to run IBM MQ commands
10. Start the queue managers and applications.

   a) Run the **setmqm** command to associate the queue managers with `Inst_1`.

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

Notes:

- The **`setmqm`** step is optional only in the case where migration is from IBM WebSphere MQ 7.0.1 to a later release. In this case, the **`strmqm`** command automatically associates the queue manager with its own installation.
- If you are migrating between any other releases of the product, you must use **`setmqm`** to associate the queue managers with the new installation manually.

If you have multiple installations, note that queue managers that were configured to start automatically, and remain after uninstalling IBM WebSphere MQ 7.0.1, automatically start under any other existing IBM WebSphere MQ 7.1 (or later) 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 required installation before uninstalling IBM WebSphere MQ 7.0.1.

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

```
strmqm QM1
strmqm QM2
```

You must start IBM MQ before you start any listeners.

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.

At this point, when the queue manager data is migrated, you cannot revert to a previous release.

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

- When an application connects to a queue manager, the operating system searches its load path to load the IBM MQ library [2] . An IBM WebSphere MQ 7.1, 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 MQ loads the correct IBM 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 MQ installed.

Related concepts

"Queue manager coexistence" on page 95
Queue managers, with different names, can coexist on any server as long as they use the same IBM MQ installation. On z/OS, 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 98
You can install multiple copies of IBM MQ for UNIX, Linux, and Windows on the same server. These IBM MQ copies can be at the same or different version levels. This is called a multi-installation. Multi-

[2] On Windows, the IBM 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.

installation is particularly useful when you upgrade from one IBM MQ version to a later version, because it allows you to run the earlier version alongside the later version.

Related tasks

Migrating on Windows: side-by-side

Migrating on Windows: multi-stage

"Planning to migrate IBM MQ to a later version on Windows" on page 112

"Migrating a queue manager to a later version on UNIX and Linux" on page 147
On UNIX and Linux, follow these instructions to migrate a queue manager from an earlier version to a later version of IBM MQ.

"Migrating a queue manager to a later version on Windows" on page 117
On Windows platforms, follow these instructions to migrate a queue manager from an earlier version to a later version of IBM MQ.

Installing IBM MQ server on Windows

Associating a queue manager with an installation

Changing the primary installation

Choosing an installation name

"Migrating IBM MQ library loading to a later version on UNIX and Linux" on page 164
On UNIX and Linux, no change in the way IBM MQ libraries are loaded is normally required if you upgrade from an earlier version of the product to a later version by replacing an earlier version of the product with the later version, based on the single stage scenario. However, if you choose to take advantage of multi-installation in the later version of the product, based on the side-by-side or multi-stage migration scenarios, you might have to configure the runtime environment differently, for the operating system to load the later version of the IBM MQ library.

"Migrating IBM MQ library loading to a later version on Windows" on page 134
On Windows, no change in the way IBM MQ libraries are loaded is normally required if you upgrade from an earlier version of the product to the later version by replacing an earlier version of the product with the later version, based on the single stage scenario. However, if you choose to take advantage of multi-installation in the later version of the product, based on the side-by-side or multi-stage migration scenarios, you might have to configure the runtime environment differently, for the operating system to load the later version of the IBM MQ library.

Related reference

setmqenv

setmqinst

setmqm

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

Before you begin

If you are using IBM WebSphere MQ 7.0.1, you must ensure that you are running IBM WebSphere MQ 7.0.1 Fix Pack 6 or later before installing the later version of the product on the same server. For more information about IBM WebSphere MQ 7.0.1 fix packs, see Recommended Fixes for IBM MQ.

⚠ **Attention:** From IBM MQ 9.0, the `ccsid_part2.tbl` file replaces the existing `ccsid.tbl` file, used in previous versions of the product, to supply additional CCSID information.

The `ccsid_part2.tbl` file takes precedence over the `ccsid.tbl` file and:

- Allows you to add or modify CCSID entries
- Specify default data conversion

- Specify data for different command levels

The `ccsid_part2.tbl` is applicable to the following platforms only:

- Linux Linux - all versions
- Solaris Solaris
- Windows Windows

If you have added any of your own CCSID information into your existing `ccsid.tbl` file, you should copy this information into the new `ccsid_part2.tbl` file, if you want to take advantage of the new formats in your customizations

You should copy the required information, rather than move the information, so that your existing version of IBM MQ continues to work.

About this task

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

When you are ready to migrate the queue managers, and applications, to the later 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 later version.

Procedure

1. Install the later version in a different installation directory from the earlier version.
   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.
2. Uninstall the earlier version of the product.

   When uninstalling the earlier product, you must stop all queue managers and applications that have loaded an IBM 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 MQ libraries on the server.

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

   c) Uninstall the earlier version of the product.

      Stop all local IBM MQ applications
3. Make the later version of the installation the primary installation.
   a) Run the **`setmqinst`** command

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

   Make the installation primary to avoid specifying a search path to run IBM MQ commands

Use the dspmqinst command to discover the *Installation name*, or use the default value `Installation 1`.

Doing this means that you do not have to specify a search path on IBM MQ commands.

4. Start the queue managers and applications.

   - When an application connects to a queue manager, the operating system searches its load path to load the IBM MQ library [3] . An IBM WebSphere MQ 7.1, 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 MQ loads the correct IBM MQ library for the installation the queue manager is associated with.

   During this process you continue to use queue manager QM2 while you upgrade queue manager QM1 and you use queue manager QM1 while you upgrade QM2.

   Note that each queue manager needs to be stopped in order to be associated with the new installation.

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 MQ installed.

Related tasks

Migrating on Windows: single stage
Single-stage migration is the term used to describe replacing the only installation of IBM MQ on a server, with a later version of the product. Single stage migration is also known as *upgrading in place* or *in place upgrade*. Single-stage migration preserves existing scripts and procedures for running IBM 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.

Migrating on Windows: multi-stage

"Planning to migrate IBM MQ to a later version on Windows" on page 112

Uninstalling IBM MQ on Windows systems

Installing IBM MQ server on Windows

Associating a queue manager with an installation

Changing the primary installation

Choosing an installation name

"Queue manager coexistence" on page 95
Queue managers, with different names, can coexist on any server as long as they use the same IBM MQ installation. On z/OS, UNIX, Linux, and Windows, different queue managers can coexist on the same server and be associated with different installations.

"Migrating IBM MQ library loading to a later version on UNIX and Linux" on page 164
On UNIX and Linux, no change in the way IBM MQ libraries are loaded is normally required if you upgrade from an earlier version of the product to a later version by replacing an earlier version of the product with the later version, based on the single stage scenario. However, if you choose to take advantage of multi-installation in the later version of the product, based on the side-by-side or multi-stage migration scenarios, you might have to configure the runtime environment differently, for the operating system to load the later version of the IBM MQ library.

"Migrating IBM MQ library loading to a later version on Windows" on page 134
On Windows, no change in the way IBM MQ libraries are loaded is normally required if you upgrade from an earlier version of the product to the later version by replacing an earlier version of the product with the later version, based on the single stage scenario. However, if you choose to take advantage of multi-installation in the later version of the product, based on the side-by-side or multi-stage migration

[3] On Windows, the IBM 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.

scenarios, you might have to configure the runtime environment differently, for the operating system to load the later version of the IBM MQ library.

"Multi-installation queue manager coexistence on UNIX, Linux, and Windows" on page 98
You can install multiple copies of IBM MQ for UNIX, Linux, and Windows on the same server. These IBM MQ copies can be at the same or different version levels. This is called a multi-installation. Multi-installation is particularly useful when you upgrade from one IBM MQ version to a later version, because it allows you to run the earlier version alongside the later version.

setmqenv
setmqinst
setmqm

Windows *Migrating on Windows: multi-stage*
Multi-stage migration is the term used to describe running a later version of IBM MQ alongside an earlier version on the same server. After installing the later version alongside the earlier version, you can create new queue managers to verify the later installation, and develop new applications. At the same time, you can migrate queue managers and their associated applications from the earlier version to the later version. 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 7.0.1, you must ensure that you are running IBM WebSphere MQ 7.0.1 Fix Pack 6 or later before installing a later version of the product on the same server. For more information about IBM WebSphere MQ 7.0.1 fix packs, see Recommended Fixes for IBM MQ.

⚠ **Attention:** From IBM MQ 9.0, the `ccsid_part2.tbl` file replaces the existing `ccsid.tbl` file, used in previous versions of the product, to supply additional CCSID information.

The `ccsid_part2.tbl` file takes precedence over the `ccsid.tbl` file and:

- Allows you to add or modify CCSID entries
- Specify default data conversion
- Specify data for different command levels

The `ccsid_part2.tbl` is applicable to the following platforms only:

- Linux Linux - all versions
- Solaris Solaris
- Windows Windows

If you have added any of your own CCSID information into your existing `ccsid.tbl` file, you should copy this information into the new `ccsid_part2.tbl` file, if you want to take advantage of the new formats in your customizations

You should copy the required information, rather than move the information, so that your existing version of IBM MQ continues to work.

Note:

- If an application uses COM or ActiveX it can connect to any queue manager as long as there is a primary installation.
- If you are running the IBM MQ.NET monitor in transactional mode, the queue manager it connects to must be the primary installation.

You cannot migrate these applications to the later version until you uninstall the earlier version.

About this task

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

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

Note: When you have uninstalled the earlier version, and set the later version as a primary installation, 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.

Procedure

1. Install the later version in a different installation directory from the earlier version and verify the installation.
   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.

   - You might create new queue managers running the later version, and start to develop new applications before migrating applications from the earlier version.
2. Configure the operating system so that applications load the libraries for the later version of the product.
   a) Migrate queue managers one at a time.

      The first set of applications to load the libraries for the later version of the product 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 the applications load the later version libraries, IBM MQ automatically loads the libraries for the earlier version for those applications that connect to that version.

      You can 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.

   b) Migrate IBM MQ MQI client applications

      Some of the applications might be running as IBM MQ MQI client applications on another workstation. When you migrate a queue manager, clients connected to it continue to run without loading a client library for the later version.

      You can migrate these clients later, when you need to do so.

      Important: If any IBM MQ MQI client applications are using the library for the earlier version on the server, you must eventually migrate the clients to use the later version of the product before you uninstall the earlier version.
3. Migrate an application to load the new library for the later version:
   - Run **setmqenv** to modify the local path that is searched for IBM MQ libraries.
   - Relink applications with an additional runtime load path.

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:

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

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

4. Restart the queue manager and the applications that connect to it.
   a) Set up the local environment to the installation Inst_1.

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

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

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

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

   c) Run the **strmqm** command to start QM1 and migrate it to the later version.

   ```
   strmqm QM1
   strmqm QM2
   ```

   d) Restart application 1

   The application loads the later version library and connects to QM1, which is associated with the later version of the product.
5. Migrate all queue managers and applications to the later version.

   Repeat steps “2” on page 128 and “4” on page 129, when required, until all the queue managers and applications are migrated to the later version of the product.
6. Uninstall the earlier version of the product.

   When uninstalling the earlier product, you must stop all queue managers and applications that have loaded an IBM 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 MQ libraries on the server.
   b) Stop the queue managers and listeners on the server.
   c) Uninstall the earlier version of the product.

   Stop all local IBM MQ applications
7. Make Inst_1 the primary installation.
   a) Run the **setmqinst** command

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

   Note: Use the dspmqinst command to discover the *Installation name*, or use the default value Installation 1.

   You do not have to set up a search path to run IBM MQ commands from the primary installation.

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 MQ installed.

Now that you have uninstalled the earlier version of the product, and made the later 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 libraries for the later version. If you have only one installation of the later version of the product installed, it is not necessary to run **setmqenv** to run commands.

Related concepts

"Queue manager coexistence" on page 95
Queue managers, with different names, can coexist on any server as long as they use the same IBM MQ installation. On z/OS, 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 98
You can install multiple copies of IBM MQ for UNIX, Linux, and Windows on the same server. These IBM MQ copies can be at the same or different version levels. This is called a multi-installation. Multi-installation is particularly useful when you upgrade from one IBM MQ version to a later version, because it allows you to run the earlier version alongside the later version.

Related tasks

Migrating on Windows: single stage
Single-stage migration is the term used to describe replacing the only installation of IBM MQ on a server, with a later version of the product. Single stage migration is also known as *upgrading in place* or *in place upgrade*. Single-stage migration preserves existing scripts and procedures for running IBM 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.

Migrating on Windows: side-by-side

"Planning to migrate IBM MQ to a later version on Windows" on page 112

Installing IBM MQ server on Windows

Associating a queue manager with an installation

Changing the primary installation

Choosing an installation name

"Migrating IBM MQ library loading to a later version on UNIX and Linux" on page 164
On UNIX and Linux, no change in the way IBM MQ libraries are loaded is normally required if you upgrade from an earlier version of the product to a later version by replacing an earlier version of the product with the later version, based on the single stage scenario. However, if you choose to take advantage of multi-installation in the later version of the product, based on the side-by-side or multi-stage migration scenarios, you might have to configure the runtime environment differently, for the operating system to load the later version of the IBM MQ library.

"Migrating IBM MQ library loading to a later version on Windows" on page 134
On Windows, no change in the way IBM MQ libraries are loaded is normally required if you upgrade from an earlier version of the product to the later version by replacing an earlier version of the product with the later version, based on the single stage scenario. However, if you choose to take advantage of multi-installation in the later version of the product, based on the side-by-side or multi-stage migration scenarios, you might have to configure the runtime environment differently, for the operating system to load the later version of the IBM MQ library.

Related reference

setmqenv

setmqinst

setmqm

Windows *Reverting a queue manager to an earlier version on Windows*

On Windows platforms, you can revert a queue manager to an earlier version of the product from a later version, 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 reverting 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 the later version. For more information see Backing up and restoring IBM 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 revert the queue manager to the earlier version of the product 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 MQ installations, you must identify the installation. Make sure that the commands you enter run against the correct installation; see **`setmqenv`**.

About this task

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

Important: If the queue manager is a member of one or more IBM MQ clusters, you should also review and follow the steps described in Recovering a cluster queue manager.

Procedure

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

   If you use the 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 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
   ```

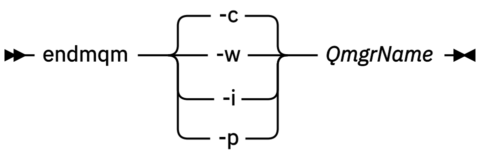
   `dspmq -o installation -o status` displays the installation name and status of queue managers associated with all installations of IBM MQ.

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

   b) Use the MQSC command **`DISPLAY LSSTATUS`** to list the status of listeners associated with a queue manager, as shown in the following example:

   ```
   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 MQ libraries they have loaded. If they do not, you must find another way to force applications to release IBM 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 MQ. The application is not informed about queue managers in the current installation being shut down.

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

shared library loaded.

Note: "Applying maintenance level updates to multi-instance queue managers on Windows" on page 52 describes how to apply maintenance to a multi-instance queue manager. A multi-instance queue manager can continue to run on one server, while maintenance is applied to another server.

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

```
endmqlsr -m QMgrName
```

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

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

   a) Uninstall the earlier installation.

   b) Reinstall the product from a manufacturing refresh.

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

   d) Restore the queue manager data from the backup taken before installing the later version.

5. Restart the earlier version queue manager.

What to do next

You might be reverting to a earlier version on a server with multiple IBM MQ installations. If one of the installations is primary, after reverting the earlier version that installation, by default, becomes the primary installation.

You must review how applications connect to an installation. After reverting to the earlier version, some applications might connect to the wrong installation.

Related concepts

BFGSS0023E errors and how to avoid them

Related tasks

Backing up and restoring a queue manager

Windows Migrating an IBM MQ MQI client on Windows

Before migrating an IBM MQ MQI client, create a migration plan. Stop all IBM MQ activity on the client workstation. Upgrade the IBM MQ MQI client installation. Make any essential configuration and application changes.

Related concepts

"IBM MQ MQI client migration" on page 81

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

Related tasks

"Migrating an IBM MQ MQI client to the latest version on IBM i" on page 190
Before migrating an IBM MQ MQI client, create a migration plan. Stop all IBM MQ activity on the client workstation. Upgrade the IBM MQ MQI client installation. Make any essential configuration and application changes.

"Migrating an IBM MQ MQI client on UNIX and Linux" on page 161
Before migrating an IBM MQ MQI client, create a migration plan. Stop all IBM MQ activity on the client workstation. Upgrade the IBM MQ MQI client installation. Make any essential configuration and application changes.

Windows *Migrating an IBM MQ MQI client to a later version on Windows*

Before migrating an IBM MQ MQI client on Windows platforms, create a migration plan. Stop all IBM MQ activity on the client workstation. Upgrade the IBM MQ MQI client installation. Make any essential configuration and application changes.

Before you begin

Before starting to migrate a client, create a migration plan. For guidance on what to include in the plan, see "Planning to migrate IBM MQ to a later version on Windows" on page 112.

About this task

IBM MQ MQI client migration is the process of converting IBM MQ MQI client configurations, and client and server channels from one version to another. Client migration is reversible. It is optional and manual on a client workstation and is required and automatic on the IBM MQ server.

You must upgrade an IBM 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.

Procedure

1. Review the IBM MQ system requirements for the later version of the product.

   See System Requirements for IBM MQ.
2. Review all the changes in IBM MQ that affect you.

   See "Changes that affect migration" on page 73.
3. End all IBM MQ activity on the workstation.
4. Upgrade the client.

   Select the appropriate option for your enterprise.

   - For a client installation on a workstation, see Installing an IBM MQ client on Windows.
   - For a client installation on an IBM MQ server, see Installing IBM MQ clients and servers on the same system.

What to do next

After upgrading the IBM MQ MQI client, you must check the client channel configuration, and verify that your IBM MQ MQI client applications work correctly with the later version of the product.

Related concepts

"IBM MQ MQI client migration" on page 81

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

Related tasks

"Planning to migrate IBM MQ to a later version on Windows" on page 112

Windows *Restoring an IBM MQ MQI client to an earlier version on Windows*

If you revert an IBM MQ MQI client from a later version of the product to an earlier version of the product, you must undo the configuration changes manually.

About this task

It is unusual to revert earlier IBM MQ MQI client libraries to a workstation. The principal tasks are listed in the following steps.

Procedure

1. End all IBM MQ activity on the workstation.
2. Uninstall the later version of the IBM MQ MQI client code.
3. Follow the client installation procedure for the platform to install the earlier version of the IBM MQ MQI client code.
4. If you configured a Client Connection Definition Table (CCDT) for a queue manager on a later version of the product, revert to using a table created by a queue manager on the earlier version.

   The CCDT must always be created by a queue manager on the same, or earlier, release to the client.

Windows Migrating IBM MQ library loading to a later version on Windows

On Windows, no change in the way IBM MQ libraries are loaded is normally required if you upgrade from an earlier version of the product to the later version by replacing an earlier version of the product with the later version, based on the single stage scenario. However, if you choose to take advantage of multi-installation in the later version of the product, based on the side-by-side or multi-stage migration scenarios, you might have to configure the runtime environment differently, for the operating system to load the later version of the IBM MQ library.

Before you begin

To migrate applications from an earlier version of the product to the later version, you must know how the operating system loads an IBM 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 MQ library that the application loads. The library name does not change from an earlier version of the product to the later version, although the contents of the library do.

Read "Multi-installation queue manager coexistence on UNIX, Linux, and Windows" on page 98 before starting this task.

Plan and install the later version of IBM MQ, and remember the installation name and whether the installation was set to primary.

About this task

To migrate an application from an earlier version of the product to the later version, you do not have to recompile or relink the application, because the IBM MQ libraries are compatible with later versions; see "Application compatibility and interoperability with later versions of IBM MQ" on page 105.

Windows searches numerous directories for load libraries, called DLLs; see Dynamic-Link Library Search Order. The build procedure for applications places the IBM MQ libraries to load before any other product libraries in the **`cl`** command. The IBM MQ `.lib` libraries must be in the PATH environment variable that 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 in the earlier release, then the effect of installing the later version of the product on the libraries that are loaded depends on which migration scenario that you are following:

Single-stage scenario

If you are replacing an earlier version of the product with the later version, based on the single stage scenario described in "Migrating on Windows: single stage" on page 120, you do not, in most cases, need to make any changes to the way IBM MQ libraries are loaded. The possible exception to this is if you changed the location of the libraries from the earlier version, or created symbolic links to the libraries.

Side-by-side and Multi-stage scenarios

If you have chosen a multi-installation approach to installing the later version of the product, based either on the side-by-side scenario described in "Migrating on Windows: side-by-side" on page 124, or the multi-stage migration scenario described in "Migrating on Windows: multi-stage" on page 127, you must investigate whether applications connecting to the later version of the product are linked to, and load libraries from, the correct installation and then modify the environment for the operating system to resolve IBM MQ dependencies for an application as appropriate. Typically, you can modify the runtime environment, rather than relink the application. You can use the following two commands to assist you in configuring the runtime environment:

- **setmqinst** sets the primary installation; see setmqinst.
- **setmqenv** initializes the command environment by setting environment variables; see setmqenv.

Table 9 on page 135 summarizes the actions needed for each of these scenarios.

Table 9. Windows configurations

Action **Scenario**	**Latest version replaces earlier version in the same location** **Single-stage**	**Latest version replaces earlier version in a different location** **Side-by-side**	**Latest version alongside earlier version** **Multi-stage**
setmqinst	**setmqinst** makes the later version installation primary. The global PATH is changed to point to the later version library and all Windows features work with the later version See note .		No. The later version installation can be primary, because an earlier version is installed.
No other configuration actions	Library loading works correctly. The global PATH contains the location of the later version libraries. Even if the later version installation is not primary, library loading works correctly. The later version libraries are in the same location as the earlier version 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 earlier version libraries. A local setting of PATH might override the global PATH that is set by **setmqinst**.	The library loading continues to work with the earlier version correctly, nothing works with the later version.

Table 9. Windows configurations (continued)

Action	Scenario	Latest version replaces earlier version in the same location Single-stage	Latest version replaces earlier version in a different location Side-by-side	Latest version alongside earlier version Multi-stage
setmqenv		Library loading works correctly. **setmqenv** sets the local PATH correctly.		Library loading works correctly, both for the earlier version and the later version. **setmqenv** sets the local PATH correctly for the later version. But the Windows features that depend on the global path do not work correctly with the later version See note . The correct earlier version is loaded, because the later version library loads the earlier version library for queue managers that have not been migrated from the earlier version.

Procedure

1. Consider which of the following questions apply to your configuration.
   - Did you follow the build procedure documented in the product documentation for the earlier version of the product? You might be following a different build procedure tailored to your development environment, or adapted from a development tool such as Microsoft Visual Studio.
   - How did you specify the load path for the earlier version?
   - Is the application is loaded by another environment, such as Eclipse, or an application server? You must modify the parameters that govern how the parent environment loads applications, not the way the parent environment is loaded.
   - Do the functions performed by an application require that the queue manager it connects to is associated with the primary installation?
   - What constraints and requirements do you have on how the load path is specified in the later version? Security rules might restrict the use of `LD_LIBRARY_PATH`.
   - Is the later version of the product installed alongside the earlier version? If IBM WebSphere MQ 7.0.1 is installed:
     - You cannot make a later installation primary.
     - You cannot install the later version in the default installation path, that was referenced by applications in IBM WebSphere MQ 7.0.1.
2. Identify the installation of the later version of the product, from which the operating system is going to load IBM MQ libraries:
   - If you have a multiple installations of the later versions to load from a server, IBM MQ checks that the installation the library was loaded from is the installation that is associated with any queue manager the application calls. IBM MQ loads the correct library if the wrong library is loaded. It is necessary to configure only one runtime environment for all IBM MQ applications.
   - 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 an earlier version installation to the later version, a link path to the earlier version installation now points to an installation containing the later version. Applications that have a fixed linkage path to the earlier version installation now load the libraries for the later installation. 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 an installation of the later version.
- If an application uses COM or ActiveX it can connect to any queue manager as long as there is a primary installation.

  Note: If an earlier version of the product is installed, COM or ActiveX server applications connect to queue managers associated only with the earlier installation. COM or ActiveX client applications are not affected by the limitation.
- If you are running the IBM 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 installations of the later version of the product, you must decide which installation to make primary, if you have chosen to make any primary. As long as applications load IBM MQ libraries from one of the later version installations, such as the primary installation, they can connect to queue managers associated with any other later version 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 later version IBM MQ library is not on the same path as the earlier release, you must recompile, or configure and relink your application to load the later version libraries.

Related concepts

Features that can be used only with the primary installation on Windows

Related tasks

Changing the primary installation

Connecting applications in a multiple installation environment

“Migrating IBM MQ library loading to a later version on UNIX and Linux” on page 164

On UNIX and Linux, no change in the way IBM MQ libraries are loaded is normally required if you upgrade from an earlier version of the product to a later version by replacing an earlier version of the product with the later version, based on the single stage scenario. However, if you choose to take advantage of multi-installation in the later version of the product, based on the side-by-side or multi-stage migration scenarios, you might have to configure the runtime environment differently, for the operating system to load the later version of the IBM MQ library.

Related reference

“Coexistence” on page 94

Queue managers, with different names, can coexist on any server as long as they use the same IBM MQ installation. On z/OS, 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 from a single instance to a multi-instance queue manager on Windows

To migrate a single instance queue manager, to a multi-instance queue manager, on Windows platforms, 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. For a list of tested environments, see Testing statement for IBM MQ multi-instance queue manager file systems. Other environments might work; a test tool is provided with IBM 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 statement for IBM MQ multi-instance queue manager file systems. 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 IBM 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.

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 (IBM WebSphere MQ 7.0.1 onwards).

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 *`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`.

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 IBM WebSphere MQ 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 following 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 138.

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.

     a. Modify the log registry key:

     ```
     HKEY_LOCAL_MACHINE\SOFTWARE\IBM\WebSphere
     MQ\Installation\MQ_INSTALLATION_NAME\Configuration\QueueManager\QMgrName\Log
     "LogPath"="share\\logs\\QMgrName\\"
     ```

     b. Modify the Prefix registry key:

     ```
     HKEY_LOCAL_MACHINE\SOFTWARE\IBM\WebSphere
     MQ\Installation\MQ_INSTALLATION_NAME\Configuration\QueueManager\QMgrName
     "Prefix"="share\\data"
     ```

   where *`QMgrName`* is the representation of the queue manager name in the existing registry key on Windows. *`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.

