
Unit 11. Backing up and restoring IBM MQ messages and object definitions

Estimated time

01:00

Overview

In this unit, you learn about the various ways that IBM MQ maintains messages. You learn about differences between circular and linear logging, the implications of using persistence, and transaction management. You also learn about the methods for capturing and restoring an object image and backing up and restoring IBM MQ object definitions.

How you will check your progress

- Review questions
- Hands-on exercises

References

IBM Knowledge Center for IBM MQ V9

Unit objectives

- Describe how IBM MQ uses logging to record significant changes to the data controlled by the queue manager
- Describe the difference between circular and linear logging
- Develop a method for backing up the IBM MQ environment
- Use a media image to recover objects that become damaged
- Save the queue manager object definitions

Backing up and restoring IBM MQ messages and object definitions

© Copyright IBM Corporation 2017

Figure 11-1. Unit objectives

The need to back up

- Allow recovery of queue managers against possible corruption or loss of data that hardware failures cause
- IBM MQ file system backup can be done as part of a full system backup



Backing up and restoring IBM MQ messages and object definitions

© Copyright IBM Corporation 2017

Figure 11-2. The need to back up

Periodically backing up MQ files and its objects allows the recovery of queue managers against possible corruption that hardware failures cause.

If the hardware fails, a queue manager is forced to stop. If any queue manager log data is lost because of the hardware failure, the queue manager might be unable to restart. Through backing up queue manager data, you might be able to recover some, or all, of the lost queue manager data.

Backing up the IBM MQ file system

- Ensure that queue managers are not running
- Locate directories where the queue manager places its data and log files
 - Default paths on Windows
 - Programs: C:\Program Files\IBM\MQ\
 - Data: C:\ProgramData\IBM\MQ
 - Default paths on UNIX and Linux
 - Programs: /usr/mqm
 - Data: /var/mqm
- Take copies of all the data from the queue manager and log file directories, including subdirectories
- Restart queue managers

[Backing up and restoring IBM MQ messages and object definitions](#)

© Copyright IBM Corporation 2017

Figure 11-3. Backing up the IBM MQ file system

The queue manager must be stopped when you back up. If you try to back up a running queue manager, the backup might not be consistent because of updates in progress when the files were copied. If possible, stop your queue manager in an orderly way.

Before you back up, you must know the locations of the MQ components on the file system. The file system layout for MQ varies by operating system. The figure lists the default installation location for MQ software components and data. The paths might be different for your installation. Use the information in the configuration files to find the directories under which the queue manager places its data and its log files.

Some of the directories might be empty, but you need them all to restore from the backup, so save all directories and subdirectories.

Preserve the ownerships of the files. For MQ on UNIX and Linux, you can preserve ownership by using the `tar` command.

Restart the queue manager after the backup is complete.

Overview of IBM MQ objects

- IBM MQ object types include:
 - Queue managers
 - Queues
 - Channels
 - Process definitions
 - Listeners
 - Namelists
 - Client connection channels
 - Services
 - Authentication information objects
- Anything that you define to IBM MQ by using MQSC

Backing up and restoring IBM MQ messages and object definitions

© Copyright IBM Corporation 2017

Figure 11-4. Overview of IBM MQ objects

This figure is a review of the various MQ object types that are defined to MQ. Each object has an object definition that can be backed up.

Why back up IBM MQ object definitions

- To re-create objects that were deleted or missing
- To simplify administration tasks in IBM MQ including:
 - Migration
 - Cloning systems
 - Auditing

Backing up and restoring IBM MQ messages and object definitions

© Copyright IBM Corporation 2017

Figure 11-5. Why back up IBM MQ object definitions

When you update applications, you might want to keep the same object definitions from the queue managers of source environment. By backing up the MQ object definitions, you can restore them on the new queue manager quickly and easily. Otherwise, a manual process would be required to restore the queue managers.

Similarly, when cloning systems (typically for clustering) or for automation of application migrations, a backup provides a quick and simple way to copy the object definitions from one queue manager to another.

If you are auditing your MQ system, backing up the object definition provides a view of all the definitions in one place.

Fundamentally, backing up object definitions is about saving time and effort by using the backup file to re-create the object definition on a new queue manager.

Capturing IBM MQ object definitions

- Save IBM MQ configuration command: `dmpmqcfg`
 - Use IBM MQ commands to save queue manager object definitions
 - Can connect to local queue managers or use client connections to remote systems
- Output from MQSC `DISPLAY` command

Backing up and restoring IBM MQ messages and object definitions

© Copyright IBM Corporation 2017

Figure 11-6. Capturing IBM MQ object definitions

You can capture object definitions by using MQSC and the `dmpmqcfg` command.

