
Exercise 9. Using a media image to restore a queue

Estimated time

00:30

Overview

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

Objectives

After completing this exercise, you should be able to:

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

Introduction

This exercise demonstrates the survival of persistent messages across a queue manager restart. It also demonstrates how a damaged queue can be recovered from its media image.

In this exercise, you create a queue manager with a linear log. To demonstrate media recovery, you then damage a queue. The act of deliberately damaging a queue should not be done in practice. It is done in this exercise to simulate a disk failure.

You can use the `amqsbcg` sample program or MQ Explorer to browse queues.

Requirements

- IBM MQ and MQ Explorer is installed on the local system

Exercise instructions

Part 1: Queue manager restart

In this part of the exercise, you create a queue manager that uses linear logging. This part of the exercise also demonstrates the survival of persistent messages across a queue manager restart.

- 1. To avoid any conflicts or problems, use MQ Explorer or the `endmqm` command to stop all running queue managers.

To use the `endmqm` command, type:

```
endmqm -i QM01
endmqm -i QMR01
```

- 2. Create a queue manager that is named QML01 that uses linear logging, runs on port 1417, and uses the SYSTEM.DEAD.LETTER.QUEUE for its dead-letter queue.

Type: `crtmqm -l1 -p 1417 -u SYSTEM.DEAD.LETTER.QUEUE QML01`

- 3. Start the queue manager.

Type: `strmqm QML01`

- 4. Use a text editor to create an MQSC command file that is named `recover.mqsc` that defines the following objects.

- a. Define two local queues that are named QL.A and QL.B. Set the default persistence of each queue to YES (persistent).

In the command file, type:

```
DEF QL(QL.A) DEFPSIST(YES) REPLACE
DEF QL(QL.B) DEFPSIST(YES) REPLACE
```

- b. Define two alias queues that are named QA.A and QA.B.

Define QA.A so that it resolves to the local queue QL.A.

Define QA.B so that it resolves to the local queue QL.B.

Set the default persistence of each alias queue to NO (not persistent).

In the command file, type:

```
DEF QA(QA.A) TARGET(QL.A) DEFPSIST(NO) REPLACE
DEF QA(QA.B) TARGET(QL.B) DEFPSIST(NO) REPLACE
```

- 5. Use MQSC to process the command file and redirect the output to a report file.

Make sure that you specify the queue manager name QML01 in the `runmqsc` command. Also, if the current directory does not contain the `recover.mqsc` file, you must specify the path for the `recover.mqsc` file.

Type: `runmqsc QML01 < recover.mqsc > recoverreport.txt`

Review the report file to verify that all commands in the file ran successfully.

- 6. Use the `amqspput` sample program to put persistent and non-persistent messages on the local queues QL.A and QL.B and alias queues QA.A and QA.B.

When the `amqspput` sample program starts, the value of the **DefPersistence** attribute of the queue determines whether the message that is put on the queue is persistent or non-persistent. Because of the queue definitions that you made in step 4, the `amqspput` sample program puts persistent messages on the local queues and non-persistent messages on the alias queues.

Type:

```
amqspput QL.A QML01
amqspput QA.A QML01
amqspput QL.B QML01
amqspput QA.B QML01
```

- 7. Use the MQ Explorer or the sample program `amqsbcg` to browse the messages on the local queue QL.A and QL.B and verify that it contains a mixture of persistent and non-persistent messages.

To use the sample program, type: `amqsbcg QL.A QML01`

The sample program returns a record for each message that is on the queue.

- If the message is persistent, the **Persistence** attribute is 1.
- If the message is not persistent, the **Persistence** attribute is 0.

To use MQ Explorer, right-click the queue in the **Queues** content view and then click **Browse Messages**. A row exists for each message on the queue. Scroll to the right until you see the **Persistence** column and verify that the queue contains both persistent and non-persistent messages.

- 8. Stop and then restart the queue manager QML01.

Type:

```
endmqm -i QML01
strmqm QML01
```

- 9. Use MQ Explorer or the sample program `amqsbcg` to browse the messages on QL.A and QL.B again. Verify that only the persistent messages on the local queue survived the restart.