   ```
   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 the following connection name:

```
echo DISPLAY CHANNEL(ENGLAND) CONNAME | runmqsc QM1
```

```
5724-H72 (C) Copyright IBM Corp. 1994, 2024.  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 IBM 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
Windows domains and multi-instance queue managers
Working with services

Related tasks

Backing up queue manager data
Changing configuration information on UNIX, Linux, and Windows systems
Moving a queue manager to MSCS storage
Verifying shared file system locking

Related reference

amqmfsck (file system check)

Related information

Testing a shared file system for compatibility with IBM MQ Multi-instance Queue Managers

Testing statement for IBM MQ multi-instance queue manager file systems

Windows Reverting to a single-instance queue manager on Windows

Revert a multi-instance queue manager to a single instance queue manager, on Windows platforms, 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:

   ```
   endmqm -w QMgrName
   ```

2. Stop the active queue manager instance.

   On the server running the active instance:

   ```
   endmqm -w (QMgrName)
   ```

3. Restart the queue manager, preventing standbys.

   On the server that is going to run the queue manager:

   ```
   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 IBM 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 Windows" on page 138 to create a multi-instance queue manager.

Windows Migrating MQ Telemetry on Windows

Follow these instructions to migrate your existing installation of MQ Telemetry to a later version of the product on Windows.

Before you begin

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

About this task

The telemetry server is included in the product as an optional installation.

For IBM WebSphere MQ 7.5, the Client Software Development Kit (the telemetry clients) is also included in the optional installation. From IBM MQ 8.0 onwards, the Client Software Development Kit is no longer supplied as part of the product. Instead, the current version of the SDK is available as IBM Messaging Telemetry Clients SupportPac.

Because MQ Telemetry is a component of IBM WebSphere MQ 7.1 and later, MQ Telemetry can either be installed with the main product, or installed after the main product has been installed. When you upgrade from a previous version of the product, you must download and use the latest version of the Client Software Development Kit.

After the successful upgrade, Windows systems retain the telemetry data in the installation directory of the product, for example: `C:\Program Files (x86)\IBM\WebSphere MQ`. Telemetry data is migrated to the later version of the product when the queue manager is started again.

Procedure

1. Create a migration plan.

   See "Planning to migrate IBM MQ to a later version on Windows" on page 112.
2. Migrate your queue managers to the later release.
3. Install MQ Telemetry.
4. Verify that the MQ Telemetry installation was successful. See Verifying the installation of MQ Telemetry .

Results

Message AMQ4616 indicates completion of the task. The existing MQTT channels and previous subscriptions are still present.

Related tasks

Installing IBM MQ - overview

Installing MQ Telemetry

Verifying the installation of MQ Telemetry

Verifying the installation of MQ Telemetry by using IBM MQ Explorer

Windows Migrating an MSCS configuration on Windows

Migrate queue managers in a Microsoft Cluster Service (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 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 an earlier version of the product to the latest version” on page 143, shows this procedure applied to a four-node cluster.

Procedure

1. Modify the possible owners of the IBM MQ resource to encompass only the Active node or nodes. With no owners assigned to Passive nodes, the IBM MQ resource that is being migrated cannot be activated.
2. Ensure that the group containing the IBM 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 MQ DLLs that have been registered.
4. Migrate the selected node by following the standard instructions in “Migrating a queue manager to a later version on Windows” on page 117. 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 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 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 an earlier version of the product to the latest version

The example in Table 10 on page 144 illustrates the steps involved in migrating a four-node MSCS cluster.

In the example IBM 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 an earlier version of the product to the latest version.

1. Select node 1 to be migrated and convert it into a Passive node with no running IBM MQ resources.
2. Modify the possible owners of the group containing the IBM MQ resources, to encompass only the required online nodes. Failover does not attempt to switch IBM 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 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 MQ libraries that have been registered for MSCS. The node goes offline.

Step 2
Migrate IBM MQ from an earlier version of the product to the latest version

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 the latest version. They are still doing no work, as they are not possible owners of any of the IBM MQ resource groups.

Step 5
Migrate the cluster from running an earlier version of the product to the latest version. 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 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 IBM MQ resource must never failover to a node running a back level version of the product.

Note: If you must revert IBM MQ to an earlier version, the IBM MQ resources must be removed from MSCS control, before performing an uninstallation of IBM MQ

Step 6

Migrate node 3 to the latest version.

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 10. 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	Earlier version	Earlier version	*Latest version*	Latest version	Latest version	Latest version	Latest version	Latest version
	Groups	QMA					*QMC, QMA*	*QMA*	QMA
Node 2	State	Online	Online	Online	Online	Online	Online	Online	Online
	Version	Earlier version	Earlier version	Earlier version	Earlier version	*Latest version*	Latest version	Latest version	Latest version
	Groups	QMB	QMB	QMB	QMB		*QMD, QMB*	QMD, QMB	*QMB*
Node 3	State	Online	Online	Online	Online	Online	Online	Online	Online
	Version	Earlier version	Earlier version	Earlier version	Earlier version	Earlier version	Earlier version	*Latest version*	Latest version
	Groups	QMC	QMC, QMA	QMC, QMA	QMC, QMA	QMC, QMA		*QMC*	QMC
Node 4	State	Online	Online	Online	Online	Online	Online	Online	Online
	Version	Earlier version	Earlier version	Earlier version	Earlier version	Earlier version	Earlier version	Earlier version	*Latest version*
	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 tasks

“Migrating a queue manager in a high-availability configuration” on page 225
High-availability configurations of queue managers can increase the availability of IBM MQ applications. If a queue manager, or server fails, it is restarted automatically on another server. You can arrange for IBM MQ MQI client applications to automatically reconnect to the queue manager. Server applications can be configured to start when the queue manager starts.

Related reference

Windows: MSCS restriction with multiple installations

Windows V 9.1.0 Migrating logs to an Advanced Format disk on Windows

An Advanced Format disk is one that has 4096 bytes per sector. The following is applicable only to the Windows platform as Advanced Format disks can be used on other platforms, without carrying out a migration procedure.

⚠ **Attention:** On Windows, prior to IBM MQ 9.1.0, (or prior to IBM MQ 9.0.4 if you are a Continuous Delivery user), IBM MQ does not support Advanced Format disks

Note the following:

- A migrated log can be used on any disk whether or not it is Advanced Format.
- If you are not using an Advanced Format disk, you do not need to migrate the log of your queue manager.
- Queue managers that are created at IBM MQ 9.1.0 (or at IBM MQ 9.0.4 or later if you are a Continuous Delivery user) can be used on an Advanced Format disk without being migrated.
- If you use a queue manager that was created before IBM MQ 9.1.0 (or before IBM MQ 9.0.4 if you are a Continuous Delivery user) on a native Advanced Format disk, without migrating the queue manager first, the queue manager will not start
- It is possible to start a queue manager on an Advanced Format disk in emulation mode without migration. However IBM MQ log writes will not be on 4k boundaries and so the queue manager will not have data integrity. Once the logs have been migrated, an Advanced Format disk in emulation mode is reliable.
- If you are not sure whether your disk is Advanced Format or not, use the Windows utility **`fsutil`** to find out.
- The Advanced Format disks that require you to migrate your log, include 4k native disks and 512-byte Emulation disks.
- Using **`migmqlog`** to change from linear logging to circular logging, or from circular logging to linear logging, also migrates the log so that the log can be used on an Advanced Format disk.

Related tasks

“Migrating logs on UNIX, Linux, and Windows” on page 233
From IBM MQ 9.1.0 you can migrate a circular log to a linear log, or from a linear log to a circular log.

Linux UNIX Migrating IBM MQ on UNIX and Linux

Migration tasks associated with UNIX and Linux platforms are grouped in this section.

About this task

This topic lists the various steps you need to take to migrate to, or migrate from, the latest version of the IBM MQ product.

See also, “Migrating from one Continuous Delivery release to another” on page 110, if you are migrating a Continuous Delivery release of the product.

If you want to migrate replicated data queue managers, follow the instructions in “Migrating replicated data queue managers” on page 228.

Related concepts

"Migration concepts and methods" on page 76
An overview of the various concepts and methods for migrating from one release of the product to another.

Related tasks

"Migrating IBM MQ on IBM i" on page 174
IBM MQ migration tasks associated with IBM i are grouped in this section.

"Migrating IBM MQ on Windows" on page 111
IBM MQ migration tasks associated with Windows platforms are grouped in this section.

"Migrating IBM MQ on z/OS" on page 195
Migration tasks associated with z/OS are grouped in this section.

Related reference

"Changes that affect migration" on page 73

Linux UNIX Planning to migrate IBM MQ to a later version on UNIX and Linux

Before migrating IBM MQ to a later version on UNIX and Linux, review the system requirements information, and the information about any changes that might affect migration, then create a migration plan.

Before you begin

If there are concepts about migration you do not understand, see "Migration concepts and methods" on page 76.

If you are migrating to IBM MQ 9.1 from IBM WebSphere MQ 7.5 or earlier, you must first migrate to an interim version. See Migration paths.

Before you use side-by-side or multi-stage migration to migrate from IBM WebSphere MQ 7.0.1, check that your IBM WebSphere MQ 7.0.1 installation is at Fix Pack 6 or later.

About this task

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

Procedure

1. Review the IBM MQ system requirements for the later version of the product.

   See System Requirements for IBM MQ.
2. Decide whether to run the earlier version and the later version of the product on the same server, and also which migration method you want to use.

   Choices are single-stage migration, side-by-side migration, or multi-stage migration. See "Migration methods on IBM MQ for Multiplatforms" on page 84.
3. Review all the changes in IBM MQ that affect you.

   See "Changes that affect migration" on page 73.
4. Review performance changes.

   See MQ Performance documents.
5. Review the readme file for the later version of IBM MQ.

   See IBM MQ, WebSphere MQ, and MQSeries product readmes.
6. Plan the sequence and timing of queue manager migrations.

- 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 “Migrating a queue manager in a high-availability configuration” on page 225.

7. Plan to migrate your queue manager to the later version.

   See “Migrating a queue manager to a later version on UNIX and Linux” on page 147.

   Backing up queue manager data is part of the queue manager migration task. An alternative approach is to install and configure a new server, then test the later version with a new queue manager on the new server. When you are ready to go into production on the later version, 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.

   From IBM MQ 9.1, a suffix letter, indicating the severity of a message (I, W, E, S or T) is appended to IBM MQ diagnostic (AMQ) messages. Existing scripts looking for error codes without the severity will fail. For example, existing scripts looking for error matching to AMQ7468 will fail. You must update the scripts to look for error codes with the severity suffix added (for example, AMQ7468I). For more information, see IBM MQ messages on Multiplatforms.
9. Decide on what regression tests to perform before putting the queue manager into production on the later version. Include in your regression tests the procedures and applications you identified in previous steps.
10. Plan to migrate your IBM MQ MQI client installations to the later version.
11. Plan to migrate your client and server applications to use new functions in the later version.
12. Decide which downloadable images you require for the migration.

    For more information, see Where to find downloadable installation images.
13. Decide which downloadable images you require for the migration.

    For more information, see Where to find downloadable installation images.

Linux UNIX Migrating a queue manager on UNIX and Linux

The procedures for migrating a queue manager to a later version of the product, and for restoring a queue manager to an earlier version of the product are detailed in this section.

Related tasks

“Migrating a queue manager to the latest version on IBM i” on page 176
Follow these instructions to migrate a queue manager on IBM i to the latest MQ version.

“Migrating a queue manager on Windows” on page 116
The procedures for migrating a queue manager to a later version of the product, and for restoring a queue manager to an earlier version of the product are detailed in this section.

“Migrating IBM MQ on z/OS” on page 195
Migration tasks associated with z/OS are grouped in this section.

Linux UNIX *Migrating a queue manager to a later version on UNIX and Linux*

On UNIX and Linux, follow these instructions to migrate a queue manager from an earlier version to a later version of IBM MQ.

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.

1. The upgrade from the earlier version to the latesr version of the product requires a full migration of queue managers. Create a migration plan. Use the planning task, “Planning to migrate IBM MQ to a later version on UNIX and Linux” on page 146, as a guide.
2. Review the IBM MQ system requirements for the later version; see System Requirements for IBM MQ.
3. Back up your system before you install a later version of IBM MQ over an earlier version. Once you have started a queue manager you cannot revert to the previous version. If you must restore the system, you cannot recover any work, such as changes to messages and objects, performed by the later version of IBM MQ. For more information about backing up your system, see Backing up and restoring IBM MQ queue manager data.
4. Review any other installed SupportPacs for their applicability to the later version.
5. If you are running on a server with multiple IBM MQ installations, you must identify the installation. Make sure that the commands you enter run against the correct installation; see **setmqenv**.

About this task

To run a command, the operating system must find the command in an IBM MQ installation. For some commands, you must run the command from the installation that is associated with the correct queue manager. IBM 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 IBM WebSphere MQ 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. Log in as a user in `group mqm`.
2. Stop all applications using the IBM MQ installation.

   If you use the 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 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
      ```

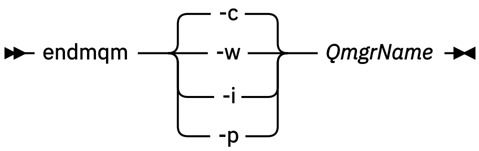
      dspmq -o installation -o status displays the installation name and status of queue managers associated with all installations of IBM MQ.

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

   b) Use the MQSC command **DISPLAY LSSTATUS** to list the status of listeners associated with a queue manager, as shown in the following example:

      ```
      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 MQ libraries they have loaded. If they do not, you must find another way to force applications to release IBM 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 MQ. The application is not informed about queue managers in the current installation being shut down.

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

Note: "Applying maintenance level updates to multi-instance queue managers on Linux" on page 35 describes how to apply maintenance to a multi-instance queue manager. A multi-instance queue manager can continue to run on one server, while maintenance is applied to another server.

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

```
endmqlsr -m QMgrName
```

4. Back up the queue manager.

   Take copies of all the queue manager's data and log file directories, including all subdirectories, and also the `qm.ini` file. For more information, see Backing up and restoring IBM MQ queue manager data.
5. Log in as root.
6. Optionally uninstall the current version of IBM MQ.

   Note, that you carry out this step only if you are doing a single stage migration; see "Migrating on UNIX and Linux: single-stage" on page 150.
7. Install the later version of the product.

   See the appropriate topic for the platform that your enterprise uses:

   AIX Installing IBM MQ Server on AIX .

   Linux Installing IBM MQ Server on Linux .

   Solaris Installing IBM MQ Server on Solaris.
8. Move the queue manager to the new IBM MQ installation.

   You need to carry out this step, only if you did not uninstall your current version of the product.

   See "Migrating on UNIX and Linux: side-by-side" on page 153 or "Migrating on UNIX and Linux: multi-stage" on page 156 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 re-create the default system objects.

You must start IBM MQ before you start any listeners.

Backing up and restoring a queue manager

Linux UNIX *Migrating on UNIX and Linux: single-stage*
Single-stage migration is the term used to describe replacing the only installation of IBM MQ on a server, with a later release. Single stage migration is also known as *upgrading in place* or *in place upgrade*. Single-stage migration preserves existing scripts and procedures for running IBM 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

Attention: From IBM MQ 9.0, the `ccsid_part2.tbl` file replaces the existing `ccsid.tbl` file, used in previous versions of the product, to supply additional CCSID information.

The `ccsid_part2.tbl` file takes precedence over the `ccsid.tbl` file and:

- Allows you to add or modify CCSID entries
- Specify default data conversion
- Specify data for different command levels

The `ccsid_part2.tbl` is applicable to the following platforms only:

- Linux Linux - all versions
- Solaris Solaris
- Windows Windows

If you have added any of your own CCSID information into your existing `ccsid.tbl` file, you should copy this information into the new `ccsid_part2.tbl` file, if you want to take advantage of the new formats in your customizations

You should copy the required information, rather than move the information, so that your existing version of IBM MQ continues to work.

About this task

In the single-stage migration scenario, the installation of the later version of the product replaces an earlier version in the same installation location.

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 later version, automatically. Queue managers are automatically associated with the installation on the later 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 later version to be the primary installation, commands such as **`strmqm`** work without providing an explicit path to the command.

You can also migrate a queue manager to a later version of the product on a system where an earlier version has been uninstalled. In this case, the queue manager data must have been retained, or restored from a backup.

Procedure

1. Stop local IBM MQ applications.
2. Stop all the queue managers and listeners.
3. Uninstall any fix packs you have installed from the previous IBM MQ version.
4. Upgrade the earlier version of the product to the later version in the same installation directory.

   - A reason for installing into the same location is to simplify application migration. If you change the installation location, you might remove IBM 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.
   - Linux UNIX The default installation path is specified as a load path in the IBM MQ build scripts for UNIX and Linux. After installation of the later version, the load libraries of the later version of IBM 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 later version.

   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.

   AIX On AIX there is no option to set the installation name, *Installation1* is set by default.

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

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

   On the following platforms, you do not have to uninstall a previous version of the product:

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

   AIX If `mqm.xr.clients` and `mqm.txclient.rte` file sets from earlier versions are installed, you must uninstall these file sets from the earlier versions.

   On the following platforms, you must uninstall the previous version of the product:

   - Linux Linux
   - Solaris Solaris

5. Optional: Make the later version of the installation the primary installation.

   a) Run the **`setmqinst`** command

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

   - Make the installation primary to avoid specifying a search path to run IBM MQ commands.
   - If there is a primary installation, UNIX and Linux applications that expect to find the IBM MQ library in `/usr/lib`, find a symbolic link to the library in `/usr/lib/32`[4]. `/usr/lib/32` is normally in the default search path. It is also specified as a load path in the IBM MQ build scripts for UNIX and Linux.

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

- It is sufficient to link applications only to `/usr/lib`. With a primary installation of the later version of the product defined on the server, an application can connect to any queue manager associated with any installation on the server. IBM MQ loads the correct library for the application.

6. Start the queue managers and applications.

   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
   ```

   Note: If you are migrating between any releases of the product, you must use **`setmqm`** to associate the queue managers with the new installation manually.

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

   ```
   strmqm QM1
   strmqm QM2
   ```

   At this point, queue manager data is migrated and you cannot revert to a previous release.

- When an application connects to a queue manager, the operating system searches its load path to load the IBM MQ library. The 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 MQ loads the correct IBM 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 MQ installed.

Related concepts

“Queue manager coexistence” on page 95
Queue managers, with different names, can coexist on any server as long as they use the same IBM MQ installation. On z/OS, 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 98
You can install multiple copies of IBM MQ for UNIX, Linux, and Windows on the same server. These IBM MQ copies can be at the same or different version levels. This is called a multi-installation. Multi-installation is particularly useful when you upgrade from one IBM MQ version to a later version, because it allows you to run the earlier version alongside the later version.

Related tasks

Migrating on UNIX and Linux: side-by-side
Migrating on UNIX and Linux: multi-stage
“Planning to migrate IBM MQ to a later version on Windows” on page 112
“Migrating a queue manager to a later version on UNIX and Linux” on page 147
On UNIX and Linux, follow these instructions to migrate a queue manager from an earlier version to a later version of IBM MQ.

“Migrating a queue manager to a later version on Windows” on page 117
On Windows platforms, follow these instructions to migrate a queue manager from an earlier version to a later version of IBM MQ.

Installing IBM MQ server on AIX
Installing IBM MQ server on Linux
Installing IBM MQ server on Solaris
Associating a queue manager with an installation
Changing the primary installation

Choosing an installation name

“Migrating IBM MQ library loading to a later version on UNIX and Linux” on page 164
On UNIX and Linux, no change in the way IBM MQ libraries are loaded is normally required if you upgrade from an earlier version of the product to a later version by replacing an earlier version of the product with the later version, based on the single stage scenario. However, if you choose to take advantage of multi-installation in the later version of the product, based on the side-by-side or multi-stage migration scenarios, you might have to configure the runtime environment differently, for the operating system to load the later version of the IBM MQ library.

“Migrating IBM MQ library loading to a later version on Windows” on page 134
On Windows, no change in the way IBM MQ libraries are loaded is normally required if you upgrade from an earlier version of the product to the later version by replacing an earlier version of the product with the later version, based on the single stage scenario. However, if you choose to take advantage of multi-installation in the later version of the product, based on the side-by-side or multi-stage migration scenarios, you might have to configure the runtime environment differently, for the operating system to load the later version of the IBM MQ library.

Related reference

setmqenv

setmqinst

setmqm

Linux UNIX *Migrating on UNIX and Linux: side-by-side*

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

Before you begin

Attention: From IBM MQ 9.0, the `ccsid_part2.tbl` file replaces the existing `ccsid.tbl` file, used in previous versions of the product, to supply additional CCSID information.

The `ccsid_part2.tbl` file takes precedence over the `ccsid.tbl` file and:

- Allows you to add or modify CCSID entries
- Specify default data conversion
- Specify data for different command levels

The `ccsid_part2.tbl` is applicable to the following platforms only:

- Linux Linux - all versions
- Solaris Solaris
- Windows Windows

If you have added any of your own CCSID information into your existing `ccsid.tbl` file, you should copy this information into the new `ccsid_part2.tbl` file, if you want to take advantage of the new formats in your customizations

You should copy the required information, rather than move the information, so that your existing version of IBM MQ continues to work.

About this task

In the side-by-side migration scenario, you install the later version of IBM MQ alongside queue managers that continue to be associated with an earlier version.

When you are ready to migrate the queue managers, and applications, to the later 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 later version.

Procedure

1. Install the later version in a different installation directory from the earlier version.

   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.

   AIX On AIX there is no option to set the installation name, *Installation1* is set by default.

   b) Verify the installation.

   Run the installation verification procedures and your own tests.

2. Uninstall the earlier version of the product.

   When uninstalling the earlier product, you must stop all queue managers and applications that have loaded an IBM 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 MQ libraries on the server.

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

   c) Uninstall the earlier version of the product.

   Stop all local IBM MQ applications

3. Make the later version of the installation the primary installation.

   a) Run the **setmqinst** command

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

   - Make the installation primary to avoid specifying a search path to run IBM MQ commands.
   - If there is a primary installation, UNIX and Linux applications that expect to find the IBM 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 MQ build scripts for UNIX and Linux.
   - It is sufficient to link applications only to `/usr/lib`. With a primary installation of the later version of the product defined on the server, an application can connect to any queue manager associated with any installation on the server. IBM MQ loads the correct library for the application.

   Use the dspmqinst command to discover the *`Installation name`*, or use the default value `Installation 1`.

   Doing this means that you do not have to specify a search path on IBM MQ commands.

4. Start the queue managers and applications.

   - When an application connects to a queue manager, the operating system searches its load path to load the IBM MQ library. The 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 MQ loads the correct IBM MQ library for the installation the queue manager is associated with.

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

During this process you continue to use queue manager QM2 while you upgrade queue manager QM1 and you use queue manager QM1 while you upgrade QM2.

Note that each queue manager needs to be stopped in order to be associated with the new installation.

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 MQ installed.

Related tasks

Migrating on UNIX and Linux: single-stage
Single-stage migration is the term used to describe replacing the only installation of IBM MQ on a server, with a later release. Single stage migration is also known as *upgrading in place* or *in place upgrade*. Single-stage migration preserves existing scripts and procedures for running IBM 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.

Migrating on UNIX and Linux: multi-stage
"Planning to migrate IBM MQ to a later version on Windows" on page 112
Installing IBM MQ server on AIX
Uninstalling IBM MQ on AIX
Installing IBM MQ server on Linux
Uninstalling IBM MQ on Linux
Uninstalling IBM MQ on Solaris
Installing IBM MQ server on Solaris
Associating a queue manager with an installation
Changing the primary installation
Choosing an installation name
"Queue manager coexistence" on page 95
Queue managers, with different names, can coexist on any server as long as they use the same IBM MQ installation. On z/OS, UNIX, Linux, and Windows, different queue managers can coexist on the same server and be associated with different installations.

"Migrating IBM MQ library loading to a later version on UNIX and Linux" on page 164
On UNIX and Linux, no change in the way IBM MQ libraries are loaded is normally required if you upgrade from an earlier version of the product to a later version by replacing an earlier version of the product with the later version, based on the single stage scenario. However, if you choose to take advantage of multi-installation in the later version of the product, based on the side-by-side or multi-stage migration scenarios, you might have to configure the runtime environment differently, for the operating system to load the later version of the IBM MQ library.

"Migrating IBM MQ library loading to a later version on Windows" on page 134
On Windows, no change in the way IBM MQ libraries are loaded is normally required if you upgrade from an earlier version of the product to the later version by replacing an earlier version of the product with the later version, based on the single stage scenario. However, if you choose to take advantage of multi-installation in the later version of the product, based on the side-by-side or multi-stage migration scenarios, you might have to configure the runtime environment differently, for the operating system to load the later version of the IBM MQ library.

"Multi-installation queue manager coexistence on UNIX, Linux, and Windows" on page 98
You can install multiple copies of IBM MQ for UNIX, Linux, and Windows on the same server. These IBM MQ copies can be at the same or different version levels. This is called a multi-installation. Multi-installation is particularly useful when you upgrade from one IBM MQ version to a later version, because it allows you to run the earlier version alongside the later version.

setmqenv
setmqinst
setmqm

Linux UNIX *Migrating on UNIX and Linux: multi-stage*

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

Before you begin

⚠ **Attention:** From IBM MQ 9.0, the `ccsid_part2.tbl` file replaces the existing `ccsid.tbl` file, used in previous versions of the product, to supply additional CCSID information.

The `ccsid_part2.tbl` file takes precedence over the `ccsid.tbl` file and:

- Allows you to add or modify CCSID entries
- Specify default data conversion
- Specify data for different command levels

The `ccsid_part2.tbl` is applicable to the following platforms only:

- Linux Linux - all versions
- Solaris Solaris
- Windows Windows

If you have added any of your own CCSID information into your existing `ccsid.tbl` file, you should copy this information into the new `ccsid_part2.tbl` file, if you want to take advantage of the new formats in your customizations

You should copy the required information, rather than move the information, so that your existing version of IBM MQ continues to work.

Note:

- If an application uses COM or ActiveX it can connect to any queue manager as long as there is a primary installation.
- If you are running the IBM MQ.NET monitor in transactional mode, the queue manager it connects to must be the primary installation.

You cannot migrate these applications to the later version until you uninstall the earlier version.

About this task

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

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

Note: When you have uninstalled the earlier version, and set the later version as a primary installation, 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.

Procedure

1. Install the later version in a different installation directory from the earlier version and verify the installation.

    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.

    AIX On AIX there is no option to set the installation name, *Installation1* is set by default.

    b) Verify the installation.

    Run the installation verification procedures and your own tests.

    - You might create new queue managers running the later version, and start to develop new applications before migrating applications from the earlier version.

2. Configure the operating system so that applications load the libraries for the later version of the product.

    a) Migrate queue managers one at a time.

    The first set of applications to load the libraries for the later version of the product 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 the applications load the later version libraries, IBM MQ automatically loads the libraries for the earlier version for those applications that connect to that version.

    You can 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.

    b) Migrate IBM MQ MQI client applications

    Some of the applications might be running as IBM MQ MQI client applications on another workstation. When you migrate a queue manager, clients connected to it continue to run without loading a client library for the later version.

    You can migrate these clients later, when you need to do so.

    Important: If any IBM MQ MQI client applications are using the library for the earlier version on the server, you must eventually migrate the clients to use the later version of the product before you uninstall the earlier version.

3. Migrate an application to load the new library for the later version:

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

      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:

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

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

    The `-k` option inserts the path to the IBM MQ load libraries at the start of the `LD_LIBRARY_PATH` environment variable, and adds the variable to the local environment; see "Loading IBM MQ libraries" on page 99.

    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**.

4. Restart the queue manager and the applications that connect to it.
   a) Set up the local environment to the installation `Inst_1`.

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

      The `-s` option sets up the environment for the installation that runs the **setmqenv** command.
   b) Run the **setmqm** command to associate `QM1` with `Inst_1`.

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

   c) Run the **strmqm** command to start `QM1` and migrate it to the later version.

      ```
      strmqm QM1
      strmqm QM2
      ```

   d) Restart application 1

      The application loads the later version library and connects to `QM1`, which is associated with the later version of the product.
5. Migrate all queue managers and applications to the later version.

   Repeat steps "2" on page 157 and "4" on page 158, when required, until all the queue managers and applications are migrated to the later version of the product.
6. Uninstall the earlier version of the product.

   When uninstalling the earlier product, you must stop all queue managers and applications that have loaded an IBM 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 MQ libraries on the server.
   b) Stop the queue managers and listeners on the server.
   c) Uninstall the earlier version of the product.

      Stop all local IBM MQ applications
7. Make `Inst_1` the primary installation.
   a) Run the **setmqinst** command

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

   - You do not have to set up a search path to run IBM MQ commands from the primary installation.
   - If you set an installation of the later version of the product 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`.

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 MQ installed.

Now that you have uninstalled the earlier version of the product, and made the later 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 libraries for the later version. If you have only one installation of the later version of the product installed, it is not necessary to run **setmqenv** to run commands.

Related concepts

“Queue manager coexistence” on page 95
Queue managers, with different names, can coexist on any server as long as they use the same IBM MQ installation. On z/OS, 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 98
You can install multiple copies of IBM MQ for UNIX, Linux, and Windows on the same server. These IBM MQ copies can be at the same or different version levels. This is called a multi-installation. Multi-installation is particularly useful when you upgrade from one IBM MQ version to a later version, because it allows you to run the earlier version alongside the later version.

Related tasks

Migrating on UNIX and Linux: single-stage
Single-stage migration is the term used to describe replacing the only installation of IBM MQ on a server, with a later release. Single stage migration is also known as *upgrading in place* or *in place upgrade*. Single-stage migration preserves existing scripts and procedures for running IBM 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.

Migrating on UNIX and Linux: side-by-side

“Planning to migrate IBM MQ to a later version on Windows” on page 112

Installing IBM MQ server on AIX

Installing IBM MQ server on Linux

Installing IBM MQ server on Solaris

Associating a queue manager with an installation

Changing the primary installation

Choosing an installation name

“Migrating IBM MQ library loading to a later version on UNIX and Linux” on page 164
On UNIX and Linux, no change in the way IBM MQ libraries are loaded is normally required if you upgrade from an earlier version of the product to a later version by replacing an earlier version of the product with the later version, based on the single stage scenario. However, if you choose to take advantage of multi-installation in the later version of the product, based on the side-by-side or multi-stage migration scenarios, you might have to configure the runtime environment differently, for the operating system to load the later version of the IBM MQ library.

“Migrating IBM MQ library loading to a later version on Windows” on page 134
On Windows, no change in the way IBM MQ libraries are loaded is normally required if you upgrade from an earlier version of the product to the later version by replacing an earlier version of the product with the later version, based on the single stage scenario. However, if you choose to take advantage of multi-installation in the later version of the product, based on the side-by-side or multi-stage migration scenarios, you might have to configure the runtime environment differently, for the operating system to load the later version of the IBM MQ library.

Related reference

setmqenv

setmqinst

setmqm

Linux UNIX *Reverting a queue manager to an earlier version on UNIX and Linux*

On UNIX and Linux, you can revert a queue manager to an earlier version of the product from a later version, 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 reverting 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 the later version. For more information see Backing up and restoring IBM 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 revert the queue manager to the earlier version of the product 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 MQ installations, you must identify the installation. Make sure that the commands you enter run against the correct installation; see **`setmqenv`**.

About this task

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

Important: If the queue manager is a member of one or more IBM MQ clusters, you should also review and follow the steps described in Recovering a cluster queue manager.

Procedure

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

   If you use the 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 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
   ```

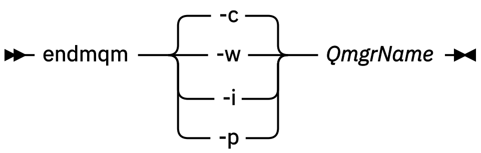
   `dspmq -o installation -o status` displays the installation name and status of queue managers associated with all installations of IBM MQ.

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

   b) Use the MQSC command **`DISPLAY LSSTATUS`** to list the status of listeners associated with a queue manager, as shown in the following example:

   ```
   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 MQ libraries they have loaded. If they do not, you must find another way to force applications to release IBM 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 MQ. The application is not informed about queue managers in the current installation being shut down.

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

    Note: “Applying maintenance level updates to multi-instance queue managers on AIX” on page 19, “Applying maintenance level updates to multi-instance queue managers on Linux” on page 35 and “Applying maintenance level updates to multi-instance queue managers on Solaris” on page 46 describe how to apply maintenance to a multi-instance queue manager. A multi-instance queue manager can continue to run on one server, while maintenance is applied to another server.

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

```
endmqlsr -m QMgrName
```

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

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

   a) Uninstall the earlier installation.

   b) Reinstall the product from a manufacturing refresh.

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

   d) Restore the queue manager data from the backup taken before installing the later version.

5. Restart the earlier version queue manager.

What to do next

You might be reverting to a earlier version on a server with multiple IBM MQ installations. If one of the installations is primary, after reverting the earlier version that installation, by default, becomes the primary installation.

You must review how applications connect to an installation. After reverting to the earlier version, some applications might connect to the wrong installation.

Related concepts

Backing up and restoring a queue manager

Related reference

BFGSS0023E errors and how to avoid them

Linux UNIX Migrating an IBM MQ MQI client on UNIX and Linux

Before migrating an IBM MQ MQI client, create a migration plan. Stop all IBM MQ activity on the client workstation. Upgrade the IBM MQ MQI client installation. Make any essential configuration and application changes.

Related concepts

“IBM MQ MQI client migration” on page 81

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

Related tasks

"Migrating an IBM MQ MQI client to the latest version on IBM i" on page 190
Before migrating an IBM MQ MQI client, create a migration plan. Stop all IBM MQ activity on the client workstation. Upgrade the IBM MQ MQI client installation. Make any essential configuration and application changes.

"Migrating an IBM MQ MQI client on Windows" on page 132
Before migrating an IBM MQ MQI client, create a migration plan. Stop all IBM MQ activity on the client workstation. Upgrade the IBM MQ MQI client installation. Make any essential configuration and application changes.

Linux UNIX *Migrating an IBM MQ MQI client to a later version on UNIX and Linux*

To upgrade a client to a later version of the product on UNIX and Linux, you must first stop all IBM MQ activity on the workstation, then uninstall the earlier version and install the later version. After you have upgraded the client, you can then make any essential configuration and application changes.

Before you begin

Before migrating an IBM MQ MQI client on UNIX and Linux, first create a migration plan. For guidance on what to include in the plan, see "Planning to migrate IBM MQ to a later version on UNIX and Linux" on page 146, as a guide.

About this task

IBM MQ MQI client migration is the process of converting IBM MQ MQI client configurations, and client and server channels from one version to another. Client migration is reversible. It is optional and manual on a client workstation and is required and automatic on the IBM MQ server.

You must upgrade an IBM 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.

Procedure

1. Review the IBM MQ system requirements for the later version of the product.

   See System Requirements for IBM MQ.
2. Review all the changes in IBM MQ that affect you.