- Save the MQ configuration by using the integrated `dmpmqcfg` command (the most comprehensive option).
- Save the output from the MQSC display queue manager, queue, and channel commands in an external file so that you have a reference of the properties and object definitions.

Capturing resource definitions with the DISPLAY command

```
C:\> runmqsc QmgrName
DISPLAY QL(*) ALL
DISPLAY QR(*) ALL
DISPLAY CHL(*) CHLTYPE(*) ALL
```

- Simple method to save properties of all the object definitions
- Output is formatted in single or multiple column formats
 - Needs reformatting to be used for restore purposes

Backing up and restoring IBM MQ messages and object definitions

© Copyright IBM Corporation 2017

Figure 11-7. Capturing resource definitions with the DISPLAY command

To use MQSC commands to see all the object definitions, enter **DISPLAY** and the **ALL** attribute. This command reports the resource definitions for the specified queue manager. This option does not provide a backup of all objects.

The output is useful for a quick overview of the object definitions; however, it is formatted in single or multi-columns, making it difficult to be used for restoration purposes. Reformatting is required to use the output for restoration purposes.

Custom code can be created to convert these reports into MQSC define commands, but custom code is burdensome and requires updating every time that a change occurs.

Save IBM MQ configuration: `dmpmqcfg`

- Back up object definitions and authorities, restore with MQSC
- Generate reports on objects and their access control
- Use it to rebuild queue manager on a new version of IBM MQ
- Must have appropriate authority to each object that is inquired
- Variety of output formats supported
 - Multi-line MQSC that can be used as direct input to `runmqsc`
 - MQSC on single line
 - MQSC with two lines that contain command and commented strings
 - `setmqaut` statements
 - Security policies statements for IBM MQ Advanced Message Security

[Backing up and restoring IBM MQ messages and object definitions](#)

© Copyright IBM Corporation 2017

Figure 11-8. Save IBM MQ configuration: `dmpmqcfg`

The `dmpmqcfg` command extracts a queue manager's configuration and displays it in MQSC syntax.

The `dmpmqcfg` command can connect to local queue managers or use client connections to remote systems.

You can specify the syntax of the backup report in the `dmpmqcfg` command. The syntax options are:

- Multi-line MQSC that can be used as direct input on a `runmqsc` control command
- MQSC with all attributes on a single line for line comparison
- MQ set authority access (`setmqaut`) statements for Linux, UNIX, and Windows queue managers
- On Linux, grant MQ authority access (`grtmqaut`) for granting access to the objects on iSeries

Backing up queue manager configuration

To take a backup copy of a queue manager's configuration:

1. Ensure that the queue manager is running.
2. Run `dmpmqcfg` command with the following options:
 - Formatting option of `-o mqsc` to create multi-line MQSC that can be used as direct input to MQSC
 - All attributes option `-a`
 - Standard output redirection to write the definitions into a file

Example:

```
dmpmqcfg -o mqsc -m MYQMGR -a > /mq/backups/MYQMGR.mqsc
```

Figure 11-9. Backing up queue manager configuration

This figure lists the steps for backing up a queue manager configuration by using the `dmpmqcfg` control command.



Important

When you back up the MQ file system, the queue manager must be stopped.

When you back up an MQ configuration by using the `dmpmqcfg` command, the queue manager must be running.

The example command in the figure creates a backup of all attributes (`-a`) of the queue manager that named MYQMGR (`-m MYQMGR`) in a multi-line MQSC format (`-o mqsc`). The results of the command `dmpmqcfg` are redirected to the `MYQMGR.mqsc` in the `/mq/backups/` directory.

Syntax examples of `dmpmqcfg`

```
dmpmqcfg [-m QMgrName] [-n ObjName] [-c Connection]
[-t ObjType] [-x ExportType] [-o Format] [-a] [-z]
```

- Example 1: Local queue manager, all objects, and authorities:

```
dmpmqcfg -m MYQMGR
```

- Example 2: Local queue manager, all object definitions of SYSTEM queues showing all attributes

```
dmpmqcfg -m MYQMGR -n SYSTEM.* -t queue -x object -a
```

- Example 3: Local queue manager, all channel authentication records, silence any warnings

```
dmpmqcfg -m MYQMGR -x chlauth -z
```

- Example 4: Local queue manager, all authorities in `setmqaut` format

```
dmpmqcfg -m MYQMGR -o setmqaut
```

- Example 5: Dynamic client connection to remote queue manager

```
dmpmqcfg -m RMTQMGR -c "DEFINE CHANNEL(SYSTEM.DEF.SVRCONN) +
CHLTYPE(CLNTCONN) CONNAME('dev.ibm.com(1414)')"
```

Backing up and restoring IBM MQ messages and object definitions

© Copyright IBM Corporation 2017

Figure 11-10. Syntax examples of `dmpmqcfg`

The figure shows examples of the `dmpmqcfg` control command.