Part 2: Media recovery

- 1. Run the `rcdmqimg` command against the queue QL.B to write an image of the queue to the log for use in media recovery.
Type: `rcdmqimg -m QML01 -t ql QL.B`
- 2. Locate the file that implements local queue QL.B within the file system and then damage the queue by deleting the file. The file location is specific to the operating system.



Linux

```
rm /var/mqm/qmgrs/QML01/queues/QL\!B
```



Windows

```
erase "C:\ProgramData\IBM\MQ\qmgrs\QML01\queues\QL!B"
```

- ___ 3. Use the **DISPLAY QUEUE** command in **runmqsc** to display the attributes of local queue QL.B. This command still works because it does not yet recognize that the queue is damaged.

Type:

```
runmqsc QML01
DIS Q(QL.B)
END
```

- ___ 4. Use **amqsput** to put some messages to local queue QL.B. It might take 10 or more messages (because of buffering) but the queue manager eventually detects that the queue is damaged and reports the damage by returning MQRC 2101 (MQRC_OBJECT_DAMAGED).

Type: **amqsput QL.B QML01**

- ___ 5. Use the **DISPLAY QUEUE** command in **runmqsc** to display the attributes of local queue QL.B.

Type:

```
runmqsc QML01
DIS Q(QL.B)
end
```

The queue manager now recognizes that a queue is damaged and reports that the MQ object QL.B is not found.

- ___ 6. Use the re-create object (**rcremqobj**) command to recover local queue QL.B from its media image.

Type: **rcremqobj -m QML01 -t ql QL.B**

You should see a message that indicates that the local queue QL.B was re-created.

- ___ 7. In **runmqsc**, check that you can now display the attributes of the queue. Type:

```
runmqsc QML01
DIS Q(QL.B)
end
```

- ___ 8. Use MQ Explorer or the **amqsbcg** sample program to browse the local queue QL.B to check whether the messages are recovered successfully.

To use the sample program to browse the queue, type: **amqsbcg QL.B QML01**

To browse the messages by using MQ Explorer, right-click the QL.B queue in the **Queues** content view and then click **Browse Messages**.

Were the non-persistent messages recovered?

Exercise cleanup

- ___ 1. Clear the messages from QL.A and QL.B by using a sample program or MQ Explorer.
To clear the messages by using a sample program, type: `amqsget QL.B QML01`
To clear the messages by using MQ Explorer, right-click the queue in the **Queues** content view, click **Clear Messages**, and then click **Clear**.
- ___ 2. Stop and then delete the queue manager QML01. It might take 1 or 2 minutes for the queue manager to be deleted.

Type:

```
endmqm -i QML01  
dltmqm QML01
```

End of exercise

Exercise review and wrap-up

Having completed this exercise, you should be able to:

- Create a queue manager with linear logging
- Put persistent and non-persistent messages on local queues
- Stop and restart a queue manager to verify that persistent messages are maintained during a queue manager restart
- Record a media image
- Recover a damaged local queue from its media image

Exercise 10. Backing up and restoring IBM MQ object definitions

Estimated time

00:30

Overview

In this exercise, you use the **dmpmqcfg** 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.

Objectives

After completing this exercise, you should be able to:

- 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

Introduction

The IBM MQ **dmpmqcfg** control command can be used to back up and restore a queue manager's configuration. As an option, you can use the **-o setmqaut** parameter to create **setmqaut** commands that can be used to re-create the queue manager security definitions.

Requirements

- IBM MQ and IBM MQ Explorer
- The queue manager QM01 that was created in Exercise 1

Exercise instructions

Part 1: Backing up queue manager configuration

- ___ 1. Display the status of the queue manager QM01 to verify that it is running.

In a command window, type: `dspmq -m QM01`

If the queue manager is not running, start it now.

- ___ 2. Run the MQ `dmpmqcfg` command to back up QM01.