   See "Changes that affect migration" on page 73.
3. End all IBM MQ activity on the workstation.

   You are now ready to upgrade the client. Follow the instructions for the appropriate platform that your enterprise uses.
4. AIX

   To upgrade the client on AIX:

   a) Uninstall your existing IBM MQ client installation.

      For more information, see Uninstalling or modifying IBM MQ on AIX.

   b) Follow the client installation procedure to install the upgraded version of the IBM MQ client:

      - For a client installation on a workstation, see Installing an IBM MQ client on AIX
      - For a client installation on an IBM MQ server, see Installing IBM MQ clients and servers on the same system.

5. Linux

   To upgrade the client on Linux:

   a) Uninstall your existing IBM MQ client installation.

      For more information, see Uninstalling or modifying IBM MQ on Linux.

   b) Follow the client installation procedure to install the upgraded version of the IBM MQ client:

      - For a client installation on a workstation, see Installing an IBM MQ client on Linux.
      - For a client installation on an IBM MQ server, see Installing IBM MQ clients and servers on the same system.

6. Solaris

   To upgrade the client on Solaris:

   a) Uninstall your existing IBM MQ client installation.

      For more information, see Uninstalling IBM MQ on Solaris.

   b) Follow the client installation procedure to install the upgraded version of the IBM MQ client:

      - For a client installation on a workstation, see Installing an IBM MQ client on Solaris.
      - For a client installation on an IBM MQ server, see Installing IBM MQ clients and servers on the same system.

What to do next

After upgrading the IBM MQ MQI client, you must check the client channel configuration, and verify that your IBM MQ MQI client applications work correctly with the later version of the product.

Related concepts

"IBM MQ MQI client migration" on page 81
IBM MQ MQI client migration is the process of converting IBM MQ MQI client configurations, and client and server channels from one version to another. Client migration can take place after upgrading the IBM MQ MQI client, and is reversible.

Related tasks

"Planning to migrate IBM MQ to a later version on UNIX and Linux" on page 146

Linux UNIX *Restoring an IBM MQ MQI client to an earlier version on UNIX and Linux*

To revert a client to an earlier version of the product on UNIX and Linux, you must uninstall the later version, and then install the earlier version.

About this task

If you revert an IBM MQ MQI client and client connection to an earlier code level, you must undo the configuration changes manually.

It is unusual to revert earlier IBM MQ MQI client libraries to a workstation.

Procedure

1. End all IBM MQ activity on the workstation.

   You are now ready to restore the client to the earlier version. Follow the instructions for the appropriate platform that your enterprise uses.

2. AIX

   To revert the client to the earlier version on AIX:

   a) Uninstall the IBM MQ MQI client code for the later version.

      For more information, see Uninstalling or modifying IBM MQ on AIX.

   b) Follow the client installation procedure to install the IBM MQ MQI client for the earlier version.

      For more information, see the client installation procedure for the earlier version that you want to install.

3. Linux

   To revert the client to the earlier version on Linux:

   a) Uninstall the IBM MQ MQI client code for the later version.

      For more information, see Uninstalling or modifying IBM MQ on Linux.

   b) Follow the client installation procedure to install the IBM MQ MQI client for the earlier version:

      For more information, see the client installation procedure for the earlier version that you want to install.

4. Solaris

   To revert the client to the earlier version on Solaris:

   a) Uninstall the IBM MQ MQI client code for the later version.

      For more information, see Uninstalling IBM MQ on Solaris.

   b) Follow the client installation procedure to install the IBM MQ MQI client for the earlier version.

      For more information, see the client installation procedure for the earlier version that you want to install.

5. If you configured a Client Connection Definition Table (CCDT) for a queue manager using the later version, revert to using a table created by a queue manager for the earlier version.

   If a 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. For more information, see MQI client: Client Channel Definition Table (CCDT).

Linux UNIX Migrating IBM MQ library loading to a later version on UNIX and Linux

On UNIX and Linux, no change in the way IBM MQ libraries are loaded is normally required if you upgrade from an earlier version of the product to a later version by replacing an earlier version of the product with the later version, based on the single stage scenario. However, if you choose to take advantage of multi-installation in the later version of the product, based on the side-by-side or multi-stage migration scenarios, you might have to configure the runtime environment differently, for the operating system to load the later version of the IBM MQ library.

Before you begin

To migrate applications from an earlier version of the product to the later version, you must know how the operating system loads an IBM 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 MQ library that the application loads. The library name does not change from an earlier version of the product to the later version, although the contents of the library do.

Read “Multi-installation queue manager coexistence on UNIX, Linux, and Windows” on page 98 before starting this task.

Plan and install the later version of IBM MQ, and remember the installation name and whether the installation was set to primary.

About this task

To migrate an application from an earlier version of the product to the later version, you do not have to recompile or relink the application, because the IBM MQ libraries are compatible with later versions; see “Application compatibility and interoperability with later versions of IBM MQ” on page 105.

The build procedure for IBM MQ applications is to include an explicit library path to the location of the IBM MQ libraries, and to `/usr/lib`, in the link step of the compiler, as shown in Figure 13 on page 165. The build procedure is the same for the later version of the product.

```
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 13. Linux C server application, 32 bit, threaded compile and link

The example shown in Figure 13 on page 165 is for Linux, but the build step for UNIX platforms is similar.

If you have followed this build procedure in the earlier release, then the effect of installing the later version of the product on the libraries that are loaded depends on which migration scenario that you are following:

Single-stage scenario

If you are replacing an earlier version of the product with the later version, based on the single stage scenario described in "Migrating on UNIX and Linux: single-stage" on page 150, you do not, in most cases, need to make any changes to the way IBM MQ libraries are loaded. The possible exception to this is if you changed the location of the libraries from the earlier version, or created symbolic links to the libraries.

Side-by-side and Multi-stage scenarios

If you have chosen a multi-installation approach to installing the later version of the product, based either on the side-by-side scenario described in "Migrating on UNIX and Linux: side-by-side" on page 153, or the multi-stage migration scenario described in "Migrating on UNIX and Linux: multi-stage" on page 156, you must investigate whether applications connecting to the later version of the product are linked to, and load libraries from, the correct installation and then modify the environment for the operating system to resolve IBM MQ dependencies for an application as appropriate. Typically, you can modify the runtime environment, rather than relink the application. You can use the following two commands to assist you in configuring the runtime environment:

- **`setmqinst`** sets the primary installation; see setmqinst.
- **`setmqenv`** initializes the command environment by setting environment variables; see setmqenv.

Table 11 on page 165 summarizes the actions needed for each of these scenarios. The examples in Table 11 on page 165 are all based on Linux, but the actions for UNIX are similar.

Table 11. UNIX and Linuxconfigurations

Action	Scenario	Latest version replaces earlier version in the same location Single-stage	Latest version replaces earlier version in a different location Side-by-side	Latest version alongside earlier version Multi-stage
`setmqinst`		**`setmqinst`** makes the later version installation primary. Symbolic links to the IBM MQ link libraries are inserted into `/usr/lib`.		No. The later version installation can be primary, because an earlier version is installed.

Table 11. UNIX and Linuxconfigurations (continued)

Action	Scenario	Latest version replaces earlier version in the same location Single-stage	Latest version replaces earlier version in a different location Side-by-side	Latest version alongside earlier version Multi-stage
No other configuration actions		Library loading works correctly. Library loading works, even without the later version 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 the earlier version correctly, nothing works with the later version.
`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 the earlier version correctly, nothing works with the later version.
`setmqenv`, with the `-k` or `-l` options set.		Library loading works correctly.		Library loading works correctly, both for the earlier version and the later version. The correct earlier version is loaded, because the later version library loads the earlier version library for queue managers that have not been migrated from the earlier version.
		The operating system finds the IBM MQ library location set by **`setmqenv`**. **`setmqenv`** adds the location to `LD_LIBRARY_PATH`. AIX This is `LIBPATH` on AIX. `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

1. Consider which of the following questions apply to your configuration.
   - Did you follow the build procedure documented in the product documentation for the earlier version of the product? You might be following a different build procedure tailored to your development environment, or adapted from a development tool.

- How did you specify the load path for the earlier version?
- Is the application is loaded by another environment, such as Eclipse, or an application server? You must modify the parameters that govern how the parent environment loads applications, not the way the parent environment is loaded.
- What constraints and requirements do you have on how the load path is specified in the later version? Security rules might restrict the use of `LD_LIBRARY_PATH`.
- Is the later version of the product installed alongside the earlier version?

2. Identify the installation of the later version of the product, from which the operating system is going to load IBM MQ libraries:

   - If you have a multiple installations of the later versions to load from a server, IBM MQ checks that the installation the library was loaded from is the installation that is associated with any queue manager the application calls. IBM MQ loads the correct library if the wrong library is loaded. It is necessary to configure only one runtime environment for all IBM MQ applications.
   - A typical choice is to set the primary installation. Setting an installation to be primary places symbolic links to the IBM MQ libraries in `/usr/lib`, and applications built have an explicit link to `/usr/lib`, and `/usr/lib` is also normally in the default library search path.
   - If you upgraded an earlier version installation to the later version, a link path to the earlier version installation now points to an installation containing the later version. Applications that have a fixed linkage path to the earlier version installation now load the libraries for the later installation. 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 an installation of the later version.
   - AIX 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 installations of the later version of the product, you must decide which installation to make primary, if you have chosen to make any primary. As long as applications load IBM MQ libraries from one of the later version installations, such as the primary installation, they can connect to queue managers associated with any other later version installation.

Related concepts

External library and control command links to primary installation on UNIX and Linux

Related tasks

Connecting applications in a multiple installation environment

Changing the primary installation

Loading IBM MQ libraries

“Migrating IBM MQ library loading to a later version on Windows” on page 134

On Windows, no change in the way IBM MQ libraries are loaded is normally required if you upgrade from an earlier version of the product to the later version by replacing an earlier version of the product with the later version, based on the single stage scenario. However, if you choose to take advantage of multi-installation in the later version of the product, based on the side-by-side or multi-stage migration scenarios, you might have to configure the runtime environment differently, for the operating system to load the later version of the IBM MQ library.

Related reference

“Coexistence” on page 94

Queue managers, with different names, can coexist on any server as long as they use the same IBM MQ installation. On z/OS, 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

Linux UNIX Migrating from a single instance to a multi-instance queue manager on UNIX and Linux

To migrate a single instance queue manager, to a multi-instance queue manager, on UNIX and Linux, 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.For a list of tested environments, see Testing statement for IBM MQ multi-instance queue manager file systems. Other environments might work; a test tool is provided with IBM 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 statement for IBM MQ multi-instance queue manager file systems. 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 IBM 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 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 in Example directory configurations on UNIX.

   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` on the shared drive. `/MQHA` is owned by the user and group `mqm` and has the access permissions `rwx`.

4. Copy the queue manager data and the logs to the share.

   Follow the procedure to back up the queue manager described in Backing up queue manager data.

   Note: Unlike on Windows, the hamvmqm utility cannot be used on UNIX.

5. Update the queue manager configuration information stored on the current queue manager server by completing the following steps:

   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 IBM MQ `mqs.ini` file, which is typically in the `/var/mqm` directory on UNIX and Linux:

   ```
   DataPath= share/data/QMgrName
   ```

   where *QMgrName* is the `Directory` name in the `QueueManager:` stanza in the `mqs.ini` file and *`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.

   ```
   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 the following connection name:

   ```
   echo DISPLAY CHANNEL(ENGLAND) CONNAME | runmqsc QM1
   ```

   ```
   5724-H72 (C) Copyright IBM Corp. 1994, 2024.  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 IBM 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 IBM MQ configuration file, mqs.ini

Related tasks

Backing up queue manager data

Changing configuration information on UNIX, Linux, and Windows systems

Create a multi-instance queue manager on Linux

Moving a queue manager to MSCS storage

Testing a shared file system for compatibility with IBM MQ Multi-instance Queue Managers

Verifying shared file system locking

Related reference

amqmfsck (file system check)

Testing statement for IBM MQ multi-instance queue manager file systems

Linux UNIX Reverting to a single-instance queue manager on UNIX and Linux

Revert a multi-instance queue manager to a single instance queue manager, on UNIX and Linux, 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:

   ```
   endmqm -w QMgrName
   ```

2. Stop the active queue manager instance.

   On the server running the active instance:

   ```
   endmqm -w (QMgrName)
   ```

3. Restart the queue manager, preventing standbys.

   On the server that is going to run the queue manager:

   ```
   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 IBM 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 UNIX and Linux" on page 168 to create a multi-instance queue manager.

Linux Cleaning up after using the `rpm` freshen or upgrade options on Linux

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 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 MQ package entries from the **`rpm`** database without removing the product from your system. You might also have partially installed IBM MQ

Procedure

Follow these steps to clean up your system.

1. Find out which IBM MQ MQ packages still have entries in your RPM database.

   ```
   rpm -qa | grep MQSeries
   ```

2. Remove all remaining IBM MQ packages from your system.

   ```
   rpm -e package-name
   ```

3. Remove the `/opt/mqm` directory.

   ```
   rm -rf /opt/mqm
   ```

Linux Rebuilding a C++ application on Linux

C++ IBM 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 MQ.

The GCC 4.1.2 libraries support SSL and TLS connections from an IBM 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 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 MQ application. For more detailed instructions about building Linux applications for IBM MQ ; see Building your procedural 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.

Linux Migrating MQ Telemetry on Linux

Follow these instructions to migrate your existing installation of MQ Telemetry on Linux to the latest version of the product.

Before you begin

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

About this task

The telemetry server is included in the product as an optional installation.

For IBM WebSphere MQ 7.5, the Client Software Development Kit (the telemetry clients) is also included in the optional installation. From IBM MQ 8.0 onwards, the Client Software Development Kit is no longer supplied as part of the product. Instead, the current version of the SDK is available in IBM Messaging Telemetry Clients SupportPac.

Because MQ Telemetry is a component of IBM MQ, MQ Telemetry can either be installed with the main product, or installed after the main product has been installed. When you upgrade from a previous version of the product, you must download and use the latest version of the Client Software Development Kit.

After the successful upgrade, Linux systems retain all telemetry data kept in `/var/mqm`. Telemetry data is migrated to the later version of the product when the queue manager is started again.

Procedure

1. Create a migration plan.

   See "Planning to migrate IBM MQ to a later version on UNIX and Linux" on page 146.
2. Migrate your queue managers to the latest release.
3. Install MQ Telemetry.
4. Verify that the MQ Telemetry installation was successful. See Verifying the installation of MQ Telemetry .

Results

Message AMQ4616 indicates completion of the task. The existing MQTT channels and previous subscriptions are still present.

Related tasks

Installing MQ Telemetry

Verifying the installation of MQ Telemetry

Verifying the installation of MQ Telemetry by using IBM MQ Explorer

IBM i Migrating IBM MQ on IBM i

IBM MQ migration tasks associated with IBM i are grouped in this section.

Procedure

- For information about creating a migration plan, see "Planning to migrate IBM MQ to a later version on IBM i" on page 174.
- For information about migrating an IBM MQ classes for JMS and IBM MQ classes for Java client, see "Migrating an IBM MQ classes for JMS and Java client on IBM i" on page 175.
- For information about migrating a queue manager from a previous release, see "Migrating a queue manager to the latest version on IBM i" on page 176 and "Migrating a queue manager to a later version on IBM i - alternative method" on page 187.
- For information about upgrading an IBM MQ system, see "Upgrading an entire IBM MQ system on IBM i" on page 190.
- For information about upgrading an IBM MQ MQI client installation, see "Migrating an IBM MQ MQI client to the latest version on IBM i" on page 190.
- For information about converting a single instance queue manager to a multi-instance queue manager, see "Migrating from a single instance to a multi-instance queue manager on IBM i" on page 191.
- For information about reverting a multi-instance queue manager to a single instance queue manager, see "Reverting to a single-instance queue manager on IBM i" on page 194.

Related concepts

"Migration concepts and methods" on page 76
An overview of the various concepts and methods for migrating from one release of the product to another.

Related tasks

"Migrating IBM MQ on UNIX and Linux" on page 145
Migration tasks associated with UNIX and Linux platforms are grouped in this section.

"Migrating IBM MQ on Windows" on page 111
IBM MQ migration tasks associated with Windows platforms are grouped in this section.

"Migrating IBM MQ on z/OS" on page 195
Migration tasks associated with z/OS are grouped in this section.

Related reference

"Changes that affect migration" on page 73

IBM i Planning to migrate IBM MQ to a later version on IBM i

Before migrating IBM MQ to a later version on IBM i, review the system requirements information, and the information about any changes that might affect migration, then create a migration plan.

Before you begin

If there are concepts about migration you do not understand, see "Migration concepts and methods" on page 76.

If you are migrating to IBM MQ 9.1 from IBM WebSphere MQ 7.1 or earlier, you must first migrate to an interim version. See Migration paths.

Before you use side-by-side or multi-stage migration to migrate from IBM WebSphere MQ 7.0.1, check that your IBM WebSphere MQ 7.0.1 installation is at Fix Pack 6 or later.

About this task

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

Procedure

1. Review the IBM MQ system requirements for the later version of the product.

   See System Requirements for IBM MQ.
2. Review all the changes in IBM MQ that affect you.

   See "Changes that affect migration" on page 73.
3. Review performance changes.

   See MQ Performance documents.
4. Review the readme file for the later version of IBM MQ.

   See IBM MQ, WebSphere MQ, and MQSeries product readmes.
5. Plan the sequence and timing of queue manager migrations.

   - 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 "Migrating a queue manager in a high-availability configuration" on page 225.
6. Plan to migrate your queue manager to the later version.

   See IBM i - Migrating a queue manager to the later release or Migrating a queue manager to the later release, alternative method

   Backing up queue manager data is part of the queue manager migration task. An alternative approach is to install and configure a new server, then test the later version with a new queue manager on the new server. When you are ready to go into production on the later version, copy the queue manager configuration and data to the new server.
7. Plan to update any manual or automated procedures you have written with changes to messages and codes.

   From IBM MQ 9.1, a suffix letter, indicating the severity of a message (I, W, E, S or T) is appended to IBM MQ diagnostic (AMQ) messages. Existing scripts looking for error codes without the severity will fail. For example, existing scripts looking for error matching to AMQ7468 will fail. You must update the scripts to look for error codes with the severity suffix added (for example, AMQ7468I). For more information, see IBM MQ messages on Multiplatforms.
8. Decide on what regression tests to perform before putting the queue manager into production on the later version. Include the procedures and applications you identified in previous steps in your regression tests.
9. Plan to migrate your IBM MQ MQI client installations to the later version.
10. Plan to migrate your client and server applications to use new functions in the later version.

IBM i Migrating an IBM MQ classes for JMS and Java client on IBM i

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

Before you begin

SupportPac MQ88 is installed.
: If you try to install the latest version of IBM MQ classes for Java anyway, the installation fails with a warning requesting you to uninstall the old client. You must follow the steps in this task to uninstall IBM MQ classes for Java and IBM MQ classes for JMS.

A previous version of IBM MQ classes for Java is installed.
: Installation of the latest version of IBM MQ classes for Java uninstalls the previous version automatically. Do not follow the steps in this task.

About this task

The steps in this task uninstall the IBM MQ classes for JMS and Java.

Procedure

To uninstall the previous IBM MQ Java client:

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

   ```
   DLTLICPGM LICPGM(5648C60) OPTION(*ALL)
   ```

2. If the previous step failed to delete the IFS directory `/QIBM/ProdData/mqm/java` and its subdirectories, use the **EDTF** command, for example:

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

   and select option 9 against the `java` directory.

IBM i Migrating a queue manager to the latest version on IBM i

Follow these instructions to migrate a queue manager on IBM i to the latest MQ version.

Before you begin

1. Create a migration plan. Use the planning task, Planning migration to the latest version, as a guide.
2. Review the IBM MQ system requirements for the latest version of the product; see System Requirements for IBM MQ
3. Review any other installed SupportPacs for their applicability to the latest version of the product.

About this task

There are two types of migration:

- The migration takes place on the same machine, optionally accompanied by a hardware upgrade. This migration is referred to as a *slip installation*. On IBM i, uninstalling the earlier version before you install the later version is optional.
- The migration takes place on a different machine. This migration is referred to as a *side-by-side installation*.

  A side-by-side installation lets you prepare the new environment first, without interrupting the queue manager. It also gives you the limited option of reverting to use the earlier version installation, if the migration is unsuccessful. It is limited, because you cannot restore the queue manager data from the later version. You must restart processing with the queue manager data at the point you stopped the queue manager on the earlier release.

  If you decide to do a side-by-side installation, you must prepare the new server first, installing the prerequisite software.

If you want to add Advanced Message Security to your system, you must select Option (2) when you install the product; see Installing Advanced Message Security on IBM i for further information.

Related tasks

"Migrating a queue manager on UNIX and Linux" on page 147
The procedures for migrating a queue manager to a later version of the product, and for restoring a queue manager to an earlier version of the product are detailed in this section.

"Migrating a queue manager on Windows" on page 116

The procedures for migrating a queue manager to a later version of the product, and for restoring a queue manager to an earlier version of the product are detailed in this section.

“Migrating IBM MQ on z/OS” on page 195
Migration tasks associated with z/OS are grouped in this section.

IBM i *Installation methods on IBM i*

Select a slip installation or a side-by-side installation to upgrade IBM MQ for IBM i.

About this task

A slip installation upgrades IBM MQ for IBM i on a computer with an earlier version is installed.

A side-by-side installation upgrades IBM MQ for IBM i on a different computer. You must save your queue managers before you start.

Follow the steps in the following tasks to carry out an upgrade.

The steps for both forms of upgrade are identical, except that you do not carry out the actions described in “Restore queue managers after upgrading IBM MQ on IBM i” on page 185 for a slip installation.

IBM i *End IBM MQ activity on IBM i*

End IBM MQ applications and connections, and remove any unwanted or indoubt messages.

About this task

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

Procedure

1. Sign on to the system with a user profile that has *ALLOBJ special authority, for example QSECOFR.
2. Stop all applications that are using the existing version of IBM MQ.

   To identify applications using the queue manager, use the command `WRKMQM`, option 22, Work with queue manager jobs, to help find them. Ignore jobs starting with AMQ* or RUN* and focus on your application job names.
3. End all channels for all queue managers on the system. To do so, use the WRKMQMCHL command and select option 15.
4. On each queue manager, end the command server. To do so, enter the command:

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

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

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

IBM i *Quiesce IBM MQ on IBM i*

Stop all queue managers. If necessary force all queue managers to stop, tidy up shared memory and end all jobs in the QMQM subsystem.

About this task

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

Procedure

Prepare to quiesce queue managers:

1. Sign on to a new interactive IBM i session, ensuring that you are not accessing any IBM MQ objects.
2. Ensure that you have the following authorities:
   - *ALLOBJ authority, or object management authority for the QMQM library.
   - Sufficient authority to use the ENDSBS command.
3. Warn all users that you are going to stop IBM MQ.
4. Stop the mqweb server by entering the following command:

   ```
   ENDMQWEB
   ```

Quiesce all queue managers:

5. Run the ENDMQM command:

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

   Where *15* is a timeout value in seconds.

   If the ENDMQM command has not completed within a reasonable period (at least 10 minutes), run the WRKMQM command. This command identifies the queue managers that are still ending. Then force each one in turn to stop by running the following command:

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

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

   Complete the tidying up of shared memory by running the following command:

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

If the commands in the previous step do not complete, end the subsystem immediately:

6. Run the following command:

   ```
   ENDSBS SBS(QMQM) OPTION(*IMMED)
   ```

If the command in the previous step also does not complete, use the operating system command ENDJOB to end all jobs in the subsystem QMQM:

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

7. If a QMGR must be shut down manually, end the jobs (ENDJOB) in the following order. Wait a few minutes for AMQA* or AMQZ* jobs to tidy up.
   a. RUNMQLSR - TCP listener (multi-threaded)

b. AMQCLMAA - TCP listener (single-threaded)
c. AMQRMPPA - Channel process pooling job
d. RUNMQCHI - channel initiator
e. AMQCRSTA - receiving MCA jobs
f. RUNMQCHL - sending MCA jobs
g. AMQCRS6B - LU62 receiver channel
h. AMQPCSEA - command server
i. RUNMQTRM - Application trigger monitor
j. RUNMQDLQ - Dead letter queue handler
k. AMQFCXBA - IBM Integration Bus Worker Job
l. AMQFQPUB - Queued Publish/Subscribe Daemon
m. RUNMQBRK - IBM Integration Bus Control Job
n. AMQZMUC0 ('0' is a zero) - Utility Manager
o. AMQZMUF0 ('0' is a zero) - Utility Manager
p. AMQZMUR0 ('0' is a zero) - Utility Manager
q. AMQZMGR0 ('0' is a zero) - Process Controller
r. AMQRRMFA - cluster repository manager
s. AMQZDMAA - deferred message manager
t. AMQZFUMA - object authority manager
u. AMQZLSA0 ('0' is a zero) - LQM agents
v. AMQZLAA0 ('0' is a zero) - LQM agents
w. AMQZXMA0 ('0' is a zero) - Execution Controller

8. Run the following command:

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

9. Run the following command:

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

Where *05* is a timeout value in seconds.

10. Manually clean up shared memory.

Run the following command:

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

then:

a. Take option 5 for **&SYSTEM** and check that the following directories are empty: isem, esem, msem, ssem, and shmem.
b. Take option 5 for **QMGRNAME** and check that the following directories are empty:- isem, esem, msem, ssem, and shmem.
c. Take option 5 for **&ipcc** in the QMGRNAME directory and check that the following directories are empty:- isem, esem, msem, ssem, and shmem.
d. Take option 5 for **&qmpersist** in the QMGRNAME directory and check that the following directories are empty:- isem, esem, msem, ssem, and shmem.

e. Take option 5 for **&app** and check that the following directories are empty: `isem, esem, msem, ssem,` and `shmem`.

IBM i *Save IBM MQ data on IBM i*

Save IBM MQ data after removing unwanted FDC, trace, and JOB files.

Before you begin

You need to have completed the tasks to remove unwanted and indoubt messages and quiesced IBM MQ.

About this task

Procedure

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

   ```
   CRTSAVF FILE(QGPL/ queue_manager_library )
   ```

   where the *queue_manager_library* name consists of the name of the queue manager preceded by QM.
2. Save your queue manager libraries into the save files. To do so, issue the commands:

   ```
   SAVLIB LIB( queue_manager_library ) DEV(*SAVF)
   SAVF(QGPL/ queue_manager_library )
   ```

3. Remove all unwanted FDC data from directory:

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

4. Remove old FDC files with the command:

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

   This command cleans up all files with an extension of 'FDC' in the IFS.
5. Remove old JOB files with the command:

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

   This command cleans up all files with an extension of 'JOB' in the IFS.
6. Remove all unwanted trace data from directory, or remove the whole directory:

   ```
   QIBM/UserData/mqm/trace
   ```

7. Remove all trace files with the command:

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

8. Create a save file for IBM MQ IFS data. To do so, issue the command:

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

9. Save your IBM MQ IFS data, using the command:

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

10. If you are going to run IBM MQ on a new machine, transfer the save files to the new machine.

IBM i *Install IBM MQ server on IBM i*

Install the IBM MQ server in its primary language.

Before you begin

You have completed planning the installation, obtained the installation disks, and set the system values; see Setting system values.

About this task

Install the IBM MQ server and force object conversion. Object conversion migrates objects from the older to the newer version. By performing it now, rather than when an object is first used, you avoid slowing down the first use of the upgraded product.

After following the optional step to pre-agree the license, the **RSTLICPGM** command runs without requiring any interactive input. Otherwise the license agreement is displayed for you to accept. See License requirements.

Procedure

1. Sign on to the system with a user profile that has `*ALLOBJ` special authority, for example `QSECOFR`.
2. Optionally pre-agree the license terms and conditions by running the command,

   ```
   CALL PGM (QSYS/QLPACAGR) PARM ('5724H72' 'V8R0M0' '0000' 0)
   ```

   Where the parameters of **PARM** are,

   5724H72

   The product identifier for IBM i.

   V9R1M0

   The version, release, and modification level.

   0000

   The option number for the *BASE IBM MQ product option.

   0

   Unused error structure.
3. Install IBM MQ for IBM i, base product, and primary language.

   ```
   RSTLICPGM LICPGM (5724H72) DEV (installation device) OPTION (*BASE) OUTPUT (*PRINT)
   ```

   where the parameters of RSTLICPGM are,

   LICPGM (5724H72)

   The product identifier for IBM i.

   DEV (*installation device*)

   The device from which the product is to be loaded, typically an optical drive, for example, `OPT01`.

   OPTION (*BASE)

   Install the base IBM MQ for IBM i product.

   Unspecified parameters

   Unspecified parameters such as **RSTOBJ** `(*ALL)`, revert to defaults. The command installs both IBM MQ and the language files for the primary language of your system. For installing additional languages see Installing translated versions.

What to do next

Install any Progam Temporary Fixes (PTF) that have been issued.

IBM i *Install samples on IBM i*

Install the IBM MQ samples

Before you begin

If you have not already done so, sign on to the system with a user profile that has `*ALLOBJ` special authority, for example `QSECOFR`.

About this task

Install the samples.

After following the optional step to pre-agree the license, the **RSTLICPGM** command runs without requiring any interactive input. Otherwise the license agreement is displayed for you to accept. See License requirements.

Procedure

1. Optionally pre-agree the license terms and conditions by running the command,

```
 CALL PGM (QSYS/QLPACAGR) PARM ('5724H72' 'V8R0M0' '0001' 0)
```

   Where the parameters of **PARM** are,

   5724H72
   The product identifier for IBM i.

   V9R1M0
   The version, release, and modification level.

   0001
   The option number for the samples.

   0
   Unused error structure.

2. Install the samples using the command:

```
RSTLICPGM LICPGM (5724H72) DEV (installation device) OPTION (1) OUTPUT (*PRINT)
```

   Where the parameters of RSTLICPGM are,

   LICPGM (5724H72)
   The product identifier for IBM i.

   DEV (*installation device*)
   The device from which the product is to be loaded, typically an optical drive, for example, `OPT01`.

   OPTION (1)
   Install the samples for IBM i.

   OUTPUT (*PRINT
   The output is printed with the spooled output of the job.

IBM i *Install translated versions on IBM i*

Install translated versions of IBM MQ from a choice of national-languages.