Using `dmpmqcfg`: Output MQSC

```
*****
* Script generated on 2016-10-31 at 15.44.12
* Script generated by user 'JSMITH' on host 'dev.IBM.COM'
* Queue manager name: MYQMGR
* Queue manager platform: Windows
* Queue manager command level: (900/900)
* Command issued: dmpmqcfg -m MYQMGR
*****
ALTER QMGR +
* ALTDATE(2016-10-31) +
* ALTTIME(14.32.42) +
CCSID(37) +
CLWLUSEQ(LOCAL) +
* COMMANDQ(SYSTEM.ADMIN.COMMAND.QUEUE) +
* CRDATE(2016-10-31) +
* CRTIME(14.32.42) +
* PLATFROM(WINDOWS) +
* QMID(MYQMGR_2011-10-31_14.32.42) +
SSLKEYR('/QIBM/UserData/ICSS/Cert/Server/Default') +
* VERSION(09000000) +
FORCE
SET AUTHREC +
PROFILE('self') +
PRINCIPAL('QMQM') +
OBJTYPE(QMGR) +
...
...
```

Backing up and restoring IBM MQ messages and object definitions

© Copyright IBM Corporation 2017

Figure 11-11. Using `dmpmqcfg`: Output MQSC

The figure shows an example of a multi-line MQSC report that the `dmpmqcfg` command creates when the `-o mqsc` option is specified.

Some of the attributes in the `ALTER QMGR` command are commented because they are unique to the queue manager. These attributes are not used when you create a queue manager.

Using `dmpmqcfg`: Output `setmqaut`

```
#####
# Script generated on 2016-10-31 at 15.59.09
# Script generated by user 'JSMITH' on host 'dev.IBM.COM'
# Queue manager name: JSMITH
# Queue manager platform: Windows
# Queue manager command level: (900/9--)
# Command issued: dmpmqcfg -m MYQMGR -o setmqaut
#####
setmqaut -m MYQMGR -t qmgr -p QMQM +altusr +chg +connect +crt +dlt +dsp
+inq +set +setall +setid +ctrl +system
setmqaut -m MYQMGR -t qmgr -g QMQMADM +altusr +chg +connect +dlt +dsp
+inq +set +setall +setid +ctrl +system
setmqaut -m MYQMGR -n SYSTEM.ADMIN.ACOUNTING.QUEUE -t queue -p QMQM
+browse +chg +clr +dlt +dsp +get +inq +put +passall +passid +set
+setall +setid
setmqaut -m MYQMGR -n SYSTEM.ADMIN.ACOUNTING.QUEUE -t queue -g QMQMADM
+browse +chg +clr +dlt +dsp +get +inq +put +passall +passid +set
+setall +setid
setmqaut -m MYQMGR -n SYSTEM.ADMIN.ACTIVITY.QUEUE -t queue -p QMQM
+browse +chg +clr +dlt +dsp +get +inq +put +passall +passid +set
+setall +setid
setmqaut -m MYQMGR -n SYSTEM.ADMIN.ACTIVITY.QUEUE -t queue -g QMQMADM
+browse +chg +clr +dlt +dsp +get +inq +put +passall +passid +set
+setall +setid
...
...
```

Backing up and restoring IBM MQ messages and object definitions

© Copyright IBM Corporation 2017

Figure 11-12. Using `dmpmqcfg`: Output `setmqaut`

The figure shows an example of a `dmpmqcfg` report that is created in the `setmqaut` format. You can run this report against a queue manager to restore MQ object authority.

Overview of security definitions

- Security definitions in this context refer to IBM MQ OAM security definitions, not UNIX or Windows security profiles or SSL data
- OAM security definitions
 - Define the authorizations that are given to a specified user group
 - Use `setmqaut` to create

Backing up and restoring IBM MQ messages and object definitions

© Copyright IBM Corporation 2017

Figure 11-13. Overview of security definitions

The `setmqaut` commands that the `dmpmqcfg -o setmqaut` command generates are MQ OAM commands. They do not include Linux, UNIX, or Windows security profiles or SSL data.

It is important to understand that the security definitions refer to the MQ OAM authorities.

Need to back up security definitions

- Restoration of security definitions
- Migration between environments
- Audit purposes

Backing up and restoring IBM MQ messages and object definitions

© Copyright IBM Corporation 2017

Figure 11-14. Need to back up security definitions

Similar to backing up object definitions, backing up security definitions allows users to save time and effort by using the backup file to re-create the security definitions rather than doing it manually.