Use the default formatting option of MQSC and all attributes. Use standard output redirection to store the definitions into a file.

```
dmpmqcfg -m QM01 -a > QM01.mqsc
```

- ___ 3. Using a text editor, examine the resulting file, and answer the following questions.

- ___ a. Verify that SYSTEM.* objects appear in the backup file. Can you suppress generation of the SYSTEM.* objects?
- ___ b. Is the REPLACE option present in the generated definitions?
- ___ c. What does REPLACE do?

Part 2: Restoring the object definitions

In this part of the exercise, you simulate the running of the generated backup file. This action would be done if you needed to redefine all your MQ objects. In this exercise, you are not replacing any definitions because some of the SYSTEM queues are not empty. Redefining the SYSTEM queue objects effectively removes all cluster, channel state, and authorities that are stored in the SYSTEM queues.

- ___ 1. Use MQSC to check whether any of the SYSTEM queues on QM01 contain messages.

In MQSC, type:

```
DIS QL(SYSTEM.*) WHERE(CURDEPTH GT 0)
```

- ___ 2. Use the `runmqsc` command with the `-v` option to run in verify mode and simulate the restore function without replacing the definitions. Redirect the output to a file so that you can verify the success or failure of each command.

```
runmqsc -v QM01 < QM01.mqsc > QM01.out
```



Important

Ensure that you use the `-v` option so that you do not overwrite the existing configuration in the lab environment.

- ___ 3. Use a text editor to examine the output in `QM01.out` file that you created in Step 2.

Were there any errors? If yes, find and correct the error, and then rerun the command with the `-v` switch.

- ___ 4. Use an MQSC command to verify that messages on the SYSTEM queues on QM01 are intact.

In MQSC, type:

```
DIS QL(SYSTEM.* ) WHERE(CURDEPTH GT 0)
```

Ensure that the queue depths match the queue depths reported in Part 2, Step 1.

Part 3: Dumping security definitions

In this part of the exercise, you use an MQ command and the MQ Explorer to write security definitions to a file.

- ___ 1. Use the MQ `dmpmqcfg` command with the `-x authrec` option to write the queue manager security definitions to a file. Use the `-o setmqaut` option to format the output as a series of `setmqaut` commands.

```
dmpmqcfg -o setmqaut -x authrec -t qmgr -m QM01 > QM01-setmqaut.mqsc
```

- ___ 2. Use MQ Explorer to write the OAM definitions for QM01.

- ___ a. In the MQ Explorer Navigator, right-click QM01 and then click **Object Authorities > Save All**.

- ___ b. On Windows, select a directory, enter a file name, and then click **Save**.

On Linux, select a directory, enter a file name, and then click **OK**.

- ___ 3. Examine the output from the security definitions file.

Find the `setmqaut` entries for QL.A.

- ___ 4. Run MQ command: `dmpmqaut -m QM01 -n QL.A -t q`

This command writes a formatted view of the authorities for the queue QL.A.

- ___ 5. Examine the output from the `dmpmqaut` command and compare it with the output from the `dmpmqcfg` command.

The `dmpmqcfg` command writes the security definitions as a list of `setmqaut` commands. These commands can then be used to re-create the security definitions on a new queue manager.

In contrast, `dmpmqaut` writes a formatted view of the authorities. The formatted view is useful for finding all the authorities that are set against a particular MQ object. The output cannot be used to re-create the security definitions without reformatting.

End of exercise

Exercise review and wrap-up

Having completed this exercise, you should be able to:

- 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

Exercise solutions

Part 1, Step 3: Using a text editor, examine the resulting file, and answer the following questions.

- a. Verify that SYSTEM.* objects appear in the backup file. Can you suppress generation of the SYSTEM.* objects?

Answer: **No**.

- b. Is the REPLACE option present in the generated definitions?

Answer: **Yes**.

- c. What does REPLACE do?

Answer: **The REPLACE and NOREPLACE options control whether any existing definitions are replaced with the new ones. The MQ default is NOREPLACE.**

Part 2 and Part 3: Sample solution files are provided in the lab files `Lab10\solution` subdirectory.