About this task

The following language versions are available for IBM i:

Table 12. National-language versions of IBM MQ for IBM i

Language ID	Language
2909	Belgian English
2966	Belgian French MNCS (Multi-National Character Set)
2981	Canadian French MNCS
2975	Czech
2950	English uppercase
2924	English uppercase and lowercase
2984	English US DBCS
2938	English US uppercase DBCS
2928	French
2940	French MNCS
2929	German
2939	German MNCS
2976	Hungarian
2932	Italian
2942	Italian MNCS
2962	Japanese
2986	Korean
2978	Polish
2979	Russian
2989	Simplified Chinese
2931	Spanish

IBM MQ for IBM i is installed in the language that is the primary language on your system.

You can install additional versions of the product in any of the languages shown in Table 12 on page 183. To do so complete the following steps:

Procedure

1. Sign on to the system with a user profile that has *ALLOBJ special authority
2. Issue the following command specifying the appropriate language ID:

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

   This installs the commands, message file, and panel groups into the relevant QSYS library for the language. For example, library QSYS2928 is used for French. If this QSYS29nn library does not exist, it is created by the **RSTLICPGM** command.

Results

Note:

1. To run the Japanese language version of IBM MQ for IBM i, the CCSID of the job must be 939 (5035) rather than 930 (5026) because IBM MQ uses lowercase English characters.
2. If you are installing IBM MQ for IBM i onto a machine for which the primary language is not on the CD, the installation program prompts you to load a CD containing the product in that language. If, however, you have only one product CD, this means that the IBM MQ product has not been translated into your language. To get around this issue, proceed as follows:
    - Install the product in one of the supplied languages, and then add the corresponding QSYS29nn library into the *system library* list (for example using command **CHGSYSLIBL**). At the same time, check that there are no IBM MQ *CMD, *MENU, or *MSGF objects in libraries higher up the library list. If some exist, then either delete these objects (because they refer to an earlier version of IBM MQ) or reorder the System Library list (because the product has been installed in more than one of the supplied languages).

IBM i *Verify the installation on IBM i*

How to check that your installation has been successful.

Procedure

1. To ensure that the product has loaded correctly, issue the Display Software Resources (DSPSFWRSC) command and check that the licensed program 5724H72 is listed. If you have installed the base and the optional samples, you see:

```
Resource
ID    Option Feature Description
5724H72  *BASE  5050 IBM MQ for IBM i
5724H72  *BASE  2924 IBM MQ for IBM i
5724H72  1    5050 IBM MQ for IBM i - Samples
```

2. Press F11, while viewing the Display Software Resources screen, and you see the library and version number of the products installed:

```
Resource          Feature
ID    Option Feature  Type  Library  Release
5724H72  *BASE  5050   *CODE  QMQM   V9R1M0
5724H72  *BASE  2924   *LNG   QMQM   V9R1M0
5724H72  1    5050   *CODE  QMQMSAMP V9R1M0
```

3. If you have installed additional language versions, you also see entries for these versions. For example, if you have installed the French version, for which the language ID is 2928, you see:

    a)
    ```
    Resource
    ID    Option Feature Description
    5724H72  *BASE  2928 IBM MQ for IBM i
    ```

    b) and when you press F11:

    ```
    Resource           Feature
    ID    Option Feature  Type  Library   Release
    5724H72  *BASE  2928  *LNG  QSYS2928  V9R1M0
    ```

4. Use the command DSPMQMVER to check exactly what version you have installed. For example, for V9R1M0, it reports:

```
Version: 9.1.0.0
```

IBM i *Verify the upgrade on IBM i*

After you have verified the installation, start the IBM MQ subsystem, check the queue managers, and take a fresh media recovery checkpoint.

About this task

To verify that you have migrated to the latest version IBM MQ for IBM i, successfully:

Procedure

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

   ```
   CHGUSRPRF USRPRF( YOUR PROFILE ) GRPPRF(QMQMADM)
   CHGUSRPRF USRPRF( YOUR PROFILE ) SUPGRPPRF(QMQMADM)
   ```

2. Start the IBM MQ subsystem with the command:

   ```
   STRSBS SBSD(QMQM/QMQM)
   ```

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

   ```
   WRKMQM
   ```

   Use option 14 against each queue manager to start it.

   Use option 5 against each queue manager to check its attributes.
4. You can use the other options to check your queue manager objects. For example, check your queues using option 18, check your channels using option 20, and so on.
5. Take a fresh media recovery checkpoint, using the following command:

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

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

IBM i *Restore queue managers after upgrading IBM MQ on IBM i*

Complete the side-by-side upgrade by restored the saved queue managers onto the server that you have upgraded.

Before you begin

Note: Carry out this task only if you are performing a side-by-side upgrade.

Ensure that you have saved your queue manager data, see "End IBM MQ activity on IBM i" on page 177, and installed and verified the upgrade.

About this task

Transfer the queue manager data, and journal receivers, onto the server that has been upgraded.

Procedure

1. Restore the queue manager libraries for every queue manager, using the command:

```
RSTLIB SAVLIB( queue_manager_library ) DEV(*SAVF) (*PRINT)
SAVF(QGPL/ queue_manager_library )
```

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

2. Restore the IBM MQ IFS data, using the command:

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

3. To associate the journal receivers, issue the command WRKJRN on the journal AMQAJRN in each queue manager library, by pressing *PF4* and selecting option 9.
4. If you want to set up your work management environment, job descriptions, and pools, see the Administering IBMi for guidance. Otherwise, use the default setup.

IBM i *After Upgrading on IBM MQ for IBM i*

Tasks to perform after you have upgraded IBM MQ for IBM i.

About this task

Satisfy yourself the upgrade has completed successfully.

Procedure

Delete the saved data in the save files in QGPL. This data was saved in "Save IBM MQ data on IBM i" on page 180.

IBM i *Post installation tasks for IBM i*

Tasks to perform after you have installed IBM MQ for IBM i, and before using it.

About this task

When you have correctly installed IBM MQ for IBM i on your system:

Procedure

1. For the latest product information for IBM i, see System requirements for IBM MQ .
2. To install and apply all fix packs, see "Applying maintenance level updates on IBM i" on page 21.
3. Where you have more than one system and a mixture of releases of OS/400 or IBM i, and IBM MQ, you must take care when compiling CL programs. You must compile CL programs either on the system they are to run on, or on one with an identical combination of releases of OS/400 or IBM i, and IBM MQ. When you install later versions of IBM MQ, delete all IBM MQ commands from previous releases in any QSYSVvRrMm libraries using the QSYS/DLTCMD command.
4. If you have not installed IBM MQ on your system before, you must add user profiles to the QMQMADM group profile. Make all user profiles that are to be used for creating and administering queue managers members of the QMQMADM group profile, using the command CHGUSRPRF.
   a) Start the IBM MQ subsystem, by issuing the command:

      ```
      STRSBS SBSD(QMQM/QMQM)
      ```

      Note: The subsystem must be started after each IPL of the system, so you might choose to start it as part of your system startup process.
5. Create the system-default objects. The system-default objects are created automatically when you issue the CRTMQM command to create a queue manager. For example: CRTMQM

MQMNAME(QMGRNAME) ASP(*SYSTEM). You can refresh them using the STRMQM command (Warning: this command will replace any existing default objects). For example: STRMQM MQMNAME(QMGRNAME) RDEFSYS(*YES). Refer to the onscreen help for information about using this command.

Note: on the command STRMQM MQMNAME(QMGRNAME) RDEFSYS(*YES):

- The command does not re-create the objects, it performs a CRTxxxx REPLACE(*YES) for all of the SYSTEM.* objects.
- This means that it refreshes the parameters on the objects back to their defaults. So if, for example, on the SYSTEM.DEFAULT.LOCAL.QUEUE object, TRGENBL had previously been changed to *YES, then, when the command is run, it is changed back to TRGENBL(*NO).
- If any messages exist on a queue, they are left intact, because the queues are not physically deleted.
- The contents of the SYSTEM.AUTH.DATA.QUEUE are untouched when this command is run.
- So, if the contents of this (or any other significant queue) become corrupt, it must be physically deleted and re-created either from scratch, or from a backup.

Results

You are now ready to start using IBM MQ for IBM i.

Note: When you install IBM MQ for IBM i, two user profiles are created:

- QMQM
- QMQMADM

These two objects are central to the correct running of IBM MQ for IBM i. Do not alter or delete them. If you do, IBM cannot guarantee correct behavior of your product.

If you uninstall IBM MQ and data, these profiles are deleted. If you uninstall IBM MQ only, these profiles are retained.

IBM i Migrating a queue manager to a later version on IBM i - alternative method

An alternative method of migrating a queue manager from an earlier version to a later version

Before you begin

1. Review the IBM MQ system requirements for the later version of the product; see System Requirements for IBM MQ
2. Review any other installed SupportPacs for their applicability to the later version of IBM MQ.

About this task

There are various parts to this form of migration:

1. As part of upgrading the IBM MQ product, carry out the following tasks:
   a. “Preparing to install IBM MQ on IBM i” on page 188
   b. “Install IBM MQ server on IBM i” on page 188
2. Following the IBM MQ product upgrade, carry out the following task:
   a. “Post installation tasks” on page 189

IBM i Preparing to install IBM MQ on IBM i

Carry out the following tasks to prepare your system for an upgrade.

Procedure

1. Stop the IBM MQ queue managers by issuing the following command:

   ```
   ENDMQM MQMNAME(*ALL) OPTION(*IMMED) ENDCCTJOB(*YES) RCDMQMIMG(*YES)
   TIMEOUT(30)
   ```

   Ensure that the user profile issuing this command has `*ALLOBJ` authority.
2. Create a save file for every queue manager library on your system. To do so, issue the command:

   ```
   CRTSAVF FILE(QGPL/ queue_manager_library )
   ```

   where the *queue_manager_library* name consists of the name of the queue manager preceded by `QM`.
3. Save your queue manager libraries into the save files. To do so, issue the commands:

   ```
   SAVLIB LIB( queue_manager_library ) DEV(*SAVF)
   SAVF(QGPL/ queue_manager_library )
   ```

4. Create a save file for IBM MQ IFS data. To do so, issue the command:

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

5. Save your IBM MQ IFS data, using the command:

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

6. If you are going to run IBM MQ on a new machine, transfer the save files to the new machine.
7. Issue the following command before you upgrade your IBM MQ product, only if the upgrade is required on the same machine.

   a) `DLTMQM` *`QMgrName`*

   b) `ENDSBS SBS(QMQM) OPTION(*IMMED)`

   c) `WRKOBJLCK OBJ(QMQM) OBJTYPE(*LIB)`

   Relinquish any locks on the system.

IBM i Install IBM MQ server on IBM i

Install the IBM MQ server in its primary language and force object conversion.

Before you begin

In either of the following cases, ensure that you have completed the planning and set the system values; see Setting system values

- If you have obtained the product through the Passport Advantage and Passport Advantage Express® web site, follow the instructions in the `EGA.README.txt` file.
- If you have obtained the product on disk, follow the instructions within this topic.

About this task

Install the IBM MQ server and force object conversion. Object conversion migrates objects from the older to the newer version. By performing it now, rather than when an object is first used, you avoid slowing down the first use of the upgraded product.

After following the optional step to pre-agree the license, the **RSTLICPGM** command runs without requiring any interactive input. Otherwise the license agreement is displayed for you to accept. See License requirements.

Procedure

1. Sign on to the system with a user profile that has *ALLOBJ special authority, for example QSECOFR.
2. Optionally pre-agree the license terms and conditions by running the command,

   ```
   CALL PGM (QSYS/QLPACAGR) PARM ('5724H72' 'V8R0M0' '0000' 0)
   ```

   Where the parameters of **PARM** are,

   5724H72
   The product identifier for IBM i.

   V9R1M0
   The version, release, and modification level.

   0000
   The option number for the *BASE IBM MQ product option.

   0
   Unused error structure.
3. Install IBM MQ for IBM i, base product, and primary language.

   ```
   RSTLICPGM LICPGM (5724H72) DEV (installation device) OPTION (*BASE) OUTPUT (*PRINT)
   ```

   where the parameters of RSTLICPGM are,

   LICPGM (5724H72)
   The product identifier for IBM i.

   DEV (*installation device*)
   The device from which the product is to be loaded, typically an optical drive, for example, OPT01.

   OPTION (*BASE)
   Install the base IBM MQ for IBM i product.

   Unspecified parameters
   Unspecified parameters such as **RSTOBJ** (*ALL), revert to defaults. The command installs both IBM MQ and the language files for the primary language of your system. For installing additional languages see Installing translated versions.

What to do next

Install any Progam Temporary Fixes (PTF) that have been issued.

To install the IBM MQ samples, see: "Install samples on IBM i" on page 182.

Post installation tasks

Actions required after upgrading IBM MQ.

About this task

Install the samples.

Carry out these steps after installing the product.

Procedure

1. Issue the following commands:
   a) STRSBS SBSD(QMQM/QMQM)

   b) `CRTMQM MQMNAME(QMgrName) DFTQMGR(*YES)`

      You receive the message " IBM MQ queue manager created."

   c) `STRMQM MQMNAME(QMgrName)`

      You receive the message " IBM MQ queue manager `'QMgrName'` started."

2. Issue the following command:

   ```
   STRMQMMQSC SRCMBR(QMgrName) SRCFILE(*CURLIB/QMQSC) OPTION(*RUN)
   MQMNAME(QMgrName)
   ```

3. Reapply IBM MQ Authorities by issuing the command: `CALL PGM(*CURLIB/QMgrName)`

   a) You must compile the CLP as follows:

      ```
      CRTCLPGM PGM(*CURLIB/QMgrName) SRCFILE(*CURLIB/QMAUT) SRCMBR(*PGM)
      ```

IBM i Upgrading an entire IBM MQ system on IBM i

How to upgrade an IBM MQ system on IBM i

Before you begin

Ensure that you have backed up your entire system.

About this task

To upgrade an IBM MQ system on IBM i you carry out a slip installation.

See "Installation methods on IBM i" on page 177 for further information.

Related tasks

"Migrating a queue manager on Windows" on page 116
The procedures for migrating a queue manager to a later version of the product, and for restoring a queue manager to an earlier version of the product are detailed in this section.

IBM i Migrating an IBM MQ MQI client to the latest version on IBM i

Before migrating an IBM MQ MQI client, create a migration plan. Stop all IBM MQ activity on the client workstation. Upgrade the IBM MQ MQI client installation. Make any essential configuration and application changes.

Before you begin

1. Create a migration plan. Use the planning task, "Planning to migrate IBM MQ to a later version on IBM i" on page 174, as a guide.

Procedure

1. Review the IBM MQ system requirements for the later version of the product.

   See System Requirements for IBM MQ.

2. Review all the changes in IBM MQ that affect you.

   See "Changes that affect migration" on page 73.

3. End all IBM MQ activity on the workstation.
4. Upgrade the client.

   To upgrade an IBM MQ MQI client for IBM i installation on a workstation; see Client installation procedure on IBM i .

What to do next

Complete the tasks in your migration plan, such as verifying IBM MQ MQI client applications work correctly with the latest version.

Related concepts

Client installation procedure on IBM i

"IBM MQ MQI client migration" on page 81
IBM MQ MQI client migration is the process of converting IBM MQ MQI client configurations, and client and server channels from one version to another. Client migration can take place after upgrading the IBM MQ MQI client, and is reversible.

Related tasks

"Migrating an IBM MQ MQI client on UNIX and Linux" on page 161
Before migrating an IBM MQ MQI client, create a migration plan. Stop all IBM MQ activity on the client workstation. Upgrade the IBM MQ MQI client installation. Make any essential configuration and application changes.

"Migrating an IBM MQ MQI client on Windows" on page 132
Before migrating an IBM MQ MQI client, create a migration plan. Stop all IBM MQ activity on the client workstation. Upgrade the IBM MQ MQI client installation. Make any essential configuration and application changes.

Installing IBM MQ MQI clients on the same machine as the server

IBM i Migrating from a single instance to a multi-instance queue manager on IBM i

To migrate a single instance queue manager to a multi-instance queue manager, on IBM i, 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. IBM i has been tested with multi-instance queue managers and is known to work. For a list of tested environments, see Testing statement for IBM MQ multi-instance queue manager file systems. The support statement has detailed version and prerequisite information for each environment it lists. Other environments might work; a test tool is provided with IBM 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 statement for IBM MQ multi-instance queue manager file systems. 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 IBM 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 IBM i, QMQM, QMQMADM, and any other user profiles that are granted access to the share must have the same passwords on all the servers

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.

   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 IBM i, follow the instructions to create a network share using NetServer.

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. elect one of these methods:

   - Follow the instructions in Backups of IBM MQ for IBM i 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 192.

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 IBM i,

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 IBM MQ `mqs.ini` file, which is typically in the `/QIBM/UserData/mqm` directory on IBM i:

```
DataPath= share/data/QMgrName
```

Where, *QMgrName* is the `Directory` name in the `QueueManager:` stanza in the `mqs.ini` file on IBM i. *`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 on the server that ran the queue manager in the previous release.

```
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, 2024.  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 IBM 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

IBM i Multi-instance queue managers on IBM i

Shared file system

IBM i The IBM MQ configuration file mqs.ini - IBM i

Related tasks

Backing up queue manager data

Testing a shared file system for compatibility with IBM MQ Multi-instance Queue Managers

Verifying shared file system locking

Related reference

`amqmfsck` (file system check)

Testing statement for IBM MQ multi-instance queue manager file systems

IBM i Reverting to a single-instance queue manager on IBM i

Revert a multi-instance queue manager to a single instance queue manager, on IBM i, 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:

```
ENDMQM MQMNAME (QMgrName) *WAIT
```

2. Stop the active queue manager instance.

   On the server running the active instance:

```
ENDMQM MQMNAME (QMgrName) *WAIT
```

3. Restart the queue manager, preventing standbys.

   On the server that is going to run the queue manager:

```
STRMQM MQMNAME (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 IBM 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 IBM i" on page 191 to create a multi-instance queue manager.

z/OS Migrating IBM MQ on z/OS

Migration tasks associated with z/OS are grouped in this section.

Before you begin

If you are migrating from IBM WebSphere MQ 7.1.0 to IBM MQ 9.1, you should migrate to IBM MQ 8.0 first.

Related concepts

"Migration concepts and methods" on page 76
An overview of the various concepts and methods for migrating from one release of the product to another.

"Migration paths" on page 75
An overview of the migration paths between different IBM MQ versions. For some IBM MQ for z/OS migration paths, you can revert to the version you were using prior to migration. For IBM MQ for Multiplatforms, you cannot easily revert to a previous version.

Related tasks

"Migrating IBM MQ on IBM i" on page 174
IBM MQ migration tasks associated with IBM i are grouped in this section.

"Migrating IBM MQ on UNIX and Linux" on page 145
Migration tasks associated with UNIX and Linux platforms are grouped in this section.

"Migrating IBM MQ on Windows" on page 111
IBM MQ migration tasks associated with Windows platforms are grouped in this section.

Related reference

"Changes that affect migration" on page 73

Related information

WebSphere MQ Support, Migration PTFs

z/OS Migrating buffer pools

From IBM MQ 9.1.0, buffer pools below the bar are deprecated. You should consider migrating your buffer pools to above the bar..

See defining your buffer pools or ALTER BUFFPOOL on z/OS for more information.

Note that a future release will ignore the value of the **LOCATION** attribute, and all buffer pools will be located above the bar.

z/OS V 9.1.0 OPMODE on z/OS

In earlier versions of IBM MQ for z/OS the availability of new functions and backward migration was controlled by the **OPMODE** parameter in the CSQ6SYSP macro. In IBM MQ 9.1.0 this no longer applies, and the **OPMODE** parameter is ignored.

None of the new function provided in IBM MQ 9.1.0 has any impact on backwards migration.

A queue manager started at IBM MQ 9.1.0 can be backwards migrated to IBM MQ 8.0.0, provided that:

1. It has previously been started at IBM MQ 8.0.0
2. The IBM MQ 9.1.0 backwards migration PTFs have been applied to the IBM MQ 8.0.0 installation before starting at IBM MQ 9.1.0.

A queue manager started at IBM MQ 9.1.0 can be backwards migrated to IBM MQ 9.0.0, provided that:

1. It has previously been started at IBM MQ 9.0.0
2. The IBM MQ 9.1.0 backwards migration PTFs have been applied to the IBM MQ 9.0.0 installation install before starting at IBM MQ 9.1.0.

Following the instructions described in “Preparing to migrate a single IBM MQ for z/OS queue manager” on page 205, will ensure that backwards migration is possible.

A queue manager started at an IBM MQ 9.0.n CD release can be forward migrated to IBM MQ 9.1.0. However it cannot be subsequently backwards migrated to IBM MQ 9.0.n CD release.

A queue manager started at IBM MQ 9.1.0 cannot be backwards migrated to a version earlier than IBM MQ for z/OS 8.0.0. Queue managers running at IBM WebSphere MQ 7.0.0, 7.0.1, or 7.1.0 must first be migrated to IBM MQ 8.0.0 or 9.0.0, before being migrated to IBM MQ 9.1.0.

Important: Before migrating to IBM MQ 9.1.0, you should first start your IBM MQ 8.0.0 or 9.0.0 queue manager in NEWFUNC mode. That is, for IBM MQ 8.0.0, start your queue manager with **OPMODE**=(*NEWFUNC,800*), and for IBM MQ 9.0.0, start your queue manager with **OPMODE**=(*NEWFUNC,900*) or **OPMODE**=(*NEWFUNC,800*).

If you, instead, leave your IBM MQ 8.0.0 or 9.0.0 queue manager in **COMPAT** mode, and then migrate to IBM MQ 9.1.0, you will be able to enable function that is not compatible with **COMPAT** mode in IBM MQ 8.0.0 or 9.0.0, such as buffer pools with a **LOCATION** of *ABOVE*. This will cause problems if you subsequently migrate back to IBM MQ 8.0.0 or 9.0.0, because you will need to change to *NEWFUNC* mode before you start your queue manager, otherwise the queue manager might not start.

The DISPLAY SYSTEM command no longer returns information on **OPMODE**. The following messages have been added to queue manager start up to indicate whether backwards migration is available or not:

- CSQY040I: `Backwards migration is not supported`

  This message will be output when backwards migration is not supported. This is the case when the queue manager:

  1. Has been created at the current release
  2. Has been migrated from an earlier CD release
  3. Has been migrated from a release earlier than IBM MQ 8.0.0
- CSQY039I: `Backwards migration is supported to version V.R.M`

This will be output when backwards migration is supported. `V.R.M` will be one of 8.0.0 or 9.0.0, which is the version of the product at which the queue manager had previously been started.

Note that this statement assumes that the IBM MQ 9.1.0 backwards migration PTFs have been installed at either IBM MQ 8.0.0 or 9.0.0, before starting at IBM MQ 9.1.0.

Related tasks

Using CSQ6SYSP

z/OS Upgrade and migration of IBM MQ on z/OS

You can install new releases of IBM MQ to upgrade IBM MQ to a new release, or version level. Multiple installations at the same or different levels can coexist on the same z/OS instance. Running a queue manager at a higher level requires migration.

From IBM MQ for z/OS 9.0.0, the way you upgrade the systems in your enterprise has changed. See IBM MQ Release Types for more information.

Important: LTS Backwards migration is possible only from a Long Term Support (LTS) release.

When you install a new VRM level of IBM MQ on z/OS using SMP/E, it creates a set of IBM MQ libraries. The libraries for different VRM levels of IBM MQ can coexist on the same instance of z/OS. You can then run different queue managers against different release levels of IBM MQ on the same z/OS instance.

If you start a queue manager running on a later release level, then migration of the queue manager to that release level is required. Even if the difference is only in the modification level, some migration might be required. The migration tasks that you must perform to migrate from one version to another are documented in “Planning to migrate IBM MQ to a later version on z/OS” on page 200; see also “Changes that affect migration” on page 73.

The strategy for upgrading queue managers at IBM MQ 9.1.0 or later is as follows:

1. Apply the backwards migration PTFs to all the queue managers you are going to upgrade. After applying the PTFs, you can run queue managers of different levels in the same queue sharing groups. You can also reverse the migration of a queue manager back to your current level.
2. Upgrade the first queue manager.
3. Check all your existing applications run correctly on this queue manager.
4. Bring all the queue managers in a queue sharing group up to the new level, and check that existing applications continue to work correctly.
5. To enable new IBM MQ 9.1.0 or later, function, restart all queue managers within the queue sharing group.

The backwards migration PTFs have two distinct purposes: [6]

1. To allow queue managers at the earlier release level to coexist with ones at the later release level. In particular for queue managers to coexist in the same queue sharing group.
2. To handle queue manager data and logs formatted using the data definitions of the later release.

Characteristics of different types of upgrade on z/OS

When you upgrade from one release to another on z/OS, the impact of the change depends on the extent of the change in VRM level. The VRM codes are explained in IBM MQ release types and versioning.

Note that migration is required if the version, release, or modification number changes.

LTS Upgrades from IBM MQ 8.0.0 or 9.0.0 to an IBM MQ 9.1.0 Long Term Support (LTS) release are reversible.

CD Upgrades to a Continuous Delivery (CD) release are not reversible.

[6] Backwards migration changes might be shipped as a single or multiple fixes.

Related concepts

“Queue manager coexistence” on page 95
Queue managers, with different names, can coexist on any server as long as they use the same IBM MQ installation. On z/OS, UNIX, Linux, and Windows, different queue managers can coexist on the same server and be associated with different installations.

“Backward migration to earlier supported releases of IBM MQ for z/OS” on page 203
After installation of a new release of IBM MQ for z/OS, you carry out queue manager migration by stopping the queue manager, which is running with the prior release of code, and restarting the queue manager using the new release of code.

“Migration methods on IBM MQ for Multiplatforms” on page 84
There are three main methods of migrating from one release to another: Single-stage migration (called a slip installation on IBM i), side-by-side migration, and multi-stage migration. Multi-stage migration is not an option for IBM i.

z/OS Migrating from earlier unsupported releases of IBM MQ for z/OS

You must take into consideration whether you are upgrading a production system, or a test system, before you undertake the migration process.

Production systems

For production systems you must, firstly, migrate the unsupported release of IBM MQ to IBM MQ 8.0 or IBM MQ 9.0, following the instructions given in the documentation for that release. For more information about where to find the documentation for older versions of the product, see Documentation for older versions of IBM MQ

You can then migrate to IBM MQ 9.1 following the instructions in this section.

Important: Ensure that your system is stable at IBM MQ 8.0 or IBM MQ 9.0, before migrating to IBM MQ 9.1, so that you have a system to revert to, if necessary.

Test systems

For a test system, it might be appropriate to migrate directly to IBM MQ 9.1 instead.

Before you begin the migration, take complete backups of your system, to ensure that you can restart from backups if you need to use the old release again.

IBM MQ 9.1 migrates IBM MQ objects and messages during the first startup at IBM MQ 9.1. If new attributes have been added to objects since the earlier release then the attributes are set to their default values.

Important: After you have migrated to IBM MQ 9.1 using this method, you cannot revert to the original version.

You can restart a queue manager, at the original version, using the full set of backups taken before migration. However, note that any changes you make to the system after the backups were taken, or while running at IBM MQ 9.1, will be lost.

z/OS Migrating IBM MQ for z/OS - order of tasks

Perform these instructions, in the order shown, to migrate a single IBM MQ for z/OS queue manager.

Before you begin

Read the information in Clustering: Best practices and Clustering: Topology design considerations to understand repositories.

About this task

The tables within this topic show the tasks required in each part of the process to migrate IBM MQ for z/OS, and the order in which these tasks must be done.

Notes:

- You must perform the tasks in the following order:

  1. Before migration
  2. Migrating to the next release
  3. Post-migration tasks

  and the order of tasks within each table.

Table 13. Before migration

Task	For your own use
1. Review the security control of your system	
2. Make your existing queue manager ready for migration	
3. Install the new code	
4. Perform a backup operation	
5. Check the MSTR and CHIN address spaces	
6. Restart your IBM MQ system	
7. Review C language channel exits	
8. Bind and grant execute authority for the Db2® plans	
9. Update the STEPLIB concatenations for Batch, TSO, and RRS applications	
10. Update the libraries you added to connect CICS to the queue manager	
11. Update the libraries for the operations and control panels to the latest version IBM MQ libraries	
12. Update system libraries to format IBM MQ dumps using the Interactive Problem Control System (IPCS)	
13. Consider the migration of other server applications	
14. Configure Advanced Message Security . **Note:** You only need to carry out this step if you are using AMS.	
15. Prepare the mqweb server for migration	

Table 14. Migrating to the next release

Task	For your own use
1. Update the initialization input data sets	
2. Stop or disconnect all the applications using the queue manager	
3. Stop the queue manager and its channel initiator	

Table 14. Migrating to the next release (continued)

Task	For your own use
4. Update STEPLIB for MSTR and the channel initiator	
5. Remove the OPMODE parameter from ZPARMS	
6. Configure a queue manager to use Advanced Message Security. **Note:** You only need to carry out this step if you have configured AMS.	
7. Start the queue manager	
8. Optionally, if a problem occurs, revert the queue manager to a previous version	
9. Optionally, migrate the mqweb server	

Table 15. Post migration tasks

Task	For your own use
1. Check the changes in behavior	
2. Modify, backup, and other administrative, jobs	
3. Update the ZPARM module, if required	
4. Post migration tasks for Advanced Message Security	
5. Perform a full regression test	
6. Migrate client applications	
7. Exploit the new functions provided by the migrated queue manager	
8. Optionally, stop the mqweb server for previous versions	

z/OS *Planning to migrate IBM MQ to a later version on z/OS*

Create a migration plan for IBM MQ for z/OS to migrate to the later version.

Before you begin

If there are concepts about migration you do not understand, see “Migration concepts and methods” on page 76.

If you are migrating to IBM MQ 9.1 from IBM WebSphere MQ 7.1, you must first migrate to IBM MQ 8.0 or IBM MQ 9.0 and restart each queue manager with OPMODE=(NEWFUNC,800) or OPMODE=(NEWFUNC,900).

About this task

Use the following steps as a guide to creating your own plan to migrate your queue managers to a later version. Incorporate the task to migrate a queue manager, “Migrating IBM MQ for z/OS - order of tasks” on page 198, into your plan.

Migration plan overview for your enterprise	
Migration phase	**Required tasks**
Phase I, before migration. See “Preparing to migrate a single IBM MQ for z/OS queue manager” on page 205 for further information.	Prepare each queue manager in your enterprise for migration.
Phase II, migrate each single queue manager in the order listed. See “Migrating a single IBM MQ z/OS queue manager to the next version of the product” on page 209 for further information.	Carry out this process for each queue manager. For queue managers in a cluster, migrate full repository queue managers before migrating any partial repository queue managers.
Phase III, post migration. See “Post migration tasks” on page 217 for further information.	Carry out a full regression test, and then explore the new function available to you. Optionally, at any time in the process, migrate your client libraries if necessary, recompile your clients using the additional features provided by the new version, and deploy the clients.