Regarding audits, backing up security definitions provides a way to view the security definition information in one place and in a format that might be more meaningful to the viewer.

Backing up security definitions

- Use the **dmpmqaut** command to create a report of the current authorizations

Sample output:

```
profile: a.b./*
object type: queue
entity: user1
type: principal
authority: get, browse, put, inq
```

- Use the **amqoamd -s** command to create a report of the current authorizations in a **setmqaut** format

Sample output:

```
setmqaut -m AJGMQ1 -n @CLASS -t channel -p andrew@IBM-L3M1562 +crt
setmqaut -m AJGMQ1 -n @CLASS -t channel -g mqm +crt
setmqaut -m AJGMQ1 -n @CLASS -t authinfo -p andrew@IBM-L3M1562 +crt
setmqaut -m AJGMQ1 -n @CLASS -t authinfo -g mqm +crt
setmqaut -m AJGMQ1 -n SYSTEM.BROKER.ADMIN.STREAM -t queue -p
MUSR_MQADMIN@IBM-L3M1562 +browse +chg +clr +dlt +dsp +get +inq +put
+passall +passid +set +setall +setid
```

[Backing up and restoring IBM MQ messages and object definitions](#)

© Copyright IBM Corporation 2017

Figure 11-15. Backing up security definitions

MQ provides three ways to back up the OAM security definitions: **dmpmqcfg**, **dmpmqaut**, and **amqoamd -s**.

The output from the **dmpmqaut** command is formatted in a report style. This output cannot be easily used for restoring security definitions unless it is reformatted.

The output from the **amqoamd -s** command, which is shown in the figure, is generated as a series of **setmqaut** commands. These commands can be used in a command window to re-create the OAM security definitions.

dmpmqaut command

```
dmpmqaut -m QMgrName [-n Profile | -l | -a] -t ObjectType
-s ServiceComponent [-p PrincipalName | -g GroupName]
[-e | -x]
```

| Attribute | Description |
|-----------------------------|---|
| -n <i>Profile</i> | Name of the profile for which to dump authorizations |
| -l | Dump only the profile name and type |
| -a | Generate set authority commands |
| -t <i>ObjectType</i> | The type of object for which to dump authorizations Possible values are: authinfo, channel, clntconn, listener, namelist, process, queue, qmgr, rqmname, service, or topic |
| -e | Display all profiles that are used to calculate the cumulative authority that the entity has for the object that is specified in -n <i>Profile</i> |
| -x | Display all profiles with the same name as specified in -n <i>Profile</i> |

Backing up and restoring IBM MQ messages and object definitions

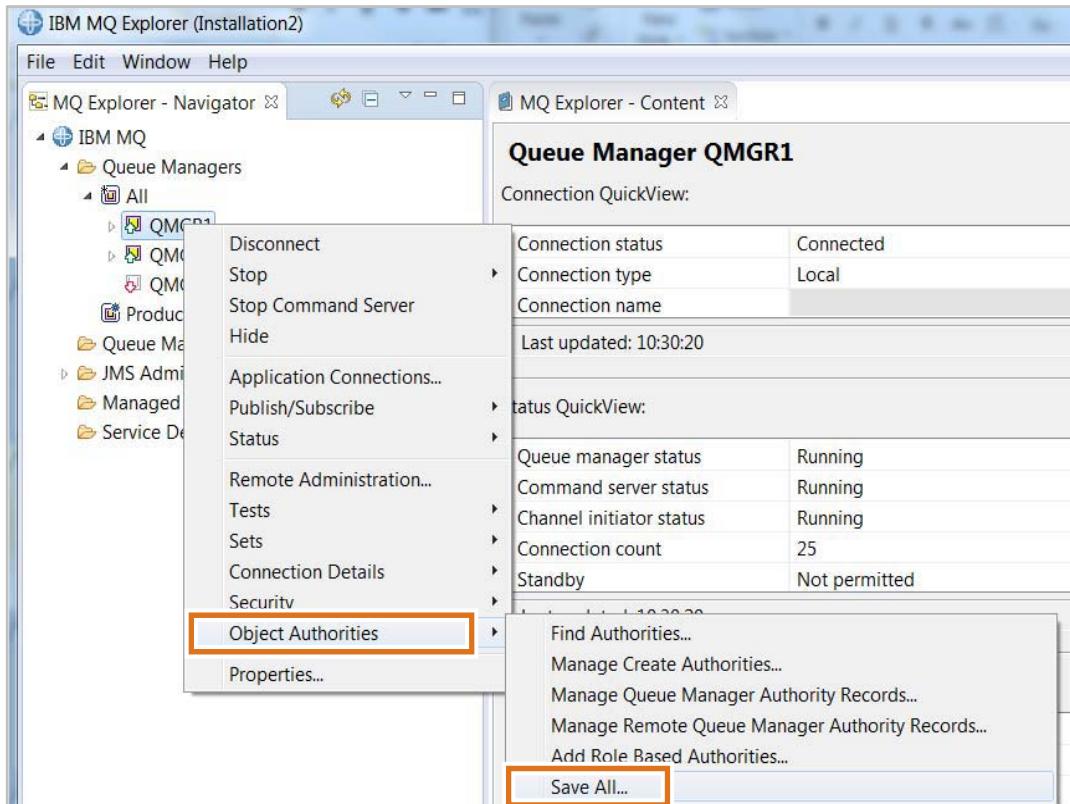
© Copyright IBM Corporation 2017

Figure 11-16. dmpmqaut command

You can use the MQ **dmpmqaut** control command to create a report of the current authorizations that are associated with a specified profile.

This figure describes the syntax for the **dmpmqaut** command.

Backing up security definitions in IBM MQ Explorer



Backing up and restoring IBM MQ messages and object definitions

© Copyright IBM Corporation 2017

Figure 11-17. Backing up security definitions in IBM MQ Explorer

As seen in the figure, the security definitions can be exported from the MQ Explorer.

The security definitions that are exported from MQ Explorer are in a format that is similar to the output from `amqoamd -s` command, which is a series of `setmqaut` commands.

Recovering persistent messages

- To restart, a queue manager requires:
 - Log records that are written since the last checkpoint
 - Log records that are written by transactions that were still active at the time queue manager stopped
- Persistent messages are recovered automatically when queue manager is restarted
- A damaged local queue can be detected only later
 - Reported as "object damaged"
 - Normally must be recovered manually

[Backing up and restoring IBM MQ messages and object definitions](#)

© Copyright IBM Corporation 2017

Figure 11-18. Recovering persistent messages

The queue manager recovers any damaged object that would prevent it from starting, but it would not normally include a local queue that is damaged. Such a queue can be detected later when an attempt is made to access it.

To restart, a queue manager requires:

- Log records that are written since the last checkpoint.
- Log records that are written by transactions that were still active when the queue manager stopped. Uncommitted persistent messages, put or got inside these transactions, are rolled back during restart.

Types of logs

| Circular | Linear |
|---|---|
| Log files are viewed as a closed loop | Log files are viewed as a sequence |
| Amount of disk space that is required for the log does not increase with time | Log file is never deleted but becomes inactive when it contains no entries that are required to restart the queue manager |
| | Can be archived when it becomes inactive |
| | Required for media recovery |

[Backing up and restoring IBM MQ messages and object definitions](#)

© Copyright IBM Corporation 2017

Figure 11-19. Types of logs

MQ records all significant changes to the data controlled by the queue manager in a recovery log. Changes include creating and deleting objects, persistent message updates, transaction states, changes to object attributes, and channel activities.

The log contains the information that is required for recovering all updates to message queues by:

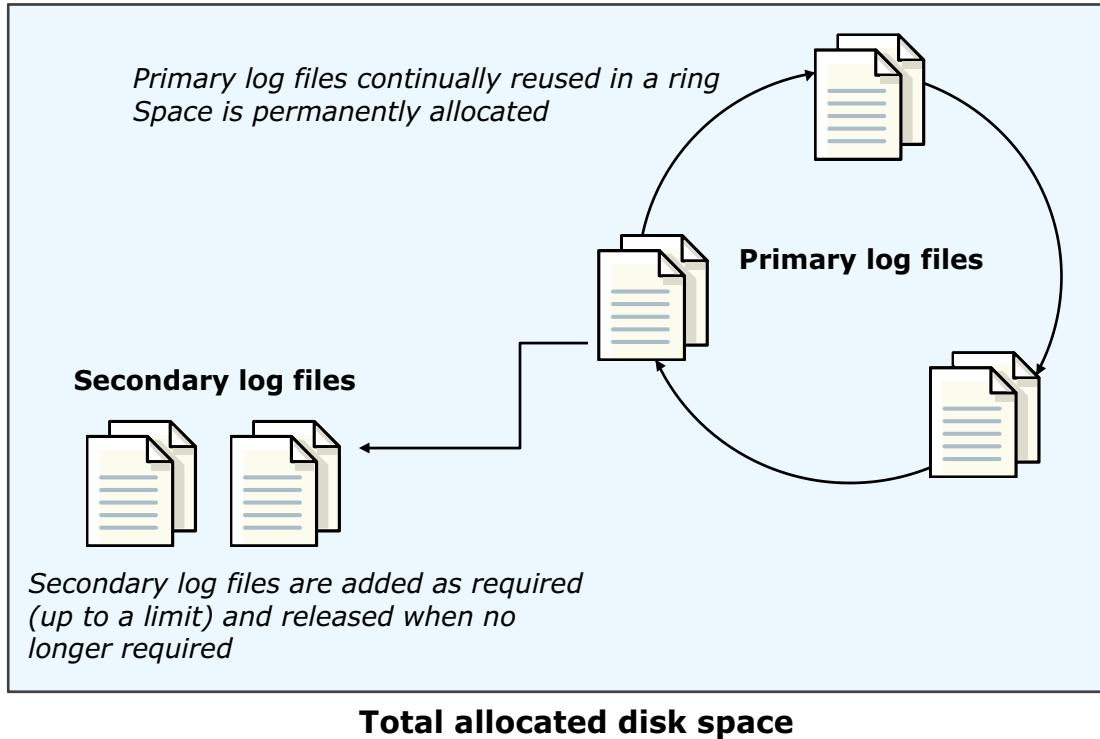
- Keeping records of queue manager changes
- Keeping records of queue updates for use by the restart process
- Allowing you to restore data after a hardware or software failure

The type of logging is selected when a queue manager is created. Unless you request linear logging when you create a queue manager, circular logging is provided by default.

A log checkpoint is taken periodically and provides a point of consistency for the queue manager data. A checkpoint is recorded in the log as a series of checkpoint records. Checkpoints reduce restart time by minimizing the log replay required.

MQ generates checkpoints automatically. They are taken when the queue manager starts, at shutdown, when logging space is running low, and after every 10,000 operations logged.

Circular logs



Backing up and restoring IBM MQ messages and object definitions

© Copyright IBM Corporation 2017

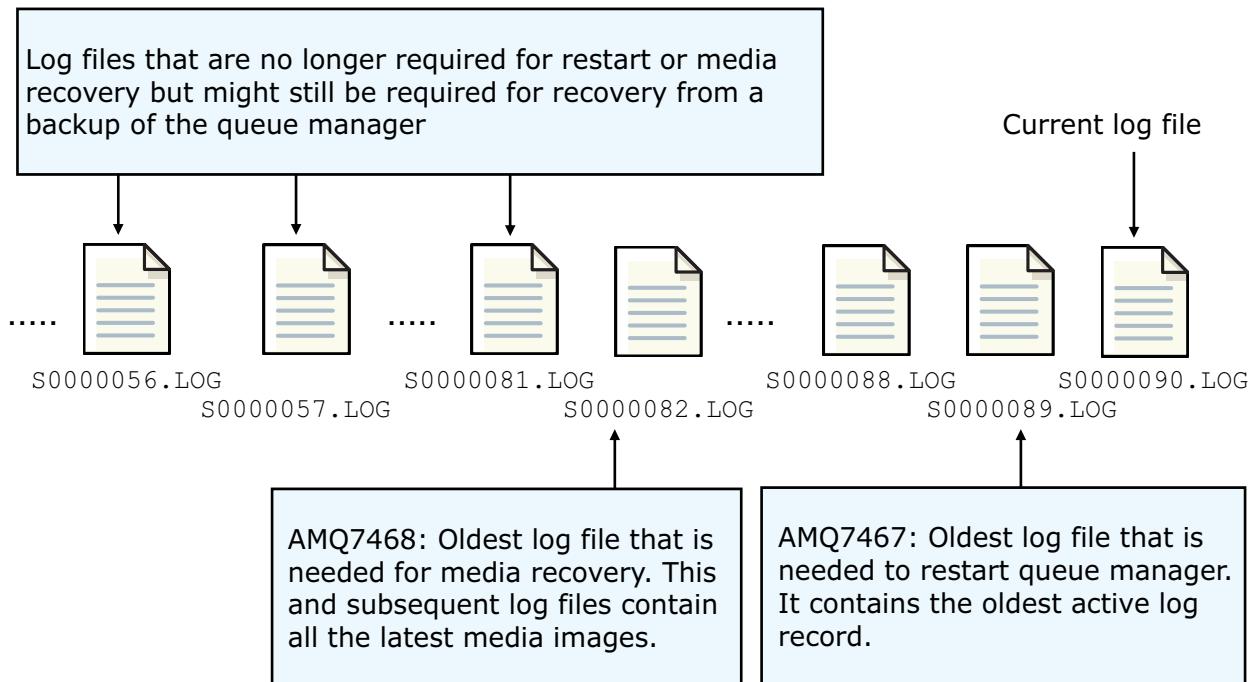
Figure 11-20. Circular logs

With circular logging, the log files are viewed as a closed ring. A log file becomes available for reuse when it contains no active log records. An active log record is one that is still required to restart the queue manager.

With circular logging, MQ can recover messages that follow a system failure but is unable to recover messages that follow a media failure. It has the advantage that the amount of disk space that is required for the log does not increase with time.

When you specify the type of logging, you can also specify the size and location of the log files if you do not want to accept the default values as specified in the MQ configuration file.

Linear logs



Safe implementation of linear logs requires regular oversight and management of the log files

[Backing up and restoring IBM MQ messages and object definitions](#)

© Copyright IBM Corporation 2017

Figure 11-21. Linear logs

With linear logs, the log files are viewed as a sequence. A log file is never deleted, but it becomes inactive when it contains no active log records. New log files are added to the sequence as required. Space is not reused. A linear log can recover from a media failure, but it requires the regular archival of inactive log files.



Important

If the logs are not actively managed, a normally operating linear logged queue manager can fail. Depending on message traffic load, message persistence, excessive MAXDEPTH and MAXMSGL settings, and disk space allocations, the logs use all available space and the queue manager stops unless you take steps to prevent it.

IBM provides several SupportPacs that manage linear log files. Typically, these tools use a scripting language to identify inactive log extents and dispose of them. The queue manager provides commands to inquire on which extents are active, as an option for administrators who want to provide their own instrumentation for this process.

Dumping the log

- Use **dmpmqlog** to dump a formatted version of the log
- Queue manager must be stopped
- By default, the dump starts from the head of the log
- Optionally, log dump can start from:
 - Base of the log
 - Log record that a specified *log sequence number* (LSN) identifies
 - Log file that a specified *extent number* (linear logs only) identifies
- Log records include:
 - Put and get of persistent messages
 - Transaction events
 - Creation, alteration, and deletion of MQ objects

[Backing up and restoring IBM MQ messages and object definitions](#)

© Copyright IBM Corporation 2017

Figure 11-22. Dumping the log

The **dmpmqlog** control command can be used to dump a formatted version of the log. It can be used only when the queue manager is not running.

The head of the log is the checkpoint that starts the active portion of the log. Normally, the head of the log would be the most recent checkpoint. However, if transactions were active when the queue manager stopped and uncommitted persistent messages were inside these transactions before the most recent checkpoint, the head of the log might be positioned at an earlier checkpoint.

- The base of the log is the first log record in the log file that contains the head of the log.
- A unique log sequence number (LSN) identifies each log record.
- Each log file has a file name of the form Snnnnnnn.LOG where nnnnnnnn is the extent number.

Options on the **dmpmqlog** command specify a different starting point for the log. The log includes the following information:

- Header information about the log, for example, whether the log is circular or linear, or the LSN of the log record at the head of the log
- Start queue manager and stop queue manager log records
- Start checkpoint and end checkpoint log records

- Put message and get message log records for persistent messages only and contains hex values for the application data and the message descriptor
- Various types of log records for events that are associated with transactions: for example, start transaction, prepare transaction, and commit transaction
- Log records that are associated with the creation, alteration, and deletion of MQ objects

Damaged objects and media recovery

- IBM MQ objects can be marked as damaged
 - Corrupted data in the queue file
 - Missing queue file
 - Disk failure
- Damaged objects can be deleted
- A damaged object can be re-created from a **linear log**
 - Known as *media recovery*
 - Queue manager records media images automatically at certain times
 - Use control command **rcdmqimg** to record media image of a local queue regularly
- Media recovery
 - Automatic if a damaged object is detected during restart
 - For a local queue, it is normally done by using control command **rcrmqobj**

Backing up and restoring IBM MQ messages and object definitions

© Copyright IBM Corporation 2017

Figure 11-23. Damaged objects and media recovery

A queue manager does not normally detect that a local queue is damaged during restart. If it was storing uncommitted persistent messages that were put or got inside a transaction that was still active when the queue manager stopped, it would detect a damaged local queue. In this case, the queue manager would automatically re-create the local queue as it must be able to roll back the transaction that did not complete. As a result, a damaged local queue is normally detected only when an attempt is made to access it.

Recording media image to a log

- Use `rcdmqimg` command to write an image of an object, or group of objects, to the log for use in media recovery
 - Requires linear logging
 - Moves the log sequence number forward and frees up old log files for archival or deletion
 - Can take a long time to run if queues contain many messages

Example: Record an image of queue manager QMGR1 in the log

```
rcdmqimg -t qmgr -m QMGR1
```

-t Type of object to record
-m Queue manager

The control command to record a media image is `rcdmqimg`.

The object name can have a trailing asterisk to record any objects with names that match the portion of the name before the asterisk.

Re-creating an object from an image

- Use the **rcrmqobj** command to re-create an object, or group of objects, from media images that are contained in the log
 - Can be used with using linear logging only
 - Use on a running queue manager

Examples

- Re-create all local queues for the default queue manager:
`rcrmqobj -t ql *`
- Re-create all remote queues that are associated with queue manager QMGR1:
`rcrmqobj -m QMGR1 -t qr *`

[Backing up and restoring IBM MQ messages and object definitions](#)

© Copyright IBM Corporation 2017

Figure 11-25. Re-creating an object from an image

The **rcrmqobj** control command can be used to re-create a damaged object.



Information

Although you can use the control commands **rcdmqimg** and **rcrmqobj** with other types of MQ objects, the simplest way to re-create such an object is to rerun the MQ command that created it. This advice applies to an alias queue, a model queue, a local definition of a remote queue, a process object, and a channel.

Valid object types for recording and re-creating images

| | |
|-------------------------|--|
| * or all | All object types |
| authinfo | Authentication information object for SSL channel security |
| channel or chl | Channels |
| clntconn or clcn | Client connection channels |
| clchltab | Client channel table |
| listener or lstr | Listener |
| namelist or nl | Namelists |
| process or prcs | Processes |
| queue or q | All types of queue |
| qalias or qa | Alias queues |
| qlocal or ql | Local queues |
| qmodel or qm | Model queues |
| qremote or qr | Remote queues |
| service or srvc | Service |
| syncfile | Channel synchronization file |
| topic or top | Topics |

Backing up and restoring IBM MQ messages and object definitions

© Copyright IBM Corporation 2017

Figure 11-26. Valid object types for recording and re-creating images

When you use a control command to record a media image and re-create objects, you can specify the object type to record and re-create. This figure lists the object types and syntax.

Unit summary

- Describe how IBM MQ uses logging to record significant changes to the data controlled by the queue manager
- Describe the difference between circular and linear logging
- Develop a method for backing up the IBM MQ environment
- Use a media image to recover objects that become damaged
- Save the queue manager object definitions

Backing up and restoring IBM MQ messages and object definitions

© Copyright IBM Corporation 2017

Figure 11-27. Unit summary

Review questions

1. If you want to re-create your security definitions, which command would be the best method to back them up?
 - A. `dmpmqaut`
 - B. `amqoamd -s`
 - C. `bkupaut`
2. True or False: It is acceptable to back up IBM MQ files while the IBM MQ queue managers are running.



Backing up and restoring IBM MQ messages and object definitions

© Copyright IBM Corporation 2017

Figure 11-28. Review questions

Write your answers down here:

- 1.
- 2.

Review answers

1. If you want to re-create your security definitions, which command would be the best method to back them up?
A. dmpmqaut
B. amqoamd -s
C. bkupaut

The answer is B. The output is given as a series of setmqaut commands, which can be used on the command interface to re-create the security definitions.

2. True or False: It is acceptable to back up IBM MQ files while the IBM MQ queue managers are running.
The answer is False. The backup is not consistent because of updates in progress when the files were copied.



Exercise: Using a media image to restore a queue

Backing up and restoring IBM MQ messages and object definitions

© Copyright IBM Corporation 2017

Figure 11-30. Exercise: Using a media image to restore a queue

In this exercise, you capture a media image of a queue, deliberately damage the queue, and then restore it.

Exercise objectives

- Capture an object media image
- Re-create an IBM MQ object from an object media image



Backing up and restoring IBM MQ messages and object definitions

© Copyright IBM Corporation 2017

Figure 11-31. Exercise objectives

See the *Course Exercises Guide* for detailed instructions.

Exercise: Backing up and restoring IBM MQ object definitions

Backing up and restoring IBM MQ messages and object definitions

© Copyright IBM Corporation 2017

Figure 11-32. Exercise: Backing up and restoring IBM MQ object definitions

In this exercise, you use the `dumpmqcfg` command to unload a queue manager's object definitions. You then create a queue manager and load the same definitions, and use MQSC commands or IBM MQ Explorer to show that the definitions are the same.

Exercise objectives

- Use IBM MQ commands to back up object definitions of a queue manager
- Use IBM MQ commands to upload object definitions to another queue manager



[Backing up and restoring IBM MQ messages and object definitions](#)

© Copyright IBM Corporation 2017

Figure 11-33. Exercise objectives

See the *Course Exercises Guide* for detailed instructions.