Procedure

1. Review the IBM MQ system requirements for the later version.

   See System requirements for IBM MQ.
2. Review all the changes in the product that affect you.

   For further information, see:

   - V 9.1.0 What's new and changed in IBM MQ 9.1.0
   - CD What's new and changed in IBM MQ 9.1.x Continuous Delivery
3. Review performance changes.

   See IBM MQ - Performance documents.
4. Review the backward and coexistence (or migration and toleration) PTFs for your current version of the product. See IBM MQ Support, Migration PTFs.

   These PTFs must be applied to your current version of the product to enable you to revert your queue managers to the current version, after the queue managers have been started at the target version.

   Note, that you can have different versions of queue managers coexisting in the same queue sharing group.

   If you are unsure which migration PTFs you require, run the following SMP/E command:

   ```
   V 9.1.0
   REPORT MISSINGFIX ZONES(mqtgtzone) FIXCAT(IBM.Coexistence.MQ.V9R1M0)
   ```

   See FIXCAT and IBM MQ Migration Installation for further information.

   ⚠ **Attention:** If a PTF requires a rebind of Db2 plans, the PTF is shipped with ++HOLD(ACTION), indicating the need for this process. In such a case, see Migrating Db2 tables to bind the plans before starting migration.

   Other FIXCAT categories are listed in IBM Fix Category Values and Descriptions.
5. Plan to install the later version early code, and activate for all queue managers on the LPAR.

   See Installing early code for more information.

   Note that:

Before migration, all systems that are running queue managers that you plan to migrate to the later version must have the early code for that version installed and running. Queue managers in queue sharing groups that contain queue managers that are to be migrated, must also be running the early code.

A queue manager must use the early code from the same release level, or a later release level.

6. Consider using aliases for the IBM MQ libraries.

   For example, use the IDCAMS utility with the DEFINE command:

   ```
   DEFINE ALIAS(NAME(MQM.SCSQANLE)RELATE(MQM.V910.SCSQANLE))
   ```

   You can use `MQM.SCSQANLE`, where applicable, in your STEPLIB, and it resolves to the actual data set.

   When you migrate to a new release, change the alias definition, rather than changing all the places in your JCL where the library is referenced.

   This process has the most benefit for your server application programs, because you can get all of the programs to refer to the new libraries at the same time.
7. Plan the sequence and timing of queue manager migrations.
   - You must install the backward migration (or migration and toleration) PTF to bring the previous version queue managers up to the latest maintenance level for that version.
   - You must install the PTF on all members of a queue sharing group, before you migrate any queue managers to the later version. You can install the PTF one member at a time, leaving the other members running.
   - If the queue manager is a member of a queue manager cluster, you must consider the order of migration of queue managers in the cluster; see "Migrating a queue manager cluster" on page 219.
   - Check that any products that require the previous version of the product also support the new version.
   - Queue managers in a QSG, and queue managers in a cluster can be migrated in parallel, but at any time there should be enough queue managers working in the QSG, and cluster, to ensure that your business can operate satisfactorily while a staged migration takes place.
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.

   Update the IBM MQ library in the application STEPLIB concatenations to the later version.

   Consider whether the application must be able to run on both the previous version and the later version. 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. V 9.1.0 Decide on what regression tests to perform after migrating.

    Include the procedures and applications you identified in steps "8" on page 202 and "9" on page 202 in your regression tests.
11. Review the tasks to customize z/OS, and the queue manager. Plan how to change the queue manager definitions and started task JCL to migrate your queue managers to the later versions.
12. Check that you are using a supported level of assembler or compiler.

    You can write IBM MQ applications using any compiler capable of generating standard OS linkage to the IBM MQ stub routines.

    Some of the data types used by IBM MQ API calls are not supported on some older compilers. You might require a more recent compiler. The following limitations are known:

    a. Assembler copy books contain blank lines, which are not tolerated by assemblers earlier than **HLASM**.

   b. Some older releases of PL/I do not support fixed `bin(63)` type. A macro defines such fields as `char(8)` when an earlier compiler is detected.
   c. Some older releases of COBOL do not support function-pointers, which are used by the `MQCB` API.
13. Plan any changes to libraries required by your applications and channel exits.
14. Plan to migrate your IBM MQ MQI client installations to the later version.
15. Plan to migrate your client and server applications to use new functions in the later version.
16. Plan to migrate other vendor software, such as WebSphere Application Server, or CICS® to use the later version.

    Update the IBM MQ libraries in the `STEPLIB` and `DFHRPL` concatenations of your CICS region JCL and restart CICS.
17. Review any other installed SupportPacs for their applicability to the later version.

What to do next

Do the task, "Preparing to migrate a single IBM MQ for z/OS queue manager" on page 205.

About the CICS-MQ adapter
IBM MQ Support, Migration PTFs
IBM MQ - SupportPacs by Product

z/OS *Backward migration to earlier supported releases of IBM MQ for z/OS*

After installation of a new release of IBM MQ for z/OS, you carry out queue manager migration by stopping the queue manager, which is running with the prior release of code, and restarting the queue manager using the new release of code.

Maintenance in a queue sharing group

V 9.1.0 In a queue sharing group, individual queue managers can be migrated forwards to IBM MQ 9.1.0, while those that remain at either IBM MQ 8.0.0 or IBM MQ 9.0.0 can continue to function. This allows you to upgrade queue sharing group queue managers to IBM MQ 9.1.0 at different times, maintaining the high availability of the queue sharing group.

The function required to enable lower level queue managers to tolerate IBM MQ 9.1.0 additions to QSGDISP(GROUP) and QSGDISP(SHARED) objects is incorporated in the same authorized program analysis reports (APARs) which provide backward migration capability.

Code levels supported

V 9.1.0 Migration support is provided from IBM WebSphere MQ 7.1.0, IBM MQ 8.0.0, and IBM MQ 9.0.0 to IBM MQ 9.1.0.

Important: If your enterprise is running IBM WebSphere MQ 7.1.0, you must first migrate to IBM MQ 8.0.0 or IBM MQ 9.0.0 before you migrate to IBM MQ 9.1.0.

V 9.1.0 The backward migration APARs are PI95928 for IBM MQ 8.0.0, and PI95939 for IBM MQ 9.0.0.

Important: LTS PTFs for these APARs must be applied on IBM MQ 8.0.0 or IBM MQ 9.0.0 prior to attempting to fall back from IBM MQ 9.1.0 Long Term Support (LTS) release.

CD Backwards migration is not supported for a Continuous Delivery (CD) release.

PTFs for these APARs are the *Migration and Toleration PTFs for IBM MQ 9.1.0* described in Planning for migration to the latest release.

Service has been discontinued for versions of the product prior to IBM WebSphere MQ 7.1.0. No backward migration capability is available for these versions.

V 9.1.0 The IBM MQ for z/OS 9.1.0 early code installed in the link pack area (LPA) is downward compatible. The code supports queue managers running at IBM MQ 8.0.0 and any service release, and IBM MQ 9.0.0 and any service release.

Once updated to the IBM MQ 9.1.0 level, and the queue manager subsystem refreshed using the REFRESH QMGR TYPE(EARLY) command, the early code need not be changed for any subsequent forward or backward migration activity

Message

```
CSQ3111I <cpf> CSQYSCMD - EARLY PROCESSING PROGRAM IS V9.1 LEVEL 008-000
```

is displayed during startup in the queue manager joblog and indicates that the queue manager is using the correct level of early code.

Limitations and restrictions

V 9.1.0 Queue managers started at IBM MQ for z/OS 9.1.0 can be backwards migrated to either IBM MQ 8.0.0 or IBM MQ 9.0.0 for as long as the queue manager had previously been started at those earlier versions, with the IBM MQ 9.1.0 backwards migration PTFs applied.

V 9.1.0 The **OPMODE** parameter in CSQ6SYSP has been removed at IBM MQ 9.1.0 and is no longer supported. See OPMODE for more information.

V 9.1.0 You should start IBM MQ 8.0.0 and IBM MQ 9.0.0 queue managers with **OPMODE**(*NEWFUNC,800*) or**OPMODE**(*NEWFUNC,900*) respectively, before migrating to IBM MQ 9.1.0.

CD Backwards migration is not supported for a Continuous Delivery (CD) release.

Certain connection types (IMS, BATCH and RRSBATCH used by WAS and Db2 stored procedures) allow an application to connect to multiple queue managers concurrently. If required, these queue managers can be running different levels of IBM MQ code. In such a scenario, the adapter code (usually referenced through a STEPLIB DD statement or environment variable) must be loaded from libraries corresponding with the highest level of the queue managers connected. This ability for the adapter code to support connections to older queue managers means that in a backward migration scenario it is possible to just restart the MSTR and CHIN procedures with the back level code, and not change connecting jobs.

The operations and controls ISPF panels, CSQOREXX, from IBM MQ for z/OS 9.0.0, are able to connect to and administer queue managers from a prior release. However, the ISPF panels from lower releases are not able to connect to IBM MQ for z/OS 9.0.0 or later. When migrating, or during fall back, either use the same version ISPF panels as the level of code the queue manager is running, or use CSQOREXX from the higher release of code. In a mixed level queue sharing group, the IBM MQ for z/OS 9.0.0, or later, panels must be used to administer IBM MQ 8.0.0 or IBM WebSphere MQ 7.1.0 queue managers, as ISPF panels from earlier releases do not tolerate responses from any IBM MQ 9.0.0, or later, queue managers.

V 9.1.4 If the queue manager to be backwards migrated has z/OS data set encryption policies applied to one or more of its active logs or page sets V 9.1.5 , or SMDS, then these policies need to be removed, and the data decrypted, prior to migrating to IBM MQ for z/OS 9.1.0 or earlier. This process is described in backwards migration considerations when using z/OS data set encryption

z/OS *Preparing to migrate a single IBM MQ for z/OS queue manager*

Review the z/OS and IBM MQ customization steps, and change any customizations before starting any queue managers with the libraries of the latest version. Follow the steps to prepare a single IBM MQ queue manager on z/OS for migration.

Before you begin

You can continue to run the queue manager on the previous version until it is ready to switch to the latest version. Preparing for the switch can be a long process. Switching from the earlier version to the latest version is a quick process. The switch to the latest version occurs when you restart the queue manager.

Tips:

1. To migrate applications, or the queue manager, when moving to a new release of IBM MQ, without changing STEPLIB, create data set aliases such as `MQM.qmgr.SCSQLOAD`, and reference them in JCL. Map the aliases to the real data sets, such as `MQM.MQV900.SCSQLOAD` or `MQM.MQV910.SCSQLOAD`.
2. You can use the z/OS command `D GRS,SYSTEM,RES=(*,MQM.V910.SCSQLOAD)` to display which jobs are using the specified data set, and so identify which jobs and JCL need to be changed.

About this task

The steps are based on the setup procedure for new queue managers; see Customizing your queue managers.

To prepare to migrate an IBM MQ queue manager on z/OS, you need to carry out the detailed steps in this topic, using the links within this overview.

1. Review the security control of your system; see step “1” on page 205.
2. Make your existing queue manager ready for migration; see step 2.
3. Install the new code and make target libraries available to all z/OS systems that are running queue managers, and grant access; see step 3.
4. Perform a back up operation of each queue manager in your enterprise; see step 4.
5. Review definitions of the user IDs for the queue manager (MSTR) and channel initiator (CHIN) address spaces; see step 5.
6. Restart your IBM MQ systems; see step 6.
7. Review C language channel exits; see step 7.
8. V 9.1.0 Bind, and grant execute authority for the Db2 plans; see step 8.
9. Update the STEPLIB concatenations for Batch, TSO, and RRS applications; see step 9.
10. Update the libraries you added to connect CICS to the queue manager; see step 10.
11. Update the libraries for the operations and control panels to the latest version IBM MQ libraries; see step 11.
12. Update system libraries to format IBM MQ dumps using the Interactive Problem Control System (IPCS); see step 12.
13. Consider the migration of other server applications; see step 13.
14. Configure Advanced Message Security (AMS); see step 14.
15. Prepare the mqweb server for migration; see step 15.

Procedure

1. Review your security control for queue sharing groups, the channel initiator, and all queue managers accessing the coupling facility list structures.
2. Make your IBM MQ configuration ready for migration.

   a) Apply current maintenance to the current version libraries.

   Refer to the Preventive Service Planning (PSP) bucket for your version of IBM MQ; see PSP Buckets - How to find them on Web.

   b) Apply the migration and toleration PTFs to the version of the IBM MQ code that your enterprise uses; see IBM MQ Support, Migration PTFs.

      Note that the "migration and toleration" PTFs are also known as the "backwards migration" PTFs; they are the same PTFs.

      If you are unsure which migration PTFs you require, run the following SMP/E command:

      V 9.1.0

      ```
      REPORT MISSINGFIX ZONES(mqtgtzone) FIXCAT(IBM.Coexistence.MQ.V9R1M0)
      ```

      See FIXCAT and IBM MQ Migration Installation for further information.

      ⚠ **Attention:** If a PTF requires a rebind of Db2 plans, the PTF is shipped with ++HOLD(ACTION), indicating the need for this process. In such a case, see Migrating Db2 tables to bind the plans before starting migration.

      Other FIXCAT categories are listed in IBM Fix Category Values and Descriptions.

      V 9.1.0 There is an additional category of `TargetSystem-RequiredService.MQ.V9R1M0` enabling other products to run with IBM MQ 9.1.0.

3. Install the new code and make target libraries available to all z/OS systems that are running queue managers, and grant access.

   You must carry out the following procedure for each z/OS system.

   a) Copy the IBM MQ target libraries to the system, and install the early code for the new version (once for each z/OS system).

   b) Make the latest version early code and target libraries available on each z/OS system that is running queue managers.

      This updates the LPA. See Update the z/OS link list and LPA for more information.

   c) APF authorize the load libraries and grant access to the data sets using your external security system.

      See APF authorize the IBM MQ load libraries for more information.

      Ensure that the APF lists in SYS1.PARMLIB are updated to ensure that any changes made dynamically remain in effect after an IPL.

   d) Copy the file system zFS and mount it read only.

      You only need zFS if the IBM MQ for z/OS Unix System Services Component is installed. See the Program Directory for further information. For download links for the Program Directories, see IBM MQ for z/OS Program Directory PDF files.

4. Perform a back up operation for each queue manager in your enterprise, so that you have a copy of all objects and JCL before you make any changes.

   This makes rolling back to the current system easier, if you require to do so.

   a) Back up your IBM MQ defined objects, for example using CSQUTIL COMMAND MAKEDEF(..)

      See Using the COMMAND function of CSQUTIL for more information.

   b) Back up:

      - Started task procedures for the queue manager and the channel initiator.
      - The Initialization input data sets used in the CSQINP1 and CSQINP2 concatenations
      - The system parameter module (ZPARM) libraries
      - The JCL libraries containing the configuration definitions for your queue manager
      - Other libraries containing queue manager configuration or procedures.

Note: You might also make a back up of page sets, BSDSs, and active logs as a fallback option. See How to back up and recover page sets for more information on backing up IBM MQ resources.

5. Check that the MSTR and CHIN address spaces run under user IDs that have OMVS segments defined, with a valid UID, to enable calling UNIX System Services (USS).
6. Restart your IBM MQ system to run with the migration and toleration PTFs.

   Check the `qmgr.REFRESH.QMGR` security profile is set up, and that you have authority to issue the REFRESH QMGR command.

   While the queue manager is stopped, refresh the queue manager so that it uses the new early code using the command REFRESH QMGR TYPE(EARLY). See REFRESH QMGR for more information.

   a) Restart the queue managers and monitor the whole system in your enterprise closely to ensure that there are no issues.

      Depending on the size and complexity of your enterprise this can take a considerable length of time, so you must plan for this in your migration schedule.

      If you require fall back at this stage, use normal maintenance procedures to revert to the code for the previous version before PTF application.
7. Review C language channel exits.

   Ensure your C language channel exits are using the following statement:

   ```
   #pragma environment(function-name)
   ```

   as defined in the C systems programming environment for system exits, described in the z/OS C/C++ Programming Guide.
8. V 9.1.0

   If you are using queue sharing groups, bind, and grant execute authority for the Db2 plans.

   You must have applied the migration and toleration PTFs to all the queue managers in the queue sharing group, before proceeding to bind and grant execute authority to the Db2 plans.

   Customize and run the CSQ45BPL and CSQ45GEX samples in hlq.SCSQPROC. Tailor these members to your environment, using your Db2 subsystem names and data set names.

   The header information in CSQ45BPL and CSQ45GEX describes how to customize the samples:

   - CSQ45BPL of hlq.SCSQPROC contains the plan names required for the latest version of IBM MQ
   - CSQ45GEX of hlq.SCSQPROC contains the authorities required

   See steps 5 and 6 of Set up the Db2 environment.
9. If you are using queue sharing groups, run CSQ5PQSG.

   You must specify the **MIGRATE QSG** or **MIGRATE DSG** function, to verify that all queue managers in the QSG or DSG are at a version that is compatible with IBM MQ 9.1.0.

   If any incompatible queue managers are found, issue the DISPLAY SYSTEM command on each queue manager in the QSG. Identify any queue managers where DISPLAY SYSTEM reports a compatibility level of 701 or 710, for example:

   ```
   
       OPMODE(COMPAT, 701)
       OPMODE(COMPAT, 710, 710)
   ```

   Recompile the system parameters module (ZPARM) for each such queue manager with OPMODE=(NEWFUNC,800) or OPMODE=(NEWFUNC,900) and the queue managers restarted, prior to rerunning CSQ5PQSG to validate compatibility with V910.
10. Update the libraries you added to STEPLIB concatenations, to make Batch, TSO, and RRS adapters available to applications, so that they can always load the highest version IBM MQ library in the system.

    Change the STEPLIB for the Batch, TSO, and RRS applications to reference the libraries of the new version of the product, then restart the applications.

See Set up Batch, TSO, and RRS adapters for more information.

IBM MQ libraries include:

thlqual.SCSQANLx
This library contains error message information for your national language. The letter 'x' represents the letter for your national language.

thlqual.SCSQAUTH
This library contains the code that the applications use.

Notes: You can:

a. Connect applications that reference the latest version IBM MQ libraries in STEPLIB to a queue manager that is running on the latest version, or an earlier version. You must not connect applications that reference a STEPLIB from an earlier version to a queue manager running on a later version.
b. Use the following command, replacing *thlqual* with the High Level Qualifier for your installation, to check which jobs are running with the specified library:

   ```
   TSO ISRDDN ENQ 'thlqual.SCSQANLE'
   ```

   You can then modify the JCL for those jobs accordingly.
c. Make use of data set aliases for convenience, to avoid the need to change the data sets referenced by STEPLIB.

11. Update the libraries you added to connect CICS to the queue manager.

    You must update the IBM MQ libraries in the STEPLIB and DFHRPL concatenations of your CICS region JCL and restart CICS. You are then able to use the most recent IBM MQ features.

    The connection between IBM MQ and CICS is provided by CICS libraries that you must update. Without this change, you are not able to use the most recent IBM MQ features. You must change the SCSQCICS library in the DFHRPL concatenation provided by IBM MQ, and also the STEPLIB concatenation.

    For each CICS region that is connected to an IBM MQ queue manager, ensure that there is a separate CICS started procedure JCL.

    This ensures that the modification of reference to a certain version of IBM MQ libraries in the CICS started procedure JCL only has impact for that single CICS region. In this way you can migrate one queue manager, and only the CICS region or regions connected to it, which makes staged migration possible.

    CICS STEPLIB has thlqual.SCSQAUTH, and DFHRPL has thlqual.SCSQCICS, thlqual.SCSQLOAD, and thlqual.SCSQAUTH. For more information, see Setting up the CICS - IBM MQ adapter.
12. Update the libraries for the operations and control panels to the latest version IBM MQ libraries.

    For more information, see Set up the operations and control panels.

    Note: You can connect the operations and control panel that references the latest version IBM MQ libraries to the queue manager that is running on the latest version, or an earlier version. You must not connect the operations and control panel that references IBM MQ libraries from an earlier version to a queue manager running on a later version.
13. Update system libraries to format IBM MQ dumps using the Interactive Problem Control System (IPCS).

    For more information, see Include the IBM MQ dump formatting member.
14. Migrate other software, such as WebSphere Application Server for z/OS, IBM Integration Bus, or IMS to use the version of IBM MQ that you need.

    a) WebSphere Application Server for z/OS

    If you are running in an application server environment where a bindings connection is being used, you need to update the WAS STEPLIB with IBM MQ libraries.

See IBM MQ libraries and the WebSphere Application Server for z/OS STEPLIB for further information.

You also need to configure the IBM MQ messaging provider with native libraries from the new version of the IBM MQ installation; see Configuring the IBM MQ messaging provider with native libraries for further information.

Use the latest level of native libraries in USS.

b) IMS

For more information, see Setting up the IMS adapter.

15. Configure Advanced Message Security (AMS).

    If the queue manager is configured to use Advanced Message Security (AMS), perform the steps in the Preparing to migrate Advanced Message Security section of the Migrating Advanced Message Security topic.

16. If you have configured the IBM MQ Console or REST API, you might need to prepare the mqweb server for migration.

    The mqweb server that hosts the IBM MQ Console and REST API can only connect directly to queue managers at the same version of IBM MQ.

    If you only have one queue manager on the z/OS system, you do not need to do anything now. You can migrate the existing mqweb server at the same time as the queue manager.

    If you have more than one queue manager on the z/OS system, that will be running at different versions of IBM MQ during migration, create a new mqweb server at the latest level.

    a) Create a new mqweb server by following the procedure in Create the mqweb server.

    b) Create a started task procedure for the new mqweb server by following the procedure in Create a procedure for the mqweb server.

    c) Copy any relevant configuration settings from the `mqwebuser.xml` configuration file of the existing server to the `mqwebuser.xml` file of the new mqweb server.

       The `mqwebuser.xml` file is in the *`WLP_user_directory`*`/servers/mqweb` directory, where *WLP_user_directory* is the directory that was specified when the **`crtmqweb`** script ran to create the mqweb server definition.

       Configure the new server to use different HTTP and HTTPS ports to any existing servers.

    d) Start the new mqweb server by issuing the MVS™ **`START procname`** command, where *procname* is the name of the new mqweb server started task procedure.

Results

You have prepared your IBM MQ queue manager on z/OS for migration.

What to do next

Follow the instructions in "Migrating a single IBM MQ z/OS queue manager to the next version of the product" on page 209 to migrate the queue manager.

z/OS *Migrating a single IBM MQ z/OS queue manager to the next version of the product*

Carry out the instructions in this topic to migrate a single IBM MQ queue manager on z/OS.

About this task

To migrate an IBM MQ queue manager on z/OS to a different version, you need to carry out the:

- Process described in "Preparing to migrate a single IBM MQ for z/OS queue manager" on page 205
- Detailed steps in this topic, using the links within this overview.

1. Update the initialization input data sets; see step “1” on page 210.
2. Stop or disconnect the applications; see step “2” on page 210.
3. Stop the queue manager and its channel initiator; see step “3” on page 210.
4. Update STEPLIB for the queue manager and the channel initiator; see step “4” on page 210.
5. Remove the OPMODE parameter from ZPARMS if you are migrating to IBM MQ 9.1; see step “5” on page 211
6. Configure a queue manager to use Advanced Message Security; see step “6” on page 211.
7. Start the queue manager and its channel initiator; see step “7” on page 211.
8. Optionally, revert the queue manager to a previous version; see step “8” on page 211.
9. Optionally, migrate the mqweb server; see step “9” on page 211.

Procedure

1. Update the initialization input data sets.

   Each IBM MQ queue manager gets its initial definitions from a series of commands contained in the IBM MQ initialization input data sets. These data sets are referenced by the Data Definition (DD) names CSQINP1, CSQINP2, and CSQINPT defined in the queue manager started task procedure.

   See Customize the initialization input data sets for more information on changes to the initialization input data sets.

   The samples supplied for the initialization input data sets can change between versions of IBM MQ. You must review the customization you have made previously to CSQINP1, CSQINP2, and CSQINPT, and merge them into the initial definitions provided with the latest version of the product. See “Changes to initialization input data sets ” on page 211 for more information.

   Secure the server-connection channels used by clients; see Securing remote connectivity to the queue manager.
2. Stop or disconnect all the applications using the queue manager (for example, CICS, IMS, or batch) and the IBM MQ channels that are connected to other queue managers.
3. Stop the queue manager and its channel initiator.
4. Update STEPLIB for the queue manager (MSTR) and the channel initiator (CHIN).

   a) Update your procedure to start the queue manager.

      Change the STEPLIB for the queue manager to reference the new version of the libraries.
      See Create procedures for the IBM MQ queue manager.

      IBM MQ uses z/OS memory objects above the bar for some functions. You must allow the queue manager to access storage above the bar.
      Your installation might have customized the SMFPRMxx member of SYS1.PARMLIB, or the **IEFUSI** exit to provide a default limit for jobs using virtual storage above the 2 GB bar. Check these limits give sufficient memory for a queue manager. A reasonable starting allocation is 2 GB. The message CSQY220I displays the amount of virtual storage currently used and available.
      If your installation does not have a default limit for storage above the bar, or if you want to use a different limit for your queue manager, you can provide a queue manager-specific restriction on the amount of virtual storage available above the bar for memory objects by coding a **MEMLIMIT** parameter on the JCL of the queue manager stored procedure, xxxxMSTR, for example:

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

      MEMLIMIT defines memory available above the bar; see Address space storage

   You must allow the queue manager to access storage above the bar because IBM MQ uses memory above the bar.

If insufficient storage is available above the bar, the queue manager reports this when starting, and stops.

b) Update your procedures for the channel initiator.

   Change the STEPLIB for the channel initiator to reference the new level of the product libraries. See Create procedures for the channel initiator.

5. Remove OPMODE from ZPARMS.

   See Update the System parameter module for more information.
6. If the queue manager is configured to use Advanced Message Security (AMS), perform the steps in Migrating Advanced Message Security .
7. Start the queue manager and its channel initiator.

   Test that everything is working correctly. If there is a problem starting the queue manager, consider reverting the queue manager to a previous version; see step “8” on page 211.
8. If a problem occurs when starting the queue manager, you might need to consider backward migration; see Reverting a queue manager to a previous release.
9. If you have configured the IBM MQ Console or REST API, migrate the mqweb server.

   If you only have one queue manager on the z/OS system, migrate the mqweb server to the latest version.

   a) Edit the mqweb server started task.

      - Change STEPLIB to reference the new version of the libraries.
      - Change INSTDIR, PATH and LIBPATH to reference the new version of the IBM MQ files in z/OS UNIX System Services.
      - Review JAVA_HOME, and change it to reference the latest 64 bit version of Java on your system, if necessary.

      For more information on configuring the mqweb server started task procedure, see Create a procedure for the mqweb server.

   b) Stop and restart the mqweb server started task.

   c) Test that the queue manager can be accessed in the IBM MQ Console and REST API.

   If you have more than one queue manager on the z/OS system, running at different versions of IBM MQ, you should already have created a new mqweb server at the latest level. Use the new mqweb server to access queue managers that have been migrated to the latest level. You might need to change the HTTP host name or port used by applications that use the REST API.

Results

You have migrated your IBM MQ for z/OS queue manager to the latest release.

What to do next

Follow the instructions in “Post migration tasks” on page 217 to complete the migration process.

z/OS V 9.1.0 *Changes to initialization input data sets*

The content of some initialization input data set samples changed in IBM MQ 9.1.

These are the important changes to the initialization input data sets since IBM MQ for z/OS 8.0.

CSQ4INSG

- The default value of **ADOPTCTX** for SYSTEM.DEFAULT.AUTHINFO.IDPWOS has changed to *YES*.
- Local queue SYSTEM.REST.REPLY.QUEUE added, to support the IBM MQ REST API.

Review the changes, and update the customized versions you are currently using as required.

Note:

The changes to the initialization data sets are required in Create procedures for the IBM MQ queue manager and Customize the initialization input data sets.

z/OS *Migrating Advanced Message Security on z/OS*
Advanced Message Security for z/OS (AMS) extends IBM MQ to provide a high level of protection for sensitive data flowing through the IBM MQ network using a public key cryptography model.

In IBM MQ for z/OS releases prior to IBM MQ 8.0, AMS was provided as a separate product. This topic describes the tasks required to migrate the AMS configuration on z/OS from that used in IBM WebSphere MQ 7.0.1 and earlier, to that used in IBM MQ 8.0 and later. These steps supplement those required to migrate a single IBM MQ for z/OS queue manager where AMS is not configured. AMS must be migrated at the same time as the queue manager, it is not supported to use Advanced Message Security 7.0.1 with IBM MQ for z/OS IBM MQ 9.1.

To enable AMS on a newly created IBM MQ for z/OS queue manager, or on a queue manager that has already been migrated to IBM MQ 9.1, see Installing Advanced Message Security on z/OS.

V 9.1.3 From IBM MQ 9.1.3, AMS is a feature of IBM MQ Advanced for z/OS and is no longer available as a separate product.

V 9.1.3 If you are installing a queue manager at IBM MQ 9.1.3, there is no longer a separate enablement module for AMS (see Installing Advanced Message Security on z/OS.

For information about licensing for Advanced Message Security for z/OS, see IBM MQ license information and IBM MQ for z/OS product identifiers.

Preparing to migrate Advanced Message Security on z/OS

To prepare to migrate an IBM MQ queue manager on z/OS using Advanced Message Security 7.0.1 or earlier, you must perform the steps in this section in addition to those listed in “Preparing to migrate a single IBM MQ for z/OS queue manager” on page 205.

1. Install the Advanced Message Security for z/OS enabling product and make the target libraries available to all z/OS systems that are running queue managers that will use AMS. You must carry out the following procedure for each z/OS system:
    a. Copy the AMS target libraries to the system.
    b. APF authorize the thlqual.SDRQAUTH target library and grant access to this data set using your external security system, see APF authorize the IBM MQ load libraries.

       Note: V 9.1.3 This step is not applicable to IBM MQ 9.1.3 or later.
    c. Ensure the LPA contains the AMS module CSQ0DRTM, see Update the z/OS link list and LPA.
    d. Ensure the program properties table (PPT) contains an entry for CSQ0DSRV, see Update the z/OS program properties table.
2. For each queue manager, set up the started task user for the AMS address space. In Advanced Message Security 7.0.1, two address spaces are used, one for the main task and another for the data services task. In IBM MQ 9.1 these are combined in to a single address space called *qmgr*AMSM. Either set up a new user for the IBM MQ 9.1 AMS address space, or grant additional authorities to one of the existing AMS started task users. See Set up the Advanced Message Security started task user for information on how to set up the started task user. If you do not use the existing data services address space user you will need to replicate the **drq.ams.keyring** key ring for the user ID associated with the IBM MQ 9.1 *qmgr*AMSM address space. See Using certificates on z/OS for information on how to set up the AMS key ring.

Migrating Advanced Message Security on z/OS

To migrate an IBM MQ queue manager on z/OS using Advanced Message Security 7.0.1 or earlier, before restarting the queue manager you must perform the steps in this section in addition to those listed in “Migrating a single IBM MQ z/OS queue manager to the next version of the product” on page 209.

1. Take a copy of the *qmgr*AMSM task for Advanced Message Security 7.0.1, in case you need to revert to your previous system.
2. Configure the queue manager to use AMS by updating the system parameter module to set SPLCAP(YES) using CSQ6SYSP, see Tailor your system parameter module and Using CSQ6SYSP.
3. V 9.1.3 Enable the AMS feature by setting AMSPROD to either AMS, ADVANCED, or ADVANCEDVUE depending on your licence entitlement. See using CSQ6USGP for more information on the CSQ6USGP macro.
4. Create or update the started task procedure for the *qmgr*AMSM address space, see Create procedures for Advanced Message Security.

Post migration tasks for Advanced Message Security on z/OS

After you have migrated an IBM MQ queue manager on z/OS that uses AMS you must perform the following tasks.

1. In IBM MQ 8.0 and later, the AMS address space is started and stopped automatically by the queue manager. If you have automation to manage the main task and data services task for Advanced Message Security 7.0.1 or earlier, this should be removed. You must also review any automated console commands for AMS because some have changed since IBM MQ 8.0.
2. Delete the started task procedures for the IBM WebSphere MQ 7.0.1 data services task and the IBM WebSphere MQ 7.0 main task if these were not called *qmgr*AMSM.

z/OS *Migrating the database tables on Db2 on z/OS to MQ V8.0 or later*

If your database is Db2 on a z/OS system, you must complete the following steps to migrate between from IBM WebSphere MQ File Transfer Edition 7.0.3 to IBM WebSphere MQ File Transfer Edition 7.0.4, and from IBM WebSphere MQ File Transfer Edition 7.0.4 to Managed File Transfer for IBM MQ 8.0 or later. The Db2 tables have different structures from previous releases. For example, there are new columns in some tables, and some variable characters columns can be larger, so the tables from previous releases have to be migrated to the Db2 format.

About this task

Important:

If you are migrating to IBM MQ 9.1 from IBM WebSphere MQ 7.1 or earlier, you must first migrate to an interim version. See Migration paths.

Managed File Transfer for IBM MQ 8.0 or later supports transferring very large files, where the size of the file is larger than can be stored in an integer (32 bit) number. There is a definition that uses BIGINT (64 bit) numbers. You can choose to use integer definitions, called `ftelog_tables_zos.sql`, or BIGINT definitions, called `ftelog_tables_zos_bigint.sql`, located in the `prod/mqf/sql` directory.

The BIGINT data type is available in IBM WebSphere MQ File Transfer Edition 7.0.4 Fix Pack 3, or Managed File Transfer for IBM WebSphere MQ 7.5.0 Fix Pack 1 or later.

To enable use of BIGINT data types, you must be using Db2 9.1 or later. INTEGER data types are used for fields which denote the sizes of files that are transferred and the table ID associated with each transfer. If you want to log transfers with file sizes greater than 2 GB, or if you want to store more than 2,147,483,648 individual transfers in your database you must use the BIGINT SQL file.

The following list outlines the processes you need to follow:

1. You have existing tables from IBM WebSphere MQ 7.0. These tables have a schema, for example `FTELOG`.
2. Create IBM MQ 8.0 tables using a different schema name, for example, FTEV8. This allows you to copy data from `FTELOG.table` to `FTEV8.table`.
3. Copy the data to the new table
4. Set the generated ID values in the new tables

5. Run the **fteMigrateLogger** command to move the properties file to a new place in the directory structure.
6. Edit the logger properties file to specify the new schema (FTEV8)
7. Edit the existing Logger JCL to use the IBM MQ 8.0 Managed File Transfer libraries.
8. Start the logger.
9. Once the logger is working you can delete the FTELOG tables.

In the following description, the Managed File Transfer product is installed in the /HMF8800 directory in USS.

Procedure

1. If you have not already stopped your database logger, stop your database logger using the **fteStopDatabaseLogger** command in USS or **P loggerjob**.
2. Issue the command **ls /HMF8800/mqft/sql** to list the SQL files in the directory.

   If you are going to use BIGINT numbers copy, ftelog_tables_zos_bigint.sql to your home directory, otherwise, copy ftelog_tables_zos.sql to your home directory.
3. Edit the file you moved to your home directory:
   a) Change ftelog to the schema name for the new tables.
   b) Ensure each index has a unique name.

      To do this, in an edit session:

      i) Exclude all lines.
      ii) Find 'CREATE UNIQUE INDEX ' ALL
      iii) Change _KEY _K8Y ALL NX
4. Check the file to make sure all of the statements are within column 71.

   If the statements are not within column 71, split the line before column 71.
5. You might be able to use this file as input to SQL, or you might want to copy it to a PDS. To do this, edit the PDS and use the **copy** command, specifying the directory and file name.
6. Check the definitions with your Db2 administrator, as there might be site standards that you need to follow.
7. Carry out the following:
   a) Copy the _zos_704-800.sql file, located in the /HMF8800/mqft/sql/ftelog_tables directory to your home directory.
   b) Edit this file. Change FTESRC to your existing schema (FTELOG) and FTEDEST to the new schema (FTEV8).
   c) Check the file to make sure all of the statements are within column 71.

      If the statements are not within column 71, split the line before column 71.
   d) If you have **DB2 RUNSTATS** jcl for the Managed File Transfer tables, create a new job specifying the new schema and tables.
8. Some tables have a generated ID to enforce a unique identifier for each row and you need to set these identifiers.

   The following tables have generated ID columns:

   - AUTH_EVENT
   - CALL
   - CALL_ARGUMENT
   - CALL_RESULT
   - FILE_SPACE_ENTRY
   - METADATA

- MONITOR_ACTION
- MONITOR_EXIT_RESULT
- MONITOR_METADATA
- SCHEDULE
- SCHEDULE_ACTION
- SCHEDULE_ITEM
- SCHEDULE_SPEC
- TRANSFER_CALLS
- TRANSFER_CD_NODE
- TRANSFER_CORRELATOR
- TRANSFER_EVENT
- TRANSFER_EXIT
- TRANSFER_ITEM
- TRANSFER_ITEM_ATTRIBUTES
- TRANSFER_STATS
- TRIGGER_CONDITION

To set the generated IDs of these tables to the correct value perform the following steps for each table:

a) Determine the maximum ID value in the existing data.

You can find this value by running this SQL statement:

```
SELECT MAX(ID) FROM schema_name.table_name
```

The value returned from this command is the maximum existing ID in the specified table.

b) Alter the table to set the ID generator to begin from a new value that is 1 higher than the value returned by the previous step.

You can set this value by running the following SQL statement:

```
ALTER TABLE schema_name.table_name ALTER COLUMN ID RESTART WITH value
```

9. Edit the database properties file to specify the new schema name:

a) If your Managed File Transfer configuration directory is `/u/userid/fteconfig` you can use the USS command **`find /u/userid/fteconfig -name databaselogger.properties`** to locate the properties file for the logger.

b) Edit this file and change `wmqfte.database.schema` to the new schema value.

10. Issue the following commands to convert the directory tree structure to V8.0.0 format before you attempt to use the logger:

a) **`fteMigrateConfigurationOptions`**

b) **`fteMigrateLogger`**

This copies the `databaselogger.properties` to `logger.properties`.

11. Edit existing Logger JCL to use the V8.0.0 Managed File Transfer libraries.

12. Start the logger.

Once the logger is working you can delete the V7 `FTELOG` tables.

z/OS LTS *Reverting a queue manager to a previous version on z/OS*

After migrating to IBM MQ for z/OS 9.1.0 Long Term Support (LTS) release, from either IBM MQ 8.0.0 or IBM MQ 9.0.0, you can backward migrate, or fallback, to the version you were using prior to migration.

Backward migration Program Temporary Fixes (PTFs) are available for both IBM MQ 8.0.0 and IBM MQ 9.0.0. Backwards migration is not supported for a Continuous Delivery (CD) release.

Before you begin

The APARs associated with the above mentioned backwards migration PTFs, document specific information relating to fallback.

V 9.1.0 From IBM MQ 9.1.0, queue managers can be backwards migrated to an earlier version if the CSQY039I message is output at queue manager start up. The message contains the version of IBM MQ code that backwards migration is supported to. If backwards migration is not supported message CSQY040I is output at queue manager start up instead.

If backwards migration is supported for the queue manager, you can go back to using your customization and startup procedure for the queue manager from that version if:

- Before migrating your queue manager to the latest version, with that version of target libraries, you applied all the migration and toleration PTFs to the queue manager on your previous version. The queue manager then started successfully with those PTFs at that previous version. This is a requirement before you can revert your queue manager to the original version.
- You saved the queue manager customization macros and JCL for running with the IBM MQ 8.0.0 or IBM MQ 9.0.0 target libraries. Though, you can use sample JCL to re-create the customization, if the originals are not available to you.

Important: If you were previously running at IBM MQ 8.0.0 or IBM MQ 9.0.0 with **OPMODE**(*COMPAT,nnn*) and you have enabled function at IBM MQ 9.1.0 which is protected by **OPMODE** in IBM MQ 8.0.0 or 9.0.0 you will need to recompile your ZPARMs to **OPMODE**(*NEWFUNC,800*) for IBM MQ 8.0.0 or **OPMODE**(*NEWFUNC,900*) for IBM MQ 9.0.0 before restarting your queue manager at those levels.

About this task

To restart the queue manager, so that it runs at the version where it was migrated from, just requires that you switch back to using the libraries for the previous version.

Note: It is not necessary to fall back the early code to the previous version, for this installation, when reverting your queue manager to an earlier version.

Early code refers to the IBM MQ load modules that must be loaded into the Link (LPA) for IBM MQ to act as a z/OS subsystem. When a command is issued to a queue manager, or when an application connects to a queue manager, the first action taken by the MQ system is to load the early code. The LPA must contain the IBM MQ early code modules from the latest version of IBM MQ running on the system. For example, if an IBM MQ 8 and IBM MQ 9 queue manager run on the same system, the early code for Version 9 must be loaded in the LPA.

See Early code for more information.

Procedure

1. Stop the listener, channel initiator, and queue manager.
2. Switch back to use the MSTR and CHINIT started procedure JCLs with the IBM MQ 8.0.0 or 9.0.0 libraries, as required.

   If data set aliases are being used for load libraries, switch the aliases to refer to the IBM MQ 8.0.0 or 9.0.0 libraries.

   For example, an alias named `MQM.MQP1.SCSQLOAD`, referring to `MQM.MQV910.SCSQLOAD`, needs to be changed to refer to `MQM.MQV900.SCSQLOAD`, or `MQM.MQV800.SCSQLOAD`, as required.
3. Revert to using the system parameter module (CSQZPARM) used with IBM MQ 8.0.0 or IBM MQ 9.0.0, prior to migration, and linking to the IBM MQ 8.0.0 or IBM MQ 9.0.0 code, as required.

   Verify the startup by starting the :

   a. Queue manager
   b. Channel initiator
   c. Listener

   separately. Check for, and resolve, any errors that occur during startup. Once all three components startup cleanly, you can combine the startup of the three components, if required.
4. Verify correct functioning of existing applications.

Results

Your queue manager will now be running at the version of code it was originally migrated from.

Note: For details about backwards migration from IBM MQ for z/OS 8.0.0 to IBM WebSphere MQ for z/OS 7.1.0, or from IBM MQ for z/OS 9.0.0 LTS release to IBM MQ for z/OS 8.0.0 or IBM WebSphere MQ for z/OS 7.1.0, refer to the instructions for IBM WebSphere MQ for z/OS 7.1.0, IBM MQ for z/OS 8.0.0, or IBM MQ for z/OS 9.0.0 in the IBM MQ product information in IBM Documentation.

z/OS *Post migration tasks*

Follow the steps to perform the tasks you need to carry out after migrating a single IBM MQ queue manager on z/OS,

About this task

After you have migrated an IBM MQ queue manager on z/OS you need to carry out the detailed steps in this topic, using the links within this overview.

1. Check the changes in behavior made by default configuration changes; see step “1” on page 217.
2. Modify the backup jobs to refer to the target version of IBM MQ libraries; see step “2” on page 217.
3. Update the ZPARM module if you have not already done so before starting the queue manager; see step “3” on page 217.
4. Configure Advanced Message Security; see step “4” on page 218.
5. Perform a full regression test; see step “5” on page 218.
6. Consider client application migration; see step “6” on page 218.
7. Exploit the new function; see step “7” on page 218.
8. Optionally, stop the mqweb server for previous versions; see step “8” on page 218.

Procedure

1. Check the changes in behavior made by default configuration changes.

   The default values of some properties might have been changed in the new version, which can lead to changes in behavior.
2. Modify backup and other administrative jobs, such as jobs to backup IBM MQ objects and channel authentication records, and MAKEDEF jobs.

   For example using CSQUTIL COMMAND MAKEDEF(..); see Using the COMMAND function of CSQUTIL to refer to the target version of IBM MQ libraries.
3. Update the system parameter (ZPARM) module if required.

   Note the following:

   - You should review changes to the ZPARM parameters between the version you have migrated from, and IBM MQ 9.1.
   - If you need to change the value of any parameters, you should generate a new ZPARM at this point. Do this by:

     a. Tailoring the ZPARM sample to use the new IBM MQ libraries
     b. Updating values for the parameters as necessary, and

   c. Recompiling, to generate the new ZPARM.

- You do not have to recompile the ZPARM, if you do not change the values of any parameters.

   V 9.1.0 For IBM MQ 9.1.0 you should ensure your ZPARM does not reference the **OPMODE** parameter as it is no longer supported. If **OPMODE** is specified, you will get a warning at assemble time.

4. If the queue manager is configured to use Advanced Message Security (AMS) perform the steps in the Post migration tasks for Advanced Message Security section of the Migrating Advanced Message Security topic.
5. Perform a full regression test.
6. Migrate client applications.

   Client applications can be considered any time throughout the migration phase.

   Clients are backwards and forwards compatible. It is advisable to migrate the client libraries to same, or later, level as the queue manager, so that the latest function is available.
7. Exploit new functions provided by the migrated queue manager.

   Your queue manager has been fully migrated to a new version level, so you can now take benefit of new capabilities.

   However, note that additional configuration might be required to enable selected new features.

   Review What's new in IBM MQ 9.1 and check which features best serve your business needs. Plan your action to develop new applications, or changing configurations, to enable those features.
8. If you created a new mqweb server for the latest version, you can stop the mqweb server for any previous versions when all queue managers on the z/OS system have been migrated to the latest version.

Results

You have completed the migration of a single IBM MQ for z/OS queue manager.

z/OS Queue sharing group migration

V 9.1.0 You can combine queue managers from different releases in a queue sharing group. Limit the time you manage a mixed group to only as long as it takes to migrate all the queue managers to the same command level. You cannot combine a queue manager at IBM MQ 9.1.0, or later, in the same queue sharing group as queue managers at, or capable of running at versions earlier than IBM MQ 8.0.0.

When you migrate queue managers in a queue sharing group, aim to migrate all the queue managers to the new version as soon as you can. Queue sharing groups can contain queue managers with a restricted mixture of versions. A mixture of queue managers in a queue sharing group is supported so that you can migrate and test the upgrade of each queue manager.

Migrate each queue manager, one at a time, leaving the queue sharing group running. At no stage is an outage of the entire queue sharing group required. Mixed groups are harder to administer, than if all the queue managers are at the same version.

Migrating each queue manager comprises the bulk of the work of migrating a queue sharing group. Approach migrating a queue sharing group as requiring some extra tasks that must be performed during the migration of each queue manager. These tasks are listed in “Migrating IBM MQ for z/OS - order of tasks” on page 198 as part of the procedure to migrate a single queue manager.

A good approach is to create a migration plan incorporating queue sharing group migration; see “Planning to migrate IBM MQ to a later version on z/OS” on page 200 for further information.

Notes: V 9.1.0

1. Unless stated otherwise, LTS and CD queue managers with the same version and release numbers can coexist in a queue sharing group without the need for a backwards migration PTF.

2. LTS version 9.1.0, and CD version 9.0.n queue managers can also coexist in a queue sharing group without the need for a backwards migration PTF.

Related reference

“MQSC commands in a mixed queue sharing group on z/OS” on page 104
Existing **MQSC** commands using new keywords and attribute values can be entered for routing to a migrated queue manager. You can enter the commands on any queue manager. Route the commands using **CMDSCOPE**. Commands with new keywords and attribute values, or new commands, routed to a previous version of queue manager, fail.

“Properties of objects in a mixed queue sharing group on z/OS” on page 104
Attributes that did not exist in earlier versions can be created and altered on queue managers of a later version in a mixed queue sharing group. The attributes are not available to queue managers in the group that are at an earlier level.

“Queue sharing group coexistence on z/OS” on page 103

z/OS Adding a new queue sharing group to an existing Db2 data sharing group in the latest version on z/OS

Follow these steps to add a new queue sharing group to an existing Db2 data sharing group in the latest version of the product. You must apply the migration and toleration PTFs to queue managers, in the previous version, in any of the queue sharing groups before adding a queue sharing group.

Before you begin

1. Review your Db2 data-sharing requirements. A single Db2 data-sharing group can be used to support multiple IBM MQ queue sharing groups.
2. You can add a new queue sharing group to a Db2 data-sharing group that already supports IBM MQ queue sharing groups containing queue managers for the previous version. You must ensure that the migration and toleration PTFs have been applied. The Db2 tables used by IBM MQ must be configured for the latest version queue managers.

Procedure

1. Set up the coupling facility.

   See Set up the coupling facility.
2. Customize and include the initialization input sample `*thlqual*.SCSQPROC(CSQ4INSS)` in the `CSQINP2` data set.

   See “Preparing to migrate a single IBM MQ for z/OS queue manager” on page 205 for more information.
3. Add the IBM MQ entries to the Db2 data-sharing group using the **CSQ5PQSG** program.

   See Add the IBM MQ entries to the Db2 data-sharing group.
4. Tailor the system parameter module to add Db2 data-sharing group and IBM MQ queue sharing group information.

   See “Preparing to migrate a single IBM MQ for z/OS queue manager” on page 205 for more information.

Migrating a queue manager cluster

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.

You must consider what the effect is of migrating some queue managers in a cluster, before all the queue managers are migrated.

Before you begin

Before starting the migration, check that no cluster-specific migration issues are identified for the migration you are intending to perform.

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 IBM MQ features
- Managing the migration of a cluster in the context of the wider IBM MQ network and the systems architecture of your organization.

About this task

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 running the latest IBM MQ major version (for LTS users) or CD version (for CD users). This ensures that the full repositories can store information from the rest of the cluster that arises from using new features.

Note: In exceptional circumstances, it might be necessary to upgrade some of your partial repositories before your full repositories.

While the product supports this configuration, in this situation be very careful to avoid use of any new clustering function on the partial repositories, until your full repositories have been upgraded, to avoid unexpected results.

Procedure

- For information about creating a migration plan for a queue manager cluster, see “Creating a migration plan for a queue manager cluster” on page 222.
- For information about creating a backout plan for the migration of a queue manager cluster, see “Creating a backout plan for queue manager cluster migration” on page 223.

- For information about how to migrate one queue manager in a queue manager cluster, see "Migrating one cluster queue manager" on page 223.

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, an IBM MQ 9.0 repository contains only IBM MQ 9.0 level attribute information. An IBM MQ 9.1 repository contains all IBM MQ 9.0 records, plus IBM MQ 9.1 records containing additional IBM MQ 9.1 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. An IBM MQ 9.0 queue manager receiving information about an IBM MQ 9.1 queue manager stores only IBM MQ 9.0 information. An IBM MQ 9.1 repository receiving an IBM MQ 9.0 record stores default values for attributes introduced in the later version. 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 an IBM MQ 9.0 full repository were to receive a record from an IBM MQ 9.1 partial repository, it would forward the IBM MQ 9.1 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 an IBM MQ 9.1 partial repository for new IBM MQ 9.1 attributes changing unexpectedly. The values might change from actual IBM MQ 9.1 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. See How long do the queue manager repositories retain information?

After migrating all full repositories to IBM MQ 9.1, 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.

Note: In exceptional circumstances, it might be necessary to upgrade some of your partial repositories before your full repositories.

While the product supports this configuration, in this situation be very careful to avoid use of any new clustering function on the partial repositories, until your full repositories have been upgraded, to avoid unexpected results.

Related concepts

How long do the queue manager repositories retain information?

Creating a migration plan for a queue manager cluster

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 82
IBM MQ supports running applications compiled and linked against previous versions of IBM MQ, with later levels of IBM MQ. Use the new version of the libraries to build the applications, once the queue managers have been upgraded.

Availability of cluster topic host queue managers

"How mixed version cluster repositories are updated" on page 221
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 migration" on page 80
After upgrading an installation, queue manager migration might be required. Migration takes place when you start a queue manager. You can remove an upgrade before you have started a queue manager. However, if you remove the upgrade after a queue manager has been started, the queue manager will not work.

"Queue sharing group migration" on page 218
V 9.1.0 You can combine queue managers from different releases in a queue sharing group. Limit the time you manage a mixed group to only as long as it takes to migrate all the queue managers to the same command level. You cannot combine a queue manager at IBM MQ 9.1.0, or later, in the same queue sharing group as queue managers at, or capable of running at versions earlier than IBM MQ 8.0.0.

Related tasks

"Migrating a queue manager in a high-availability configuration" on page 225
High-availability configurations of queue managers can increase the availability of IBM MQ applications. If a queue manager, or server fails, it is restarted automatically on another server. You can arrange for IBM

MQ MQI client applications to automatically reconnect to the queue manager. Server applications can be configured to start when the queue manager starts.

Creating a backout plan for queue manager cluster migration

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?

z/OS If the libraries of the earlier level of IBM MQ include the appropriate PTFs to be able to backward migrate, queue managers running on z/OS can be reverted to an earlier level by changing the load libraries.

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

“Queue manager migration” on page 80
After upgrading an installation, queue manager migration might be required. Migration takes place when you start a queue manager. You can remove an upgrade before you have started a queue manager. However, if you remove the upgrade after a queue manager has been started, the queue manager will not work.

Migrating one cluster queue manager

Follow these steps to migrate a single queue manager in a cluster, starting with a queue manager in your test system. Base these steps on your cluster migration plan.

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(*)
   ```

   c) Issue the command to view cluster topics.

   ```
   DISPLAY TCLUSTER(*)
   ```

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)
   ```

   c) Issue the command on the full repositories to display the cluster topics for this queue manager.

   ```
   DISPLAY TCLUSTER(*) 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 a queue manager on Windows" on page 116.

   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 IBM 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(*)
   ```

   c) Issue the command to view cluster topics and check the output against the data saved before migration.

```
DISPLAY TCLUSTER(*)
```

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, its cluster queues, and its cluster topics.
   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)
   ```

   c) Issue the command on the full repositories and check the output against the data saved before migration.

   ```
   DISPLAY TCLUSTER(*) 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 by issuing the following command:

    ```
    RESUME QMGR CLUSTER(cluster name)
    ```

12. Closely monitor the queue manager and applications in the cluster for a while.

What to do next

When you have completed the migration of one queue manager in a cluster, on your test system, complete the migration of the other queue managers in each cluster on the test system.

When you have competed the migration of all of the queue managers on your test system, migrate each of the queue managers on your production system.

Related concepts

"Queue manager migration" on page 80
After upgrading an installation, queue manager migration might be required. Migration takes place when you start a queue manager. You can remove an upgrade before you have started a queue manager. However, if you remove the upgrade after a queue manager has been started, the queue manager will not work.

Related reference

DISPLAY CLUSQMGR
DISPLAY QUEUE
RESUME QMGR
SUSPEND QMGR

Migrating a queue manager in a high-availability configuration

High-availability configurations of queue managers can increase the availability of IBM MQ applications. If a queue manager, or server fails, it is restarted automatically on another server. You can arrange for IBM

MQ MQI client applications to automatically reconnect to the queue manager. Server applications can be configured to start when the queue manager starts.

About this task

Multi For IBM MQ for Multiplatforms, high-availability configurations can be implemented 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.

Linux For Linux platforms, you can implement high availability by using replicated data queue managers (RDQMs). For migrating RDQMs, see "Migrating replicated data queue managers" on page 228.

z/OS For IBM MQ for z/OS there are several alternative techniques to increase queue manager availability; see Availability on z/OS. Migration considerations on z/OS depend on the availability techniques that are employed, and are not described in this topic. The term high-availability configuration refers only to queue managers in configurations on platforms other than z/OS.

MQ Appliance Another solution is to configure a high availability group on a pair of IBM MQ Appliances. See the Appliance documentation for details of migrating HA queue managers.

The overall principles involved in queue manager migration in a high availability configuration based on multi-instance queue managers or on a high-availability cluster are the same. 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.

Procedure

- To migrate a multi-instance queue manager, see "Migrating a multi-instance queue manager" on page 226.
- To migrate a high availability cluster queue manager, see "Migrating a high-availability cluster queue manager" on page 227.

Related tasks

"Migrating an MSCS configuration on Windows" on page 142
Migrate queue managers in a Microsoft Cluster Service (MSCS) configuration one node at a time, following these instructions.

Migrating a multi-instance queue manager

Follow the steps listed to migrate a queue manager in a multi-instance queue manager configuration.

Before you begin

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.

Procedure

Base your migration procedure on the following steps:

1. Before you start the migration process, create a different queue manager on a server, on which you have installed the upgrade.
2. 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 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.

   Ensure that you have no system management procedure running that restarts the instance automatically.
5. If you do not have a pool of servers, upgrade IBM MQ on the server that was running the standby instance
6. Decide whether downtime or recoverability is more important in the migration.
7. Optional: Follow this procedure 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.
8. Optional: Follow this procedure if availability is more important. You do not need to 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.
9. Upgrade the IBM MQ code on the server that was the active queue manager instance.
10. Start the server as the standby instance if you have not already started a standby.

“Migrating a queue manager in a high-availability configuration” on page 225
High-availability configurations of queue managers can increase the availability of IBM MQ applications. If a queue manager, or server fails, it is restarted automatically on another server. You can arrange for IBM MQ MQI client applications to automatically reconnect to the queue manager. Server applications can be configured to start when the queue manager starts.

“Migrating a high-availability cluster queue manager” on page 227
Follow the steps listed to migrate a queue manager in a high-availability queue manager configuration.

Migrating a high-availability cluster queue manager

Follow the steps listed to migrate a queue manager in a high-availability queue manager configuration.

Before you begin

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.

Procedure

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.

2. Test the upgrade by performing whatever verification checks that your enterprise requires.
3. Form two cluster pairs if you have four servers available.

   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.
4. Remove a passive server from the cluster.

   Ensure that the cluster cannot automatically restart the server. The server is made inactive.
5. Create a second location for the upgraded code, if a high-availability cluster is using a common location for IBM MQ code.
6. Install, or upgrade, IBM MQ code using the server that is not now running the queue manager.
7. Verify the upgrade by creating a different queue manager on the server, and performing whatever verification checks that your organization requires.
8. If more than half the servers remain in the cluster, remove a server, upgrade IBM MQ, and verify the upgrade.

   Each server is made inactive as part of the process. Continue until half the servers are upgraded.
9. If your active server is part of a remaining cluster, deactivate the passive servers so that the cluster cannot reactivate them automatically.
10. Decide whether downtime or recoverability is more important in the migration.
11. Optional: Follow this procedure if recoverability is more important:
    a) Stop the queue manager and remove the server from the cluster.
    b) Back up the queue manager.
12. Optional: Follow this procedure if downtime is more important:
    a) Add the migrated servers back into the cluster, as passive servers.
    b) 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, and restarts it on one of the passive servers.
13. Upgrade any remaining high-availability servers, and add them back into the cluster.

“Migrating a queue manager in a high-availability configuration” on page 225
High-availability configurations of queue managers can increase the availability of IBM MQ applications. If a queue manager, or server fails, it is restarted automatically on another server. You can arrange for IBM MQ MQI client applications to automatically reconnect to the queue manager. Server applications can be configured to start when the queue manager starts.

“Migrating a multi-instance queue manager” on page 226
Follow the steps listed to migrate a queue manager in a multi-instance queue manager configuration.

MQ Adv. Linux V 9.1.0 Migrating replicated data queue managers

When you need to migrate replicated data queue managers (RDQMs), you must upgrade all nodes in a sequence. Do not try to operate with the nodes at different levels. This guidance is appropriate for moving between major releases, or CD releases, but not for applying (fix pack) maintenance.

About this task

The upgrade sequence for HA RDQM configurations consists of suspending a node, uninstalling IBM MQ and RDQM support, installing the newer version of IBM MQ and RDQM support, then resuming the node. You then move on and repeat this sequence on the next node. Following this sequence ensures that your queue managers continue to run on one of the nodes in the HA group while the migration is in progress.

V 9.1.0 The upgrade sequence for DR RDQM configurations consists of upgrading the recovery node, running the DR queue managers on the newly upgraded recovery node, upgrading the primary node, switching the DR queue managers back to running on the primary node.

V 9.1.5 The upgrade sequence for DR/HA RDQM configurations consists of upgrading the HA group on the recovery site, performing a managed failover from the main site to the recovery site, and then upgrading the HA group on the main site before failing the queue managers back to the main site.

Scripts are provided that you run to uninstall and install IBM MQ and RDQM.

The upgrade location must be the same as the original installation location, and all the nodes must use the same location.

You can back up a replicated data queue manager before migration.

Related tasks

Installing RDQM (replicated data queue managers)

V 9.1.0 Migrating HA RDQMs

Follow this sequence of steps to upgrade all the RDQM nodes in an HA group and so migrate the replicated data queue managers (RDQMs).

About this task

You should upgrade all the nodes in an HA group in the same sequence to avoid operating with the nodes in the group at different levels.

If you have configured your HA group such that one node acts as a primary for all RDQMs, with the other two nodes as secondaries, you should upgrade the secondary nodes first and leave the primary node until last.

The sequence in which you upgrade, and the nodes that are marked as preferred and second preferred locations for RDQMs, affect where the RDQMs fail over to as you upgrade. During the migration sequence, while nodes are running different levels, the options for failing over are limited. An RDQM running on a lower level node can fail over to a higher level node but, once a queue manager has been started at the new level, it cannot fail over to a lower level node. You should choose an upgrade sequence and use preferred and second preferred locations settings to keep queue managers running on the lower level nodes for as long as possible. You should make changes to preferred and second preferred location settings before you suspend nodes, to ensure that the changes are effective immediately.

V 9.1.0 If you are also running DR RDQMs on any of the nodes, you should deal with these queue managers at the same time by following the instructions in "Migrating DR RDQMs" on page 230.

Procedure

1. To upgrade the first node in the HA group:
   a) Suspend the node from the HA group by running the following command:

      ```
      rdqmadm -s
      ```

      Any RDQMs currently running on the node move to another node in the HA group (their second preference, if one is defined for that RDQM).
   b) Run the uninstall script to uninstall IBM MQ and RDQM.

      ```
      MQ_INSTALLATION_PATH/Advanced/RDQM/uninstallRDQMsupport
      ```

      The script can take some time to run, without indication that it is still running.
   c) Accept the license for the new version by running the **mqlicense** script.
   d) Run the installation script from the `/Advanced/RDQM` directory of your installation media, accepting the license when prompted:

      ```
      installRDQMsupport
      ```

      The `installRDQMSupport` script installs only the default MQ components, for example:

- MQSeriesClient.x86_64
- MQSeriesRDQM.x86_64
- MQSeriesRuntime.x86_64
- MQSeriesSamples.x86_64
- MQSeriesServer.x86_64

If you require any other component (for example, AMQP, MFT, AMS and so on), then these components must be installed manually.

e) If required, set this installation as the primary IBM MQ installation, using the **setmqinst** command. See setmqinst (set IBM MQ installation).

f) Resume the node in the HA group by entering the following command:

```
rdqmadm -r
```

Any RDQMs that have this node as their preferred location will resume running on this node.

2. Repeat the steps for the second node in the HA group. (Use the same path as you did on the first node.)
3. Repeat the steps for the third node in the HA group. (Use the same path as you did on the first node.)

What to do next

Verify that the installation on all three nodes matches by checking the RDQM configuration in the `qm.ini` files.

Related reference
rdqmadm (administer replicated data queue manager cluster)

Linux V 9.1.0 Migrating DR RDQMs

Follow this sequence of steps to upgrade the primary and recovery nodes in a disaster recover replicated data queue manager (DR RDQM) configuration.

About this task

The suggested sequence for upgrading your nodes is to upgrade your recovery node, then run your DR queue managers there while you then upgrade your primary node. When both nodes are upgraded you can restore the original primary and recovery roles.

If you do not require to run your DR queue managers during the upgrade procedure, then you can omit the steps for failing over to the recovery node. You can just stop your DR queue managers and restart them after you have upgraded both nodes.

If you are also running HA RDQMs on either of the nodes, you should deal with these queue managers at the same time by following the instructions in “Migrating HA RDQMs” on page 229.

Procedure

- To upgrade while continuing to run your DR queue managers:

  a) Upgrade your recovery node:

  a. Run the uninstall script to uninstall IBM MQ and RDQM.

  ```
  MQ_INSTALLATION_PATH/Advanced/RDQM/uninstallRDQMsupport
  ```

  b. Accept the license for the new version by running the **mqlicense** script.

  c. Run the installation script from the `/Advanced/RDQM` directory of your installation media:

  ```
  installRDQMsupport
  ```

    d. If required, set this installation as the primary IBM MQ installation, using the **setmqinst** command. See setmqinst (set IBM MQ installation).

  b) Turn the DR queue managers into secondary instances on your primary node by entering the following commands for each queue manager:

    a. Stop the queue manager:

    ```
    endmqm -r QMname
    ```

    b. Make the queue manager into a secondary instance:

    ```
    rdqmdr -m QMname -s
    ```

  c) Run the queue managers on the recovery node by completing the following steps:

    a. Make each queue manager into a primary instance:

    ```
    rdqmdr -m QMname -p
    ```

    b. Start each queue manager:

    ```
    strmqm qmname
    ```

  d) Upgrade the primary node:

    a. Run the uninstall script to uninstall IBM MQ and RDQM.

    ```
    MQ_INSTALLATION_PATH/Advanced/RDQM/uninstallRDQMsupport
    ```

    b. Accept the license for the new version by running the **mqlicense** script.

    c. Run the installation script from the `/Advanced/RDQM` directory of your installation media:

    ```
    installRDQMsupport
    ```

    d. If required, set this installation as the primary IBM MQ installation, using the **setmqinst** command. See setmqinst (set IBM MQ installation).

  e) On the recovery node, make the queue managers into secondary instances once more:

  ```
  rdqmdr -m QMname -s
  ```

  f) On the primary node, make the queue managers into primary instances and start them:

  ```
  rdqmdr -m QMname -p
  strmqm qmname
  ```

- To upgrade while not running your DR queue managers:

  a) Upgrade your recovery node:

    a. Run the uninstall script to uninstall IBM MQ and RDQM.

    ```
    MQ_INSTALLATION_PATH/Advanced/RDQM/uninstallRDQMsupport
    ```

    b. Accept the license for the new version by running the **mqlicense** script.

    c. Run the installation script from the `/Advanced/RDQM` directory of your installation media:

    ```
    installRDQMsupport
    ```

    d. If required, set this installation as the primary IBM MQ installation, using the **setmqinst** command. See setmqinst (set IBM MQ installation).

  b) Stop each queue manager on your primary node:

  ```
  endmqm QMname
  ```

  c) Upgrade the primary node:

a. Run the uninstall script to uninstall IBM MQ and RDQM.

```
MQ_INSTALLATION_PATH/Advanced/RDQM/uninstallRDQMsupport
```

b. Accept the license for the new version by running the **mqlicense** script.

c. Run the installation script from the /Advanced/RDQM directory of your installation media:

```
installRDQMsupport
```

d. If required, set this installation as the primary IBM MQ installation, using the **setmqinst** command. See setmqinst (set IBM MQ installation).

d) Start the queue managers on your primary node:

```
strmqm qmname
```

V 9.1.5 Migrating DR/HA RDQMs

Follow these steps to upgrade all the RDQM nodes in both HA groups in a DR/HA configuration, and so migrate the replicated data queue managers (RDQMs).

About this task

The suggested sequence for upgrading your nodes is to upgrade the HA group at your recovery site, then run your DR/HA queue managers there while you upgrade the HA group at your main site. When both HA groups are upgraded you can restore the original main and recovery roles.

Procedure

1. Upgrade the HA group at your recovery site, following the procedure described in “Migrating HA RDQMs” on page 229.
2. Perform a managed failover of the DR/HA RDQMs running on your main site so they run on a node in the HA group of your newly upgraded recovery site by following the procedure described in “Migrating DR RDQMs” on page 230.
3. Upgrade the HA group at your main site.
4. Fail the DR/HA RDQMs running on the HA group of the recovery site back to the main site.

Related tasks

“Migrating HA RDQMs” on page 229
Follow this sequence of steps to upgrade all the RDQM nodes in an HA group and so migrate the replicated data queue managers (RDQMs).

“Migrating DR RDQMs” on page 230
Follow this sequence of steps to upgrade the primary and recovery nodes in a disaster recover replicated data queue manager (DR RDQM) configuration.

Moving a queue manager to a different operating system

Follow these instructions to move a queue manager from one operating system to another. Note that this is **not** a migration of a queue manager.

About this task

You move a queue manager by re-creating it on the target system. The procedure re-creates the configuration of the queue manager, it does not attempt to re-create the current state of the queue manager by, for example, unloading and reloading queues.

Procedure

1. Log in to the source system as a user in the IBM MQ administrators (mqm) group.
2. Save the configuration information of the queue manager that you want to move by typing the following command:

   ```
   dmpmqcfg -a -m QM_name > QM_file
   ```

   Where:

   - *QM_name* is the name of the queue manager that you want to move.
   - *QM_file* is the name and path of a local file on the source system that the configuration information is written to.

   See **dmpmqcfg** for more information.
3. If the queue manager is part of a distributed configuration, quiesce the queue manager. Ensure that there are no messages in flight, then stop the queue manager.
4. If you are moving from one version of the product to another, migrate the queue manager on your current operating system, to the latest version.

   If your current operating system is:

   - Windows Windows, see "Migrating a queue manager to a later version on Windows" on page 117
   - Linux UNIX UNIX or Linux, see Migrating a queue manager on UNIX and Linux

   You should ensure that your existing applications still work.
5. Create an empty queue manager on your new operating system, using **crtmqm**.
6. Copy your object definitions across to the new queue manager you have just created, using **dmpmqcfg**.

   You should take great care when you copy the object definitions, because some manual modification of the definitions might be needed:

   - Various attributes need to be checked and might need to be changed. These include:
     - Ip addresses and ports in channels, listeners, and other objects
     - Security information such as user IDs
     - **startcmd** in services
     - Various other attributes.
   - Durable subscribers that are not administered might lose messages.
   - Other queue managers might also need to be changed so that their channels connect to the moved queue manager.

   Once the definitions have been copied across, the application messages from the queue manager, on your original operating system, need to be copied to the queue manager on your new operating system, using an application that moves messages. Then check your applications still work.

ULW V 9.1.0 Migrating logs on UNIX, Linux, and Windows

From IBM MQ 9.1.0 you can migrate a circular log to a linear log, or from a linear log to a circular log.

Before you begin

Decide whether you want to use linear or circular logging by reviewing Types of logging.

Procedure

- To migrate your queue manager log from being linear to circular, see "Migrating the log of your queue manager from linear to circular" on page 234.

- To migrate your queue manager log from being circular to linear, see “Migrating the log of your queue manager from circular to linear” on page 235.

Related concepts

“Migrating logs to an Advanced Format disk on Windows” on page 145
An Advanced Format disk is one that has 4096 bytes per sector. The following is applicable only to the Windows platform as Advanced Format disks can be used on other platforms, without carrying out a migration procedure.

ULW V 9.1.0 Migrating the log of your queue manager from linear to circular

Follow this sequence of steps to migrate your queue manager log from being linear to circular.

Before you begin

Before you migrate, take a backup of your queue manager.

Decide whether you want to migrate your log in place, or migrate the log to a new location by specifying the **-ld** option on the **migmqlog** command. If you are moving your queue manager from an old disk to a new Advanced Format disk, it might be convenient to use the **-ld** option.

Make sure that you have enough space to migrate your log, as the log might grow during migration.

Note the following:

- You do not need to migrate the log of your queue manager in order to use IBM MQ 9.1.0.
- **migmqlog** might take some minutes to complete if your log is very large. However, the command outputs progress messages from time to time.
- If, for any reason (for instance due to a power outage) **migmqlog** stops before it has completed the process, rerun the same **migmqlog** command on the partially migrated logs to complete the migration.
- If you specified the **-ld** option, **migmqlog** updates the log path in the `qm.ini` file for you, so when you start your queue manager it will use the migrated log.
- Do not pass a relative path when using the **-ld** option; only use an absolute path.
- **migmqlog** does not update any queue or queue manager objects.

Procedure

1. Login as a member of the `mqm` group.
2. If you have already not done so, stop the queue manager.
3. If you have not already done so, take a backup of your queue manager.
4. Run the following command:

   ```
   migmqlog -m QMgrName -lc
   ```

   See **migmqlog** for more information.

Related tasks

“Migrating the log of your queue manager from circular to linear” on page 235

Follow this sequence of steps to migrate your queue manager log from being circular to linear.

ULW V 9.1.0 Migrating the log of your queue manager from circular to linear

Follow this sequence of steps to migrate your queue manager log from being circular to linear.

Before you begin

Before you migrate, take a backup of your queue manager.

Decide whether you want to migrate your log in place, or migrate the log to a new location by specifying the **-ld** option on the **migmqlog** command. If you are moving your queue manager from an old disk to a new Advanced Format disk, it might be convenient to use the **-ld** option.

Make sure that you have enough space to migrate your log, as the log might grow during migration.

Note the following:

- You do not need to migrate the log of your queue manager in order to use IBM MQ 9.1.0.
- **migmqlog** might take some minutes to complete if your log is very large. However, the command outputs progress messages from time to time.
- If, for any reason (for instance due to a power outage) **migmqlog** stops before it has completed the process, rerun the same **migmqlog** command on the partially migrated logs to complete the migration.
- If you specified the **-ld** option, **migmqlog** updates the log path in the `qm.ini` file for you, so when you start your queue manager it will use the migrated log.
- Do not pass a relative path when using the **-ld** option; only use an absolute path.
- **migmqlog** does not update any queue or queue manager objects.

About this task

⚠ **Attention:** After you have migrated, a media image will not have been recorded when the queue manager starts. Plan how you intend to record media images, either automatically by setting the attributes:

- IMGSCHED
- IMGINTVL
- IMGLOGLN
- IMGRCOVO
- IMGRCOVQ

in ALTER QMGR, or manually by periodically running **rcdmqimg**.

Procedure

1. Login as a member of the `mqm` group.
2. If you have already not done so, stop the queue manager.
3. If you have not already done so, take a backup of your queue manager.
4. Run the following command:

   ```
   migmqlog -m QMgrName -ll
   ```

   See **migmqlog** for more information.
5. Start the queue manager, and set the appropriate image recovery and queue attributes for your environment.
6. Consider when to record manual images for objects that are recoverable.

Related tasks

"Migrating the log of your queue manager from linear to circular" on page 234
Follow this sequence of steps to migrate your queue manager log from being linear to circular.

Internet Protocol Version 6 (IPv6) migration

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

This section lists the IBM MQ platforms that support IPv6.

IPv6 is supported on the following IBM MQ platforms:

- AIX AIX
- IBM i IBM i
- Linux Linux
- Solaris Solaris
- Windows Windows
- z/OS z/OS

Key points in migrating to IPv6 and using IBM MQ

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

- IBM 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 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 MQ on does not support IPv6, IBM 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 host name that resolves to an IPv4 address only.
- If there are multiple domain name servers in an IBM MQ network, each host name 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 MQ on an IPv6 network.

The IPv6 protocol can only be used by IBM WebSphere MQ 6.0 or later. In order to make use of the IPv6 protocol, IBM 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 host name resolves ambiguously to both an IPv4 address and an IPv6 address.

To migrate a queue manager to use the IPv6 protocol:

1. Configure dual IPv4 and IPv6 protocols on the system where the queue manager to be migrated resides.
2. Install IBM MQ.
3. Add an entry to the DNS to resolve the host name of the system that is to be migrated, to both an IPv4 address and an IPv6 address.
4. Set the IPADDRV parameter to IPv6 (or set the LOCLADDR parameter to resolve to an IPv6 address).

⚠ **CAUTION:** Not all IPv6 software can interpret an IPv4 mapped IPv6 address. If the combination of CONNAME and LOCLADDR results in an IPv4 mapped IPv6 address, ensure that the system hosting the target queue manager is capable of handling this.

Using mapped addresses can require protocol translators in the IP network.

Migration scenarios (non-cluster topology)

It is possible to come up with a number of different interconnection possibilities, and the following sections aim to help you understand how IBM 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 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 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 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 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 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 an IBM 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

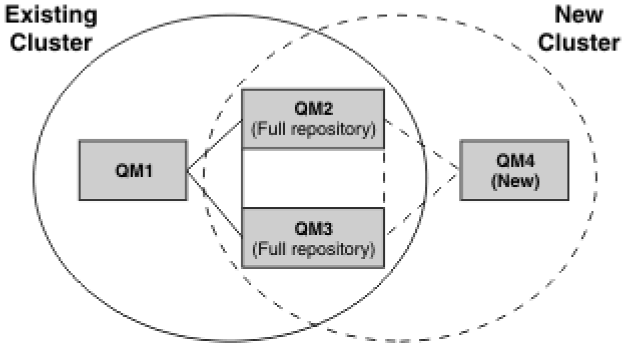
A cluster from an earlier version of the product is installed on IPv4 only capable systems and you need to connect an IPv6 only capable system into the cluster. Your network does not support adding both

IPv6 and IPv4 addresses using the same host name or you are using IP addresses rather than DNS names in the cluster channel CONNAMEs.

The problem here is likely to be that all of the systems cannot be switched to IPv6 simultaneously and some at least must remain only IPv4 capable. The systems that your new IPv6 only system communicates with must be IPv4 and IPv6 capable. We do not recommend simply adding a new set of IPv6 channels into the cluster for the IPv6 system to use, as the IPv4 system would also try to use them, resulting in communication errors.

The recommended approach is:

- Define a new cluster which contains the IPv6 only capable system or systems with new IPv6 addresses and channel definitions. The existing cluster remains, and contains the IPv4 only system definitions. The image below gives a pictorial representation of this. QM1, QM2, and QM3 represent the original IPv4 cluster. QM2, QM3, and QM4 represent the new cluster created to allow the IPv6 only capable system (QM4) to connect into your configuration.
- If you are using DNS names, you can give each of the systems separate DNS names for IPv4 and IPv6 (for example system1_IPv4.ibm.com and system1_IPv6.ibm.com).
- Define a new CLUSRCVR channel and any corresponding CLUSSDR channels using the new IPv6 names or IP addresses on each system in the new cluster. In this way the systems with only IPv4 or IPv6 capability do not see channels which they are not able to use and no communications error will result.



Note: There are both IPv4 and IPv6 definitions connecting the full repositories so that definitions for both new and existing cluster definitions are replicated between them. Also be aware that the queue managers QM1 and QM4 cannot communicate directly because they do not share a common network. They could communicate indirectly, for example by using ALIAS queues defined in the queue managers QM2 and QM3. In the configuration shown above you would need to pay attention to the ordering of application messages flowing between QM2 and QM3 because multiple routes exist, if this is relevant you could use BIND_OPEN to fix the route.

Abbreviated migration scenarios

This section gives some abbreviated scenarios for when you are thinking of installing clusters on IBM 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 16. Effects of CONNAME and LOCLADDR settings

Stack Type	CONNAME setting	LOCLADDR setting	Connection result
IPv4 only stack	IPv4 address		Channel binds to IPv4 stack

Table 16. Effects of CONNAME and LOCLADDR settings (continued)

Stack Type	CONNAME setting	LOCLADDR setting	Connection result
	IPv6 address		Channel fails to resolve CONNAME
	Host name resolves to both IPv4 and IPv6 addresses		Channel binds to IPv4 stack
	IPv4 address	IPv4 address	Channel binds to IPv4 stack
	IPv6 address	IPv4 address	Channel fails to resolve CONNAME
	Host name resolves to both IPv4 and IPv6 addresses	IPv4 address	Channel binds to IPv4 stack
	Any address	IPv6 address	Channel fails to resolve LOCLADDR
	IPv4 address	Host name resolves to both IPv4 and IPv6 addresses	Channel binds to IPv4 stack
	IPv6 address	Host name resolves to both IPv4 and IPv6 addresses	Channel fails to resolve CONNAME
	Host name resolves to both IPv4 and IPv6 addresses	Host name resolves to both IPv4 and IPv6 addresses	Channel binds to IPv4 stack
Dual IPv4 and IPv6 stack	IPv4 address		Channel binds to IPv4 stack
	IPv6 address		Channel binds to IPv6 stack
	Host name resolves to both IPv4 and IPv6 addresses		Channel binds to stack determined by IPADDRV
	IPv4 address	IPv4 address	Channel binds to IPv4 stack
	IPv6 address	IPv4 address	Channel fails to resolve CONNAME
	Host name resolves to both IPv4 and IPv6 addresses	IPv4 address	Channel binds to IPv4 stack
	IPv4 address	IPv6 address	Maps an IPv4 CONNAME to an IPv4 mapped IPv6 address. IPv6 implementations that do not support IPv4 mapped IPv6 addressing fail to resolve CONNAME
	IPv6 address	IPv6 address	Channel binds to IPv6 stack
	Host name resolves to both IPv4 and IPv6 addresses	IPv6 address	Channel binds to IPv6 stack

Table 16. Effects of CONNAME and LOCLADDR settings (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
IPv6 only stack	IPv4 address		Maps an IPv4 CONNAME to an IPv4 mapped IPv6 address. IPv6 implementations that do not support IPv4 mapped IPv6 addressing fail to resolve CONNAME
	IPv6 address		Channel binds to IPv6 stack
	Host name resolves to both IPv4 and IPv6 addresses		Channel binds to IPv6 stack
	Any address	IPv4 address	Channel fails to resolve LOCLADDR
	IPv4 address	IPv6 address	Maps an IPv4 CONNAME to an IPv4 mapped IPv6 address. IPv6 implementations that do not support IPv4 mapped IPv6 addressing fail to resolve CONNAME
	IPv6 address	IPv6 address	Channel binds to IPv6 stack
	Host name resolves to both IPv4 and IPv6 addresses	IPv6 address	Channel binds to IPv6 stack
	IPv4 address	Host name resolves to both IPv4 and IPv6 addresses	Maps an IPv4 CONNAME to an IPv4 mapped IPv6 address. IPv6 implementations that do not support IPv4 mapped IPv6 addressing fail to resolve CONNAME
	IPv6 address	Host name resolves to both IPv4 and IPv6 addresses	Channel binds to IPv6 stack
	Host name resolves to both IPv4 and IPv6 addresses	Host name resolves to both IPv4 and IPv6 addresses	Channel binds to IPv6 stack

Abbreviated scenarios: System configurations

Table 18 on page 243 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 18 on page 243 to give the configuration of the systems involved in trying to establish communication. For example:

- 71 + IPv6: Represents a queue manager from an earlier version of the product on a system with a TCP/IP version 6 stack
- 8 + 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 17. Abbreviations used in system configurations

Abbreviation	Meaning
71	queue manager from an earlier version of the product
8	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 host name of system holding the responding queue manager
IPv6DNS	DNS returns an IPv6 address only for host name of system holding the responding queue manager
DualDNS	DNS returns an IPv4 and IPv6 address for host name of system holding the responding queue manager
LOCLADDR4	The LOCLADDR parameter is set to IPv4 addressing
LOCLADDR6	The LOCLADDR parameter is set to IPv6 addressing
IPADDR4	IPADDRV is set to IPv4 addressing
IPADDR6	IPADDRV is set to IPv6 addressing

Table 18. System configurations

Originating queue manager		Responding queue manager			Result
Queue manager and Stack	**LOCLADDR**	**IPADDRV**	**Queue Manager and Stack**	**DNS Return**	
v71 + IPv6	Any	Not applicable			IP Error
v71 + IPv4 or v71 + Dual	Both LOCLADDR4 & LOCLADDR6	Not applicable	v71 + IPv4 or v71 + Dual	IPv4DNS or DualDNS	IPv4 connection can be established

Table 18. System configurations (continued)

Originating queue manager		**Responding queue manager**			**Result**
Queue manager and Stack	**LOCLADDR**	**IPADDRV**	**Queue Manager and Stack**	**DNS Return**	
v71 + IPv4 or v71 + Dual	Blank or LOCLADDR4	Not applicable	v71 + IPv4 or v71 + Dual	IPv4DNS or DualDNS	IPv4 connection can be established
v71 + IPv4 or v71 + Dual	Blank or LOCLADDR4	Not applicable	v71 + Dual	IPv6DNS	Unable to resolve CONNAME
v71 + IPv4 or v71 + Dual	Blank or LOCLADDR4	Not applicable	v71 + Dual or v8 + Dual v8 + IPv4	IPv4DNS or DualDNS	IPv4 connection can be established
v71 + IPv4 or v71 + Dual	LOCLADDR6	Not applicable			IP Error
v71 + IPv4 or v71 + Dual	Blank or LOCLADDR4 or both LOCLADDR4 & LOCLADDR6	Not applicable	v8 + IPv6	IPv6DNS	Unable to resolve CONNAME
v8 + IPv4	Blank or LOCLADDR4	Not specified	v71 + IPv4 or v71 + Dual or v8 + IPv4	IPv4DNS or DualDNS	IPv4 connection can be established
v8 + IPv4	LOCADD6	Not specified			Unable to resolve LOCLADDR
v8 + IPv4	Blank or LOCLADDR4	Not specified	v8 + IPv6	IPv6DNS	Unable to resolve CONNAME
v8 + IPv6	Blank or LOCLADDR6	Not specified	v71 + Dual	DualDNS	Attempts to start IPv6 channel and fails as there will be no IPv6 listener available
v8 + IPv6	Blank or LOCLADDR6	Not specified	v71 + IPv4	IPv4DNS	Attempts to start IPv6 channel and fails as there will be no IPv6 listener available
v8 + IPv6 or v8 + Dual	LOCLADDR6	Blank or IPADDR6	v8 + IPv6 or v8 + Dual	IPv6DNS or DualDNS	IPv6 connection can be established

Table 18. System configurations (continued)

Originating queue manager		Responding queue manager			Result
Queue manager and Stack	**LOCLADDR**	**IPADDRV**	**Queue Manager and Stack**	**DNS Return**	
v8 + Dual	LOCLADDR6	IPADDR4	v8 + Dual	IPv4DNS or DualDNS	IPv6 connection can be established where mapped addressing can be used
v8 + Dual	Blank or LOCLADDR4	IPADDR4	v71 + Dual	IPv4DNS or DualDNS	IPv4 connection can be established
v8 + Dual	Both LOCLADDR4 & LOCLADDR6	Blank or IPADDR4	v71 + Dual	IPv4DNS or DualDNS	IPv4 connection can be established
v8 + Dual	LOCLADDR4	IPADDR4			Unable to resolve LOCLADDR
v8 + Dual	LOCLADDR6 or both LOCLADDR4 & LOCLADDR6	Blank or IPADDR6	v8 + IPv6 or v8 + Dual	IPv6DNS or DualDNS	IPv6 connection can be established

V 9.1.1 Migrating existing security configurations to use the ANY_TLS12_OR_HIGHER CipherSpec

Migrating to the ANY_TLS12_OR_HIGHER CipherSpec means that your enterprise can adapt to cipher additions and deprecations without needing to make further invasive configuration changes in the future.

In general terms, the migration step to use the ANY_TLS12_OR_HIGHER CipherSpec is no different from the process you use to change any CipherSpec. That is, change the value of the CipherSpec for the channel definition at each end, and then restart the channels for the change to take effect.

The procedure described in the preceding text can be particularly challenging in clustering environments. Typically, you need to update manually defined channel definitions to a full repository one at a time.

To simplify migration, you make the change to specify ANY_TLS12_OR_HIGHER on a channel definition pairing on the responding message channel agent (that is SVRCONN, RCVR, and so on) first. This approach allows any channels that were previously designed to use a specific TLS 1.2 cipher to continue to work using that specific CipherSpec.

If you plan to change an existing cluster to use ANY_TLS12_OR_HIGHER, you first need to ensure that all members of the cluster are at IBM MQ 9.1.4, or higher, to understand the new CipherSpec value. The procedure for migration is the same as migrating from plaintext to SSL or TLS. See Upgrading clustered queue managers and channels to SSL/TLS for more information.

Once both initiating and responding channel definitions have ANY_TLS12_OR_HIGHER set as the CipherSpec, the negotiation of the cipher to use varies, based on the availability of different algorithms based on platform and maintenance levels.

Note, that although no assurance can be made on the exact CipherSpec that is chosen, the channel will only use the TLS 1.2 and higher protocols, and the list of common ciphers considers FIPS, SUITEB, and weak CipherSpec deprecations and re-enablement on both peers.

⚠ **Attention:** ANY_TLS12_OR_HIGHER does not guarantee that a specific CipherSpec will be used on a running channel, only that the negotiated CipherSpec is enabled and acceptable to IBM MQ on both ends of the channel. To request that a specific CipherSpec is used by a channel, you must specify that specific value on both ends of the channel.

If you add support for a new CipherSpec to the IBM MQ installations on the initiating and responding ends of the channel, the ANY_TLS12_OR_HIGHER CipherSpec will allow this new CipherSpec to be used automatically without making any configuration changes.

Related tasks

Enabling CipherSpecs

Related reference

ALTER CHANNEL

Migrating IBM MQ Managed File Transfer

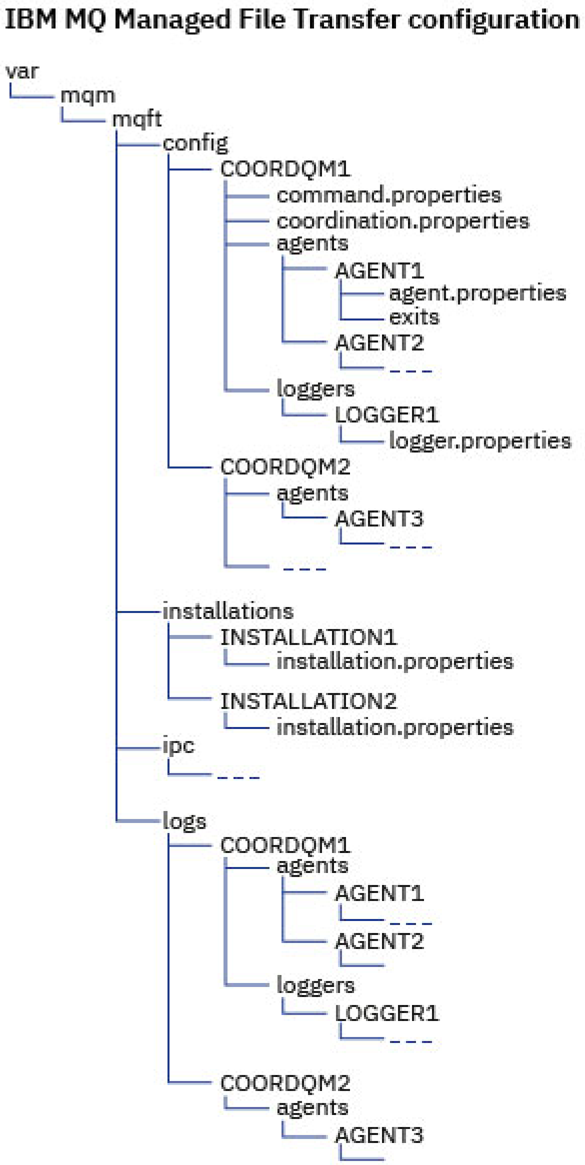
Use the following topics to guide you through various migration scenarios for IBM MQ Managed File Transfer.

Migration considerations for MFT

If you are planning to migrate Managed File Transfer, review the following information.

Configuration layout directly after installation in MFT

The diagram shows the configuration layout directly after installation as it is in IBM MQ.



Default configuration directories:

Information is split over four separate sub-directories: `config`, `installations`, `ipc`, and `logs`.

Default product root directories (MQ_DATA_PATH) are as follows:

- UNIX systems: `/var/mqm`
- Linux systems: `/var/mqm`
- Windows: the location of the configuration directory depends on the location of your primary IBM MQ installation. The default locations for primary installations are as follows:
  - 32-bit: `C:\Program Files (x86)\IBM\WebSphere MQ`
  - 64-bit: `C:\Program Files\IBM\MQ`

Configuration sub-directories are as follows:

- The *`MQ_DATA_PATH`*`/mqft/config` directory contains the parts of the configuration that are read-only for Managed File Transfer processes. For example, `agent.properties` and `command.properties`.

- The *MQ_DATA_PATH*/mqft/installations directory contains configuration information for each installation. The content of this directory is equivalent to the content of the wmqfte.properties file.
- The *MQ_DATA_PATH*/mqft/ipc directory contains IPC resources used internally to communicate between the Managed File Transfer components. Applicable to UNIX and Linux systems only.
- The *MQ_DATA_PATH*/mqft/logs directory contains the parts of the configuration that are written by Managed File Transfer processes. For example, trace information and log files.

installation.properties file

Linux UNIX On UNIX and Linux systems, the default location is *MQ_DATA_PATH*/*mqft/installations*/installation_name

Windows On Windows, the default location is *MQ_DATA_PATH*\mqft\installations\installation_name

logger.properties file

This file incorporates property information for stand-alone file loggers, stand-alone database loggers, and Java EE database loggers.

The default location is *MQ_DATA_PATH*/mqft/config/*coordination_qmgr_name*/loggers/*logger_name*.

Security changes

For IBM WebSphere MQ 7.5 or later, only users who are administrators (members of the **mqm** group) can run the following list of **fte** commands:

- fteChangeDefaultConfigurationOptions
- fteCreateAgent
- fteCreateBridgeAgent (create and configure an MFT protocol bridge agent)
- fteCreateCDAgent: create a Connect:Direct® bridge agent
- fteCreateLogger
- fteDeleteAgent
- fteDeleteLogger
- fteMigrateAgent: migrate an FTE V7.0 agent to MQ V7.5 or later
- fteMigrateConfigurationOptions: migrate an FTE V7.0 configuration to MQ V7.5 or later
- fteMigrateLogger: migrate an FTE V7.0 database logger to MQ V7.5 or later
- fteModifyAgent: run an MFT agent as a Windows service
- fteModifyLogger: run an MFT logger as a Windows service
- fteSetupCommands: create the MFT command.properties file
- fteSetupCoordination

Multi When using IBM WebSphere MQ 7.5 or later on Multiplatforms, only the user that the agent process is running under can run the fteSetAgentTraceLevel command.

z/OS For z/OS, the fteSetAgentTraceLevel command can be run by either:

- The same userid that the agent process is running as.
- Members of the group specified by the agent property **adminGroup**.

z/OS For z/OS, if the fteShowAgentDetails command is run to display information about a local agent, then the command can be run by either:

- The same userid that the agent process is running as.
- Members of the group specified by the agent property **adminGroup**.

For more information, see the **adminGroup** property in The MFT `agent.properties` file.

Security changes in IBM MQ 8.0

If you are running Managed File Transfer on IBM WebSphere MQ 7.0, and migrate to IBM MQ 8.0, the user ID information in the `MQMFTCredentials.xml` file is passed to the queue manager, but will not be acted upon. This is because the passing of user ID and password information only is supported in IBM MQ 8.0.

commandPath and agent sandboxes

For IBM MQ 8.0 and later, if an agent has been configured with an agent sandbox and the agent property **commandPath** has been set, then the directories specified by **commandPath** are automatically added to the denied paths when the agent starts. If the **commandPath** property is set on an agent which is not configured with an agent sandbox, then a new sandbox is set up automatically and the directories specified by the **commandPath** are added to the denied directories when the agent starts.

If the **commandPath** property is set on an agent which is not configured with an agent sandbox, then a new sandbox is set up automatically and the directories specified by the **commandPath** are added to the denied directories when the agent starts.

commandPath and user sandboxes

For IBM MQ 8.0 and later, if an agent has been configured with one or more user sandboxes, and has the agent property **commandPath** set, then the directories specified by **commandPath** (and all of their subdirectories) are automatically added as `<exclude>` elements to the `<read>` and `<write>` elements for each user sandbox when the agent starts up.

For more information about the **commandPath** property, see commandPath MFT property and The MFT `agent.properties` file.

Migrating MFT agents that run as a Windows service from IBM WebSphere MQ 7.5 to IBM MQ 9.0

Between IBM WebSphere MQ 7.5, IBM MQ 8.0, and IBM MQ 9.0, the default IBM MQ installation path has changed on the Windows platform.

If a queue manager is being migrated from IBM WebSphere MQ 7.5 to IBM MQ 8.0 or IBM MQ 9.0, any applications that are running on the same system as the queue manager must be reconfigured to load the IBM MQ libraries from the new installation location. This includes any IBM MQ Managed File Transfer agents that are running as a Windows service.

Carry out the following procedure:

1. Stop the agents that are associated with the IBM WebSphere MQ 7.5 queue manager. See stopping an MFT agent for more information. For example issue the following command:

   ```
   fteStopAgent  <agent_name>
   ```

2. Modify the agent to remove the service definition of the agent. See run an MFT agent as a Windows service for more information.

   For example, issue the following command:

   ```
   fteModifyAgent -agentName <agent_name>
   ```

3. Next, migrate the agent queue manager from IBM WebSphere MQ 7.5 to the later version by using the **setmqm** command. See associating a queue manager with an installation for more information about the **setmqm** command.
4. Finally, modify the agent to reconfigure the agent to run as a Windows service again, by using the **fteModifyAgent** command.

   For example:

```
fteModifyAgent -agentName AGENT1 -s -su fteuser -sp ftepassword
```

Migrating Managed File Transfer for z/OS agents to a later version

Follow these instructions to migrate an IBM MQ Managed File Transfer for z/OS agent to a later version. This topic uses IBM MQ 8.0 as the initial version and IBM MQ 9.1 as the later version.

Important: These instructions assume the following points:

- The coordination queue manager and command queue manager for the Managed File Transfer for z/OS (MFT) topology are not changing during the migration.
- The agent queue manager is not being migrated. The agent queue manager remains at the same IBM MQ for z/OS level before and after the migration.
- The agent being migrated is administered using JCL.

Step 1 - Initial setup

1. Install IBM MQ Managed File Transfer for z/OS 9.1 onto the same LPAR as IBM MQ Managed File Transfer for z/OS 8.0.
2. Locate the SBFGCMDS PDSE for IBM MQ MFT 9.1.

   For more information about the SBFGCMDS PDSE, see The SBFGCMDS library.
3. Within this data set, edit the BFGCOPY member to specify the name of the PDSE that is to be used to administer the agent.
4. Submit the BFGCOPY member. This copies the existing SBFGCMDS PDSE into a new PDSE with the name that was specified in Step "3" on page 250.

Step 2 - Modify the new PDSE

1. Within the new PDSE that you created using the BFGCOPY member, edit BFGCUSTM and update the following variables:

Variable	Value
++library++	Replace this variable with the name of the new PDSE.
armELEMENT	Set this to the same value that is used by the IBM MQ 8.0 agent.
armELEMTYPE	Set this to the same value that is used by the IBM MQ 8.0 agent.
BFG_DATA	Set this to the same value that is used by the IBM MQ 8.0 agent.
BFG_GROUP_NAME	Set this to the same value that is used by the IBM MQ 8.0 agent.
BFG_JAVA_HOME	This needs to be set to the UNIX System Services (USS) path for the Java Runtime Environment that is to be used to run the agent. Set this to same value that is used by the IBM MQ 8.0 agent.
BFG_JVM_PROPERTIES	Set this to the same value that is used by the IBM MQ 8.0 agent.
BFG_PROD	This variable should be set to the USS path for the IBM MQ Managed File Transfer for z/OS 9.1 installation.
BFG_WTO	Set this to the same value that is used by the IBM MQ 8.0 agent.
CLEAN_AGENT_PROPS	Set this to blank.
connectionQMgr	Set this to the same value that is used by the IBM MQ 8.0 agent.
coordinationQMgr	Set this to the same value that is used by the IBM MQ 8.0 agent.
CREDENTIALS_PATH	Set this to blank.

Variable	Value
DB_PROPS_PATH	Set this to the same value that is used by the IBM MQ 8.0 agent.
DB2_HLQ	Set this to the same value that is used by the IBM MQ 8.0 agent.
FTE_CONFIG	Set this to blank.
MQ_HLQ	Set this to the same value that is used by the IBM MQ 8.0 agent.
MQ_PATH	This variable needs to be set to the USS path for the IBM MQ installation associated with the agent queue manager.
NAME	Set this to the name of the agent that is being migrated.
productId	This is an additional property added in IBM MQ 9.1. If usage recording is to be used for the agent, set the property to the value MFT. Otherwise, you should leave the property blank.
QM	Set this to the same value that is used by the IBM MQ 8.0 agent. This ensures that when the agent is running IBM MQ 9.1, the agent uses the same agent queue manager that it did when running IBM MQ 8.0.
SERVICE_TYPE	AGENT
TMPDIR	Set this to the same value that is used by the IBM MQ 8.0 agent.

2. Save the modified BFGCUSTM member.
3. Submit BFGCUSTM. The member creates new members within the PDSE that can be used to administer the agent when it is using IBM MQ 9.1.

   Attention: The member requires exclusive access to the PDSE, so you must stop using it while BFGCUSTM is running.

Step 3 - Perform the migration

1. Locate the IBM MQ 8.0 PDSE that is used to administer the agent.
2. Within this PDSE, submit the BFGAGSP member to stop the agent. The agent stops once it has completed all of its managed transfers.
3. Next, locate the new IBM MQ 9.1 PDSE that has just been created for the agent.
4. Inside this PDSE, submit the BFGAGST member. This starts the agent using IBM MQ Managed File Transfer for z/OS 9.1

Related tasks

Using Managed File Transfer for z/OS

Migrating an FTE 7.0 agent to MQ version 7.5 or later

Use the **fteMigrateAgent** command to migrate an IBM WebSphere MQ File Transfer Edition 7.0 agent to IBM WebSphere MQ 7.5, or later. If an agent is configured to run as a Windows service, you must complete the optional steps in this task.

Before you begin

From IBM MQ 9.0, Managed File Transfer does not support web agents. If you attempt to use the **fteMigrateAgent** command to migrate a web agent from an earlier release to IBM MQ 9.0, an error message is displayed to explain that the migration of a web agent is not supported.

Identify instances of the IBM WebSphere MQ File Transfer Edition agent installed as part of IBM WebSphere MQ File Transfer Edition.

IBM i The information in this topic does not apply to IBM i. If you are using IBM i, migrate your queue manager to IBM MQ 8.0 before using Managed File Transfer for IBM MQ 8.0.

About this task

To migrate to IBM WebSphere MQ 7.5 or later, first stop the agent, then migrate the queue manager installation to IBM WebSphere MQ 7.5 or later, and additionally select the File Transfer components. When the queue manager has been migrated, you can migrate the agent's configuration to the IBM WebSphere MQ 7.5 or later, installation using the **fteMigrateAgent** command. Start the migrated agent using the IBM WebSphere MQ 7.5 or later, queue manager.

If the agent is connecting to its queue manager using IBM WebSphere MQ bindings mode and the queue manager is at IBM WebSphere MQ 7.0.1 Fix Pack 6 or later, you can alternatively perform a side-by-side migration by using multiple installations and the steps listed below. If you do not wish to perform a side-by-side migration, instead migrate the queue manager and agent using the steps as described above.

If the agent is connecting to its queue manager using IBM WebSphere MQ bindings mode and the queue manager is at IBM WebSphere MQ 7.0.1 Fix Pack 5 or earlier, you must either migrate this installation to IBM WebSphere MQ 7.0.1 Fix Pack 6, to permit a side-by-side migration, or you must migrate this installation directly to IBM WebSphere MQ 7.5 or later.

If the agent is connecting to its queue manager across a network as an IBM WebSphere MQ client, you can migrate the agent to IBM WebSphere MQ 7.5 or later by completing step "7" on page 252 only.

If the agent is configured as a Windows service you must run the command with the **-f** parameter. For more information, see **fteMigrateAgent**.

Procedure

1. Install IBM WebSphere MQ 7.5 or later alongside the existing queue manager installation, selecting the MQ Server and File Transfer components.
2. Stop the IBM WebSphere MQ File Transfer Edition v7.0.x agent.
3. Move the queue manager to the IBM WebSphere MQ 7.5, or later installation.
4. Use the **fteMigrateAgent** command to migrate the agent configuration from the IBM WebSphere MQ File Transfer Edition v7.0.x installation to the agent capability integrated with IBM WebSphere MQ 7.5 or later.
5. Optional: If the IBM WebSphere MQ File Transfer Edition v7.0.x agent is configured as a Windows service, complete the following optional steps:
   a) Use the **fteModifyAgent** command to reconfigure the agent so that it is no longer a Windows service.
   b) Use the **fteModifyAgent** command from the IBM WebSphere MQ 7.5 or later, installation to configure the IBM WebSphere MQ 7.5 or later, version of the agent to be a Windows service.
6. Start the IBM WebSphere MQ 7.5 or later, agent.
7. Optional: To migrate an agent connecting to its queue manager across a network as a client:
   a) Install IBM WebSphere MQ 7.5 or later, onto the system
   b) Use the **fteMigrateAgent** command to migrate the agent configuration from the WMQFTE v7.0.x installation to the agent capability integrated with IBM WebSphere MQ 7.5 or later.
   c) Stop the IBM WebSphere MQ File Transfer Edition v7.0.x agent.
   d) If the IBM WebSphere MQ File Transfer Edition v7.0.x agent is configured as a Windows service, use the **fteModifyAgent** command from the IBM WebSphere MQ File Transfer Edition v7.0.x installation to re-configure the agent so that it is not run as a Windows service
   e) If the IBM WebSphere MQ File Transfer Edition v7.0.x agent is configured as a Windows service, use the **fteModifyAgent** command from the IBM WebSphere MQ 7.5 or later installation to configure the IBM WebSphere MQ 7.5 or later agent as a Windows service.
   f) Start the IBM WebSphere MQ 7.5 or later agent.

Migrating an FTE V7.0 database logger to MQ V7.5 or later

Use the **fteMigrateLogger** command to migrate a stand-alone IBM WebSphere MQ File Transfer Edition 7.0 database logger to IBM WebSphere MQ 7.5, or later. If you have configured the database logger as a Windows service, you must complete extra migration steps.

Before you begin

Identify instances of the stand-alone database logger. You must update database logger instances before other parts of the Managed File Transfer network so that these instances can correctly process the latest versions of the transfer log messages they receive.

About this task

To migrate directly to IBM WebSphere MQ 7.5, or later, first stop the logger and then migrate this installation to the required version. When this migration is complete, use the fteMigrateLogger: migrate an FTE V7.0 database logger to MQ V7.5 or later command to migrate the database logger configuration to required version of IBM MQ.

If the queue manager is IBM WebSphere MQ File Transfer Edition 7.0.1 Fix Pack 6 or later, you can alternatively perform a side-by-side migration by using multiple installations as detailed in the following steps. If you do not wish to perform a side-by-side migration, instead migrate the queue manager and logger using the steps as described above.

If the database logger is connected to an IBM WebSphere MQ File Transfer Edition 7.0.1 Fix Pack 5 or earlier queue manager, you must first either migrate this installation to IBM WebSphere MQ File Transfer Edition 7.0.1 Fix Pack 6, to permit a side-by-side migration, or you must migrate this installation directly to IBM WebSphere MQ 7.5, or later, and additionally select the File Transfer components.

If the database logger is configured as a Windows service you must run the **fteMigrateLogger** command with the **-f** parameter. For more information, see fteMigrateLogger: migrate an FTE V7.0 database logger to MQ V7.5 or later.

Procedure

1. Install IBM WebSphere MQ 7.5, or later, alongside the existing IBM WebSphere MQ File Transfer Edition 7.0 installation, selecting the MQ Server and Managed File Transfer Service components.
2. Stop the IBM WebSphere MQ File Transfer Edition 7.0 database logger.
3. Move the queue manager to the IBM WebSphere MQ 7.5, or later installation.
4. Use the fteMigrateLogger command to migrate the database logger configuration from the IBM WebSphere MQ File Transfer Edition 7.0 installation to the IBM WebSphere MQ 7.5, or later database logger.
5. Create the tables needed for the database logger to start by running the sql scripts in *MQ_INSTALLATION_PATH*/mqft/sql. There is one script for each increase in product level, run all that are appropriate in order.

   You must run the upgrade sql scripts in version order, starting with their current level of IBM WebSphere MQ File Transfer Edition or Managed File Transfer. The available scripts, where *** can be Db2, Oracle, or z/OS, are as follows:

   - ftelog_tables_***_701-702.sql
   - ftelog_tables_***_702-703.sql
   - ftelog_tables_***_703-704.sql
   - ftelog_tables_***_704-750.sql
   - ftelog_tables_***_750-7502.sql
   - ftelog_tables_***_7502-800.sql

For example, if you are using IBM WebSphere MQ File Transfer Edition 7.0.3 and are migrating to IBM MQ 8.0.0.0, run 703-704, 704-750, 750-7502, and 7502-800.

Note: z/OS On z/OS, you can go directly from 704 to 800, using ftelog_tables_zos_704-800.sql

The scripts up to ftelog_tables_***_704-750.sql are included in IBM MQ 8.0.0.0, and are located in the *MQ_INSTALL ATION_PATH*/mqft/sql directory. Upgrades beyond 750 were not shipped with IBM MQ 8.0.0.0; if they are not present in the directory, you can download them from the link specified in APAR IT01841.

6. Optional: If the IBM WebSphere MQ File Transfer Edition 7.0 database logger was configured as a Windows service, complete the following steps:
   a) Reconfigure the database logger so that it is no longer a Windows service using the fteModifyDatabaseLogger command.
   b) Reconfigure the Managed File Transfer for IBM WebSphere MQ 7.5, or later logger so that it is a Windows service using the fteModifyLogger command.
7. Start the Managed File Transfer for IBM WebSphere MQ 7.5, or later database logger.

Results

The database logger has now been migrated from IBM WebSphere MQ File Transfer Edition 7.0 to IBM WebSphere MQ 7.5, or later.

Migrating MFT: Increasing the log db page size for Db2 on UNIX, Linux, and Windows

If your database is Db2 on UNIX, Linux or Windows, and you created your log database with a page size of less than 8 KB, you must increase the page size of the database before migrating to the IBM WebSphere MQ File Transfer Edition 7.0.3 or later tables.

Procedure

1. If you have not already stopped your database logger, stop your database logger using the **fteStopDatabaseLogger** command.
2. Back up your log database using the tools provided by Db2.
3. Use the Db2 **export** command to transfer the data from your log database tables to files on disk.

   Note: You must specify large object files for tables that include large objects. Those tables are CALL_RESULT and METADATA.
4. Drop your log database tables.
5. Create a table space with a page size of at least 8 KB and with an associated buffer pool with a page size of at least 8 KB.

   Give your new table space a name. For example, FTE8KSPACE.
6. Edit the ftelog_tables_db2.sql file so that the commands create tables in the new table space.

   In the ftelog_tables_db2.sql file, change all occurrences of the text IN "USERSPACE1" to IN "*new_tablespace_name*". For example, change IN "USERSPACE1" to IN "FTE8KSPACE".
7. Run the SQL commands in the ftelog_tables_db2.sql file against your database.
8. Use the Db2 **load** command to transfer the exported data into the new tables.

   Note:

   - **Map the column names based on the column names found in the input file**. Ensure that the input column names and target column names match up in those tables that have changed their structure.

- You must specify the IDENTITY OVERRIDE behavior on the identity column of all tables, except for MONITOR and TRANSFER. Specifying this behavior ensures that the row IDs are not regenerated during the load operation.

9. Run the Db2 **`set integrity`** command with integrity status values of **immediate** and **checked**, against the following tables in the order given:

   - CALL_ARGUMENT
   - MONITOR
   - MONITOR_ACTION
   - MONITOR_EXIT_RESULT
   - MONITOR_METADATA
   - SCHEDULE_ACTION
   - SCHEDULE
   - SCHEDULE_ITEM
   - TRANSFER
   - TRANSFER_CALLS
   - TRANSFER_EVENT
   - TRANSFER_ITEM
   - TRANSFER_STATS
   - TRIGGER_CONDITION

10. In tables with generated ID columns, set the ID generators to begin from a value one higher than the existing highest ID value.

    The following tables have generated ID columns:

    - AUTH_EVENT
    - CALL
    - CALL_ARGUMENT
    - CALL_RESULT
    - FILE_SPACE_ENTRY
    - METADATA
    - MONITOR_ACTION
    - MONITOR_EXIT_RESULT
    - MONITOR_METADATA
    - SCHEDULE
    - SCHEDULE_ACTION
    - SCHEDULE_ITEM
    - SCHEDULE_SPEC
    - TRANSFER_CALLS
    - TRANSFER_CD_NODE
    - TRANSFER_CORRELATOR
    - TRANSFER_EVENT
    - TRANSFER_EXIT
    - TRANSFER_ITEM
    - TRANSFER_ITEM_ATTRIBUTES
    - TRANSFER_STATS
    - TRIGGER_CONDITION

To set the generated IDs of these tables to the correct value perform the following steps for each table:

a) Determine the maximum ID value in the existing data.

You can find this value by running this SQL statement:

```
SELECT MAX(ID) FROM FTELOG.table_name
```

The value returned from this command is the maximum existing ID in the specified table.

b) Alter the table to set the ID generator to begin from a new value that is 1 higher than the value returned by the previous step.

You can set this value by running the following SQL statement:

```
ALTER TABLE FTELOG.table_name ALTER COLUMN ID RESTART WITH value
```

Related tasks

Migrating from the stand-alone database logger to the JEE database logger for MFT

"Migrating the database tables on Db2 on z/OS to MQ V8.0 or later" on page 213

If your database is Db2 on a z/OS system, you must complete the following steps to migrate between from IBM WebSphere MQ File Transfer Edition 7.0.3 to IBM WebSphere MQ File Transfer Edition 7.0.4, and from IBM WebSphere MQ File Transfer Edition 7.0.4 to Managed File Transfer for IBM MQ 8.0 or later. The Db2 tables have different structures from previous releases. For example, there are new columns in some tables, and some variable characters columns can be larger, so the tables from previous releases have to be migrated to the Db2 format.

V 9.1.4 Migrating Managed File Transfer agents from an earlier version

Agents migrated from versions of IBM MQ prior to IBM MQ 9.1.4 run as non highly available. You can make them run in high availability mode by carrying out the following procedure.

Procedure

1. Create the SYSTEM.FTE.HA.<agent name> queue in the agent queue manager using the following sample definition:

```
DEFINE QLOCAL(SYSTEM.FTE.HA.SRC) +
 DEFPRTY(0) +
 DEFSOPT(SHARED) +
 GET(ENABLED) +
 MAXDEPTH(0) +
 MAXMSGL(0) +
 MSGDLVSQ(PRIORITY) +
 PUT(ENABLED) +
 RETINTVL(999999999) +
 SHARE +
 NOTRIGGER +
 USAGE(NORMAL) +
 REPLACE
```

2. Provide the required authorities on the queue for the agent to open the queue for GET.
3. Create a replica of the agent configuration on another machine
4. Add the **highlyAvailable** property, and set the property to *true*, in the agent.properties file for both agent configurations.

Related concepts

Maintenance in highly available agents

Migrating MFT to a new machine with a different operating system

The core steps required to successfully achieve a migration of MFT configurations to a new system or platform. The task is primarily focused on MFT configuration migration, but also discusses queue manager migration where appropriate.

Before you begin

Ensure that any agents you are going to migrate have completed any in-progress or pending transfers, and that you have taken a back up of:

- The coordination queue manager
- Agent queue managers
- Agents
- Resource Monitors
- Transfer Templates
- Scheduled Transfers

Important: IBM MQ installation names on one system are unlikely to match the installation names on the new system unless the old and new systems only have one installation, or you specify an installation name as part of the IBM MQ installation process.

About this task

The following migration procedure is based on the scenario where QMA is both the coordination queue manager for topology, and the agent queue manager for an agent called Agent1.

Agent1 has a monitor, transfer template and scheduled transfer. QMA also connects to a queue manager called QMB running on another system using their sender and receiver channels for file transfers.

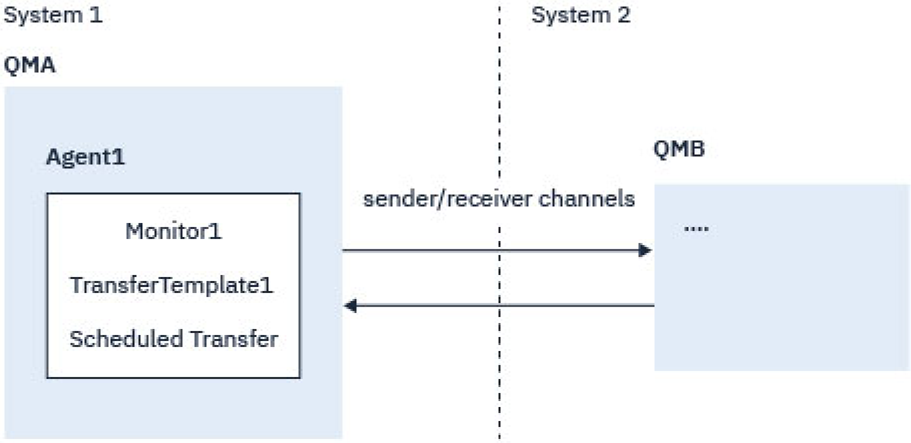


Figure 14. Migrating the MFT configuration on System 1

Attention: The following procedure explains only how to backup and restore MFT configurations. If you are migrating MFT to a new machine with the same operating system, queue manager data and log files can be backed up and restored by copying all the data files from the old system to the appropriate directories on the new system.

However, if the new machine has a different operating system, it is not possible to migrate the data files, because they are created platform specific.

Procedure

1. Backup procedure

a) Save the queue manager configuration using the **dmpmqcfg** command to rebuild it later from its definition.

For example:

```
dmpmqcfg -m QMA -a > /mq/backups/QMA.mqsc
```

b) Back up the configuration files for the agent that are stored under the IBM MQ data directory `/MQ_DATA_PATH/mqft`

The `mqft` directory normally has three sub-directories, which are `config`, `installation`, and `logs`. These contain agent installation data, configuration, and database logger files respectively.

If the agent is Protocol Bridge Agent, the `ProtocolBridgeCredentials.xml` file in the agent configuration directory also needs to backed up. This file defines the user names and credential information that the protocol bridge agent uses to authorize itself with the protocol server.

c) Export the configuration of the resource monitor to an XML file using the MFT **ftelistMonitors** command with the **-ox** option.

For example:

```
fteListMonitors -ma Agent1 -mn Monitor -ox Monitor1Definition.xml
```

d) Export transfer templates to XML files using the MFT **fteListTemplates** command with the **-x** and **-o** options.

For example, the following command creates `TransferTemplate1.xml` in the current directory:

```
fteListTemplates -x -o . TransferTemplate1
```

e) Manually back up the scheduled transfer definitions.

It is not possible to export the definitions to XML files, but you can list scheduled transfers using the MFT fteListScheduledTransfers command and backing up the definitions manually.

2. Recreate procedure

a) Recreate queue manager QMA after installing IBM MQ and MFT on the new system.

b) Restore the QMA configuration by running the **runmqsc** command to parse in the queue manager configuration saved in Step “1.a” on page 258

For example:

```
runmqsc QMA< /mq/backups/QMA.mqsc
```

c) Recreate the sender and receiver channels that connect to QMB on System two.

d) On the QMB queue manager side, update the connections details, such as host name and port number of the sender channel that connects to QMA.

e) Recreate `Agent1` by copying all of the backed up agent configuration files to the new system, and start the agent.

f) Import the XML file for `Monitor1` using the MFT fteCreateMonitor command with the **-ix** and **-f** options.

For example:

```
fteCreateMonitor  -ix Monitor1Definition.xml  -f
```

g) Publish a message containing the contents of TransferTemplate1.xml in the message body to the SYSTEM.FTE topic on the coordination queue manager.

Use a stand-alone application, and specify the topic string:

```
SYSTEM.FTE/Templates/<template_id>
```

where <`template_id`> is the transfer template ID that can be found inside the `TransferTemplate1.xml` file.

For example, if the xml contains:

```
<?xml version="1.0" encoding="UTF-8"?><transferTemplateid="a7838085-0f2a-4980-
b958-2dbbdfb22702"
version="6.00">
```

, the topic string should be:

```
SYSTEM.FTE/Templates/a7838085-0f2a-4980-b958-2dbbdfb22702
```

h) Recreate the scheduled transfers manually using the MFT **`fteCreateTransfers`** command.

Migrating IBM MQ Internet Pass-Thru

Follow this procedure to upgrade to a new version of IBM MQ Internet Pass-Thru (MQIPT), or to apply fix pack maintenance to your MQIPT installation. V 9.1.4 You can also use this procedure to upgrade from MQIPT support pack 2.1 to MQIPT for IBM MQ 9.1.

Procedure

1. Make backups of your data.

   See Making backups for details.
2. Install the new version of MQIPT.

   You can install the new version of MQIPT before uninstalling any versions of MQIPT that are currently installed. See Installing MQIPT for details.
3. Restore the backed-up data files to the MQIPT home directory to be used by the new installation.

   If the MQIPT installation directory is used as the home directory, then overwrite any newly installed copies of data files with the backed-up files.
4. Ensure that any properties that contain file names in the new `mqipt.conf` configuration file, refer to files to be used by the new installation of MQIPT.
5. Review the list of changes and new features in the new version or fix pack of MQIPT.

   If you need to make any changes to the MQIPT configuration for the new version, make the necessary changes to the new copies of the data files.
6. Stop the current version of MQIPT by issuing the following command:

   ```
   mqiptAdmin -stop
   ```
7. Start MQIPT at the latest version by issuing the following command:

   - On Windows systems:

     ```
     MQIPT_INSTALLATION_PATH\bin\mqipt MQIPT_HOME_DIR
     ```
   - On UNIX and Linux systems:

     ```
     MQIPT_INSTALLATION_PATH/bin/mqipt MQIPT_HOME_DIR
     ```

   where

   - *MQIPT_INSTALLATION_PATH* is the directory where that latest version of MQIPT is installed.
   - *MQIPT_HOME_DIR* is the MQIPT home directory containing the data files to be used by the latest installation of MQIPT.
8. Test that MQIPT works correctly at the latest version.

   After you confirm that the latest version of MQIPT is configured correctly, you can uninstall the previous version. See Uninstalling MQIPT for details.

9. V 9.1.5 If there are any passwords in your MQIPT configuration that have not been encrypted, or passwords that were encrypted prior to MQIPT in IBM MQ 9.1.5, encrypt these passwords using the latest protection method by following the procedure in Encrypting stored passwords.

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 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: