Exercise 1. Using commands to create a queue manager and queues

Estimated time

00:30

Overview

In this exercise, you use IBM MQ commands to create a queue manager, start it, and then create queues. You also create and run an IBM MQ script command file.

Objectives

After completing this exercise, you should be able to:

- Use IBM MQ commands to create a local queue manager, local queues, and alias queues
- Use IBM MQ commands to display and alter queue manager and queue attributes
- Create and run an IBM MQ command file

Introduction

In this exercise, you create and start a queue manager. You then create local queues and alter queue and queue manager attributes.

You also create a script file of IBM MQ commands and run the command file from a command.

Finally, you create alias queues and alter the queue attributes.

Requirements

- IBM MQ is installed on the system
- A text editor

Important

The exercises in this course use a set of lab files that might include scripts, applications, files, solution files, PI files, and others. The course lab files can be found in the `C:/labfiles` directory. The exercises point you to the lab files as you need them.

Exercise instructions

Part 1: Create and start a queue manager

__ 1. Log on to the VMware image or system.

Windows

Log in as `Administrator` with a password of `passw0rd`

Linux

Log in as `localuser` with a password of `passw0rd`

__ 2. Open a command (terminal) window.

__ 3. Create a queue manager that is named: **`QM01`**
Use this queue manager for all steps in this exercise.

Note: The queue manager and dead-letter queue names are case-sensitive.

Windows

Type: **`crtmqm -u SYSTEM.DEAD.LETTER.QUEUE QM01`**

Linux

Create the queue manager with user-based authorizations that are enabled by adding the **`-oa user`** option to the create queue manager command. Type:

```
crtmqm -oa user -u SYSTEM.DEAD.LETTER.QUEUE QM01
```

This queue manager (QM01) is used in a security exercise later in this course (Exercise 8) that assumes that user-based authorizations are enabled on the Linux queue manager. This option and topic are described in more detail in Unit 10.

You should see a message that the queue manager and a queue manager directory were created successfully.

__ 4. Start the queue manager.

Type: **`strmqm QM01`**

You should see a message that the queue manager QM01 started by using V9.0.0.0.

Part 2: Use the IBM MQ MQSC commands interactively

In this part of the exercise, you enter all the IBM MQ commands interactively and display the results in the command window.

__ 1. Use the control command **runmqsc** to start MQSC mode and complete the following tasks.

Type: **runmqsc QM01**

You see message that the MQSC for the queue manager QM01 is starting. You are in MQSC mode until you enter **END**. No command prompt is displayed in MQSC mode.

__ a. Display all the attributes of the queue manager.

Type: **DISPLAY QMGR**

Note: **QMGR** is a literal string and is not replaced with your queue manager name.

__ b. Verify that the DEADQ attribute is set to: SYSTEM.DEAD.LETTER.QUEUE

__ c. Display all the queues with queue names that begin with the characters **SYSTEM**.

Type: **DISPLAY Q(SYSTEM*)**

__ d. Create a local queue that is named **QL.A** with a text description of “QL.A TEXT”.

Type: **DEFINE QL(QL.A) REPLACE DESCR('QL.A TEXT')**

__ e. Display all the attributes of the local queue that is name QL.A and verify that the text that is entered in the previous step is the value for the DESC attribute.

Type: **DISPLAY QL(QL.A)**

__ f. Change the maximum number of messages that are allowed on the local queue QL.A (MAXDEPTH) from 5000 messages to 1000 messages.

Type: **ALTER QL(QL.A) MAXDEPTH(1000)**

__ g. Display the queue attributes for QL.A and verify the change to the MAXDEPTH attribute.

Type: **DISPLAY QL(QL.A)**

Note

You can use the Up arrow key in the command window to display and rerun a previous command.

Verify that the ALTER command changed only the attribute that you specified on the command. You should see that the MAXDEPTH attribute was modified and that the DESC attribute is unchanged.

__ h. Define another local queue that is named **QL.B** with a text description.

Type: **DEFINE QL(QL.B) REPLACE DESCR('QL.B Text')**

__ i. Display all the attributes of the queue and verify that the text appears in the DESC attribute.

Type: **DISPLAY QL(QL.B)**

__ j. Change the maximum number of messages that are allowed on the queue to 2000 by using the DEFINE command with REPLACE instead of the ALTER command.

Type: **DEFINE QL(QL.B) REPLACE MAXDEPTH(2000)**

__ k. Display the queue attributes of QL.B.

Type: **DISPLAY QL(QL.B)**

The DEFINE command with REPLACE changes the attributes that you specify on the command and sets the other attributes to their initial values. You should see that the MAXDEPTH attribute was modified and that the DESC attribute is now blank.

__ 2. Exit the **runmqsc** mode.

Type: **END**

Part 3: Use the IBM MQ MQSC commands in a command file

__ 1. Using a text editor such as Notepad on Windows or gedit on Linux, prepare an IBM MQ command file with the same commands that you entered interactively in Part 2 of this exercise. Name the file **CreateQs.mqsc** and save it in your lab files directory.

Windows

The lab files directory is: **C:\labfiles**

Linux

The lab files directory is: **/home/localuser/labfiles**

In the text file, enter the following commands and then save the file.

```
DIS QMGR
DIS Q(SYSTEM*)
DEF QL(QL.A) REPLACE DESCR('QL.A Text')
DIS QL(QL.A)
ALTER QL(QL.A) MAXDEPTH(1000)
DIS QL(QL.A)
DEF QL(QL.B) REPLACE DESCR('QL.B Text')
DEF QL(QL.B) REPLACE MAXDEPTH(2000)
DIS QL(QL.B)
```

__ 2. Process the command file and direct the results to a file that is named **report.txt**.

__ a. In the command window, change directories to the lab files directory that contains the MQSC file that you created in Step 1.

__ b. Run the command file and redirect the output to report.txt.

Type: **runmqsc QM01 < CreateQs.mqsc > report.txt**

__ 3. Using a text editor, open the file that is named **report.txt** and verify that each command was successful.

Part 4: Work with alias queues and queue attributes

In this part of the exercise, you create alias queues and modify queue attributes. You use these queues in Exercise 3.

__ 1. Start MQSC mode for the queue manager QM01.

Type: **runmqsc QM01**

__ 2. Verify the local queues QL.A and QL.B exist.

Type: **DIS QL(QL*)**

__ 3. Create an alias queue on QM01 that is named **QA.A** that resolves to the local queue QL.A.

Type: **DEF QA(QA.A) TARGET(QL.A)**

__ 4. Display the attributes of the alias queue QA.A and verify that queue TYPE is **QALIAS** and that TARGET is **QL.A**.

Type: **DIS QA(QA.A)**

__ 5. Modify the alias queue QA.A to inhibit PUT requests.

Type: **ALTER QA(QA.A) PUT(DISABLED)**

__ 6. Display the attributes of the alias queue QA.A and verify that queue PUT attribute is now set to **DISABLED**.

__ 7. Modify the local queue QL.B to inhibit PUT requests.

Type: **ALTER QL(QL.B) PUT(DISABLED)**

__ 8. Display the attributes of the local queue QL.B and verify that queue PUT attribute is now set to **DISABLED**.

__ 9. Create an alias queue that is named **QA.B** that resolves to the local queue QL.B.

Type: **DEF QA(QA.B) TARGET(QL.B)**

__ 10. Display the attributes of the alias queue QA.B and verify that queue TYPE is **QALIAS** and that TARGET is **QL.B**.

__ 11. Exit **runmqsc** mode.

Type: **END**

End of exercise

Exercise review and wrap-up

You should now be able to:

- Create and start a queue manager
- Use control commands and MQSC command files to create and modify local queues
- Display attributes of IBM MQ objects
- Change queue attributes and create alias queues

Exercise 2. Using IBM MQ Explorer to create queue managers and queues

Estimated time

00:30

Overview

In this exercise, you use IBM MQ Explorer to create a queue manager, start it, and then create queues. You also use IBM MQ Explorer to create queue manager sets to simplify the management of many queue managers.

Objectives

After completing this exercise, you should be able to:

- Use IBM MQ Explorer to create a local queue manager, local queues, and alias queues
- Use IBM MQ Explorer to display and modify queue manager and queue properties
- Use IBM MQ Explorer to create a queue manager set

Introduction

In this exercise, you use IBM MQ Explorer to create a queue manager. Then, you create local queues and modify queue properties. Finally, you create queue manager sets to simplify the management of multiple queue managers.

Requirements

- IBM MQ and IBM MQ Explorer are installed on the system
- A text editor

Exercise instructions

Part 1: Create a queue manager

__ 1. Start IBM MQ Explorer.

Windows

From the Windows **Start** menu, click **MQ Explorer (Installation 1)** or click the IBM MQ Explorer icon in the taskbar.

Linux

In a terminal window, type: **`MQExplorer`**

When you start IBM MQ Explorer by using this command, it runs in the foreground. You must keep this terminal window open while you are using IBM MQ Explorer.

When it is necessary to open a new terminal window for running sample programs, for example, right-click and then click **New Terminal**.

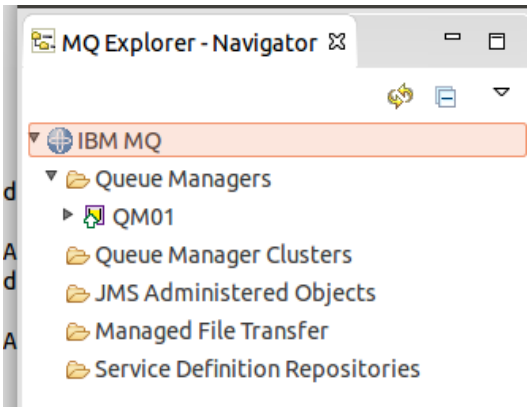
__ 2. IBM MQ Explorer automatically recognizes and manages any queue managers that are defined on the same system.

If you completed Exercise 1, verify that IBM MQ Explorer found the queue manager QM01.

__ a. In the **MQ Explorer - Navigator** view, expand **IBM MQ**.

__ b. Expand the **Queue Managers** folder.

__ c. Verify that the queue manager QM01 is listed under the **Queue Managers** folder.



__ 3. Create a queue manager that is named **QM02** that uses a dead-letter queue that is named **QM02.DLQ** and listens on port **1416**. Use this queue manager for all the remaining steps in this exercise.

__ a. In the **MQ Explorer - Navigator** view, right-click **Queue Managers** and then click **New > Queue Manager**.

__ b. For the **Queue Manager name**, type: **QM02**
For the **Dead-letter queue**, type: **QM02.DLQ**
Click **Next**.

__ c. Click **Next** to accept the default values for logging.

__ d. Click **Next** to accept the default values for the configuration options.

__ e. On the listener options page, enter **1416** for the **Listen on port number**. Click **Finish**.

__ 4. Verify that the queue manager **QM02** now appears under the **Queue Managers** folder in the **MQ Explorer - Navigator** view.

__ 5. Verify that the queue manager QM02 is running.

The status icon next to queue name should contain an up pointing green arrow. You can also click the queue manager QM02 in the **Queue Managers** folder to display and verify the queue manager status.

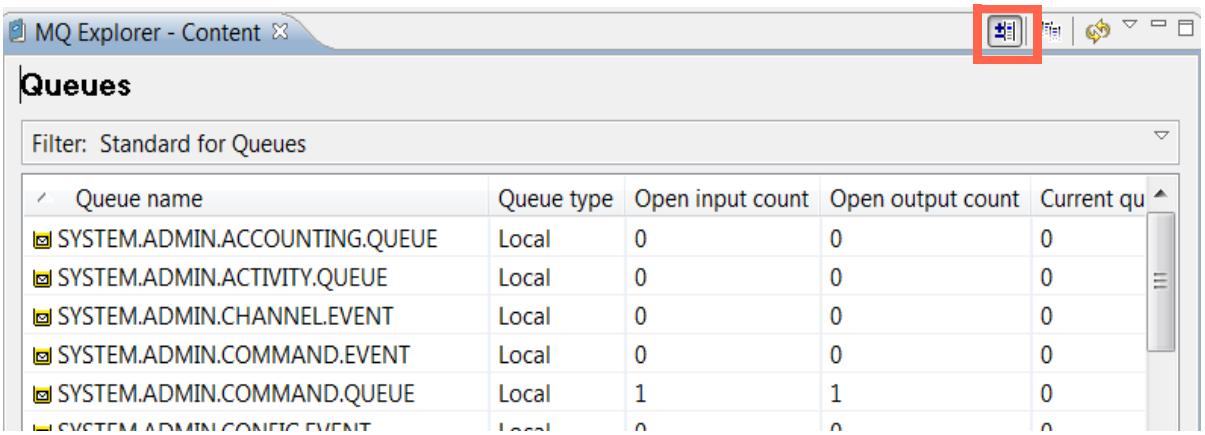
Part 2: Create and modify local queues

__ 1. Display the SYSTEM queues for QM02.

__ a. Expand **Queue Managers > QM02** in the **MQ Explorer - Navigator** view.

__ b. Click **Queues** under **QM02**.

__ c. In the **Queues** content view, click the **Show System Objects** icon.



__ 2. The **Show System Queues** icon is a toggle. In the **Queues** view, click the **Show System Objects** icon again to hide the SYSTEM queues.

__ 3. Create a local queue that is named **QM02.DLQ**. You identified this queue as the dead-letter queue for QM02 when you created the queue manager.

__ a. Right-click **Queues** under **QM02** in the **MQ Explorer - Navigator** view and then click **New > Local Queue**.

__ b. For the queue **Name**, type **QM02.DLQ** and then click **Finish**.

__ c. If a confirmation window is displayed, click **OK**.

__ d. Verify that the queue now appears in the **Queues** content view.

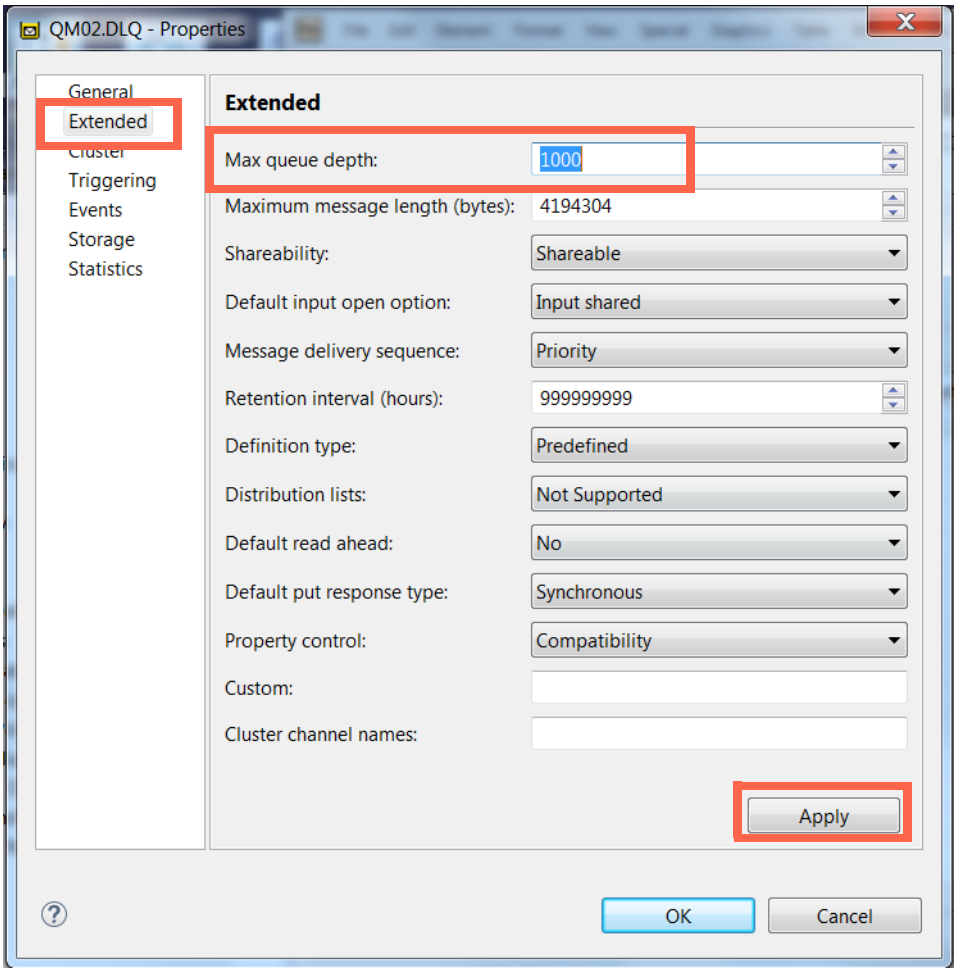
__ 4. Change the maximum number of messages that are allowed on the queue QM02.DLQ to 1000.

__ a. In the **Queues** view for QM02, right-click **QM02.DLQ** and then click **Properties**.

__ b. Click **Extended**.

__ c. Change the **Max queue depth** property value to **1000**.

__ d. Click **Apply** and then click **OK**.



__ 5. Create another local queue on QM02 that is named **QL.B**.

__ a. Right-click **Queues** under QM02 in the **MQ Explorer - Navigator** view and then click **New > Local Queue**.

__ b. For the queue **Name**, type **QL.B** and then click **Finish**.

__ c. If a confirmation window is displayed, click **OK**.

__ d. Verify that queue QL.B is now listed in the **Queues** content view for QM02.

__ 6. Change the default persistence on the local queue QL.B to **Persistent**.

__ a. In the **Queues** view for QM02, right-click **QL.B** and then click **Properties**.

__ b. On the **General** page, change the **Default persistence** property to **Persistent**.

__ c. Click **Apply** and then click **OK**.

__ d. In the **Queues** view, scroll to the right and verify that the **Default persistence** column now shows **Persistent** for QL.B.

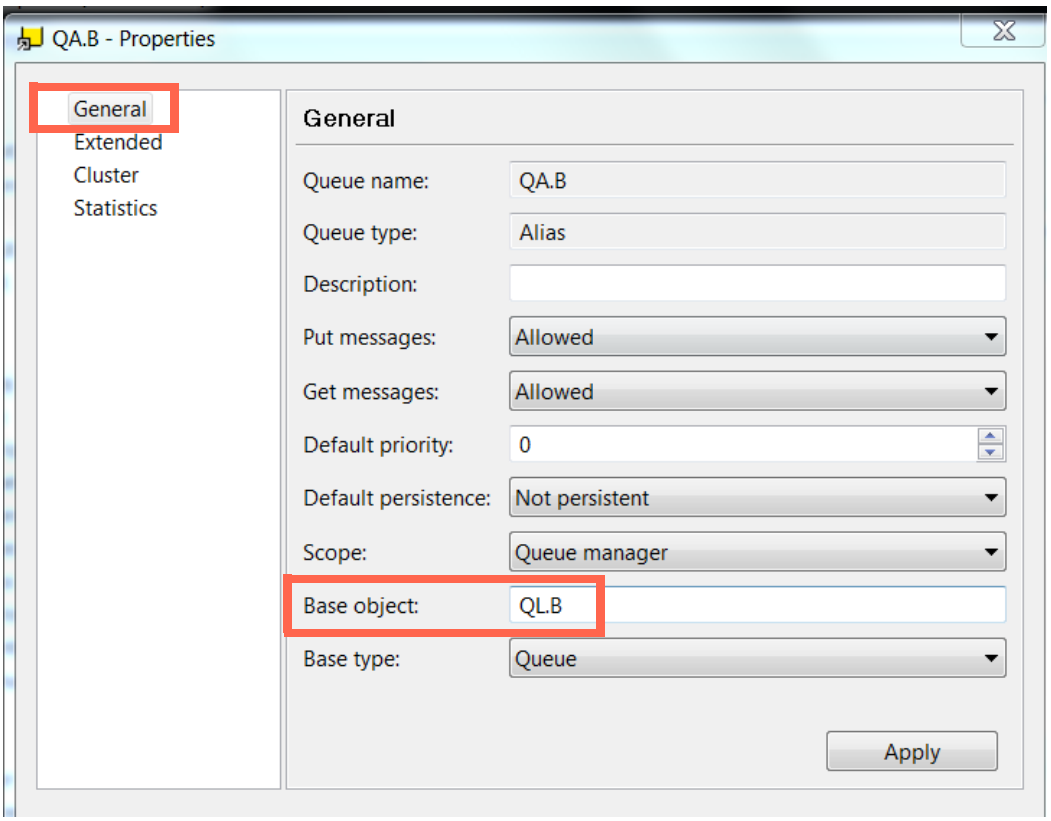
__ 7. Create an alias queue on QM02 that is named **QA.B** that resolves to the local queue QL.B.

__ a. Right-click **Queues** under QM02 in the **MQ Explorer - Navigator** view and then click **New > Alias Queue**.

__ b. For the queue **Name**, type **`QA.B`** and then click **Next**.

__ c. On the **General** page, type **`QL.B`** for the **Base object** property and then click **Finish**.

__ d. If a confirmation window is displayed, click **OK**.



__ 8. Inhibit PUT requests on the QA.B alias queue on QM02.

__ a. In the **Queues** view for QM02, right-click **QA.B** and then click **Properties**.

__ b. On the **General** page, change the **Put messages** property to **Inhibited**.

__ c. Click **Apply** and then click **OK**.

__ d. Verify that the **Put messages** column in the **Queues** view now contains **Inhibited** for the queue QA.B.

Part 3: Create queue manager sets

In this part of the exercise, you create a queue manager set that is named **Production**. You then create a custom filter that uses the value in the queue manager **Description** field to identify "Production" queue managers to include in the new set.

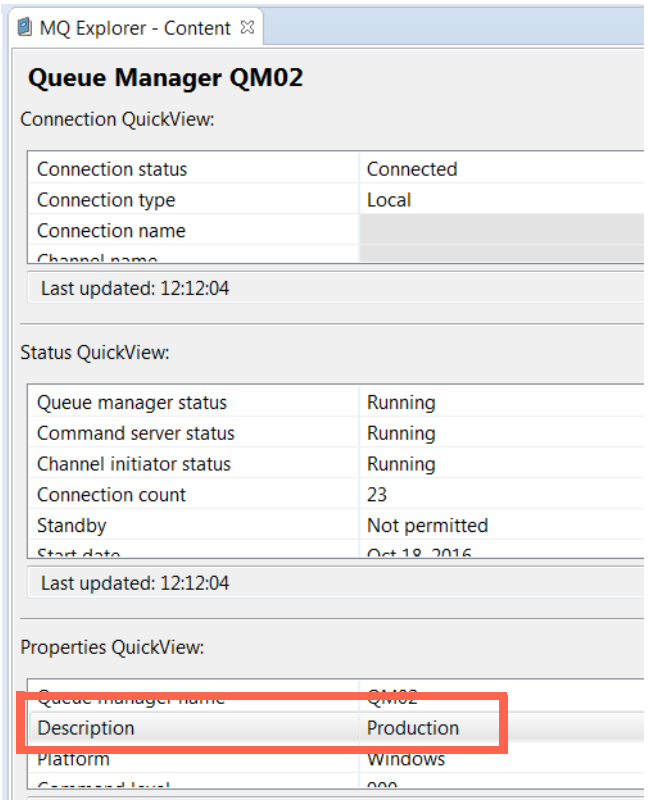
__ 1. Change the description of the QM02 queue manager to **Production**.

__ a. In the **MQ Explorer - Navigator** view, right-click **QM02** and then click **Properties**.

__ b. On the **General** properties page, type **`Production`** in the **Description** field.

__ c. Click **Apply** and then click **OK**.

__ d. In the **Queue Manager QM02** view, verify that the **Description** property is now set to **Production**.



__ 2. Create a queue manager set that is named **Production** and automatically add any queue managers with the **Description** property set to **`Production`** to the queue manager set.

__ a. Right-click **Queue Managers** in the **MQ Explorer-Navigator** view, and then click **Sets > New Set**.

__ b. For the set **Name**, type **`Production`** and then select **Automatic** for the set type. Click **Next**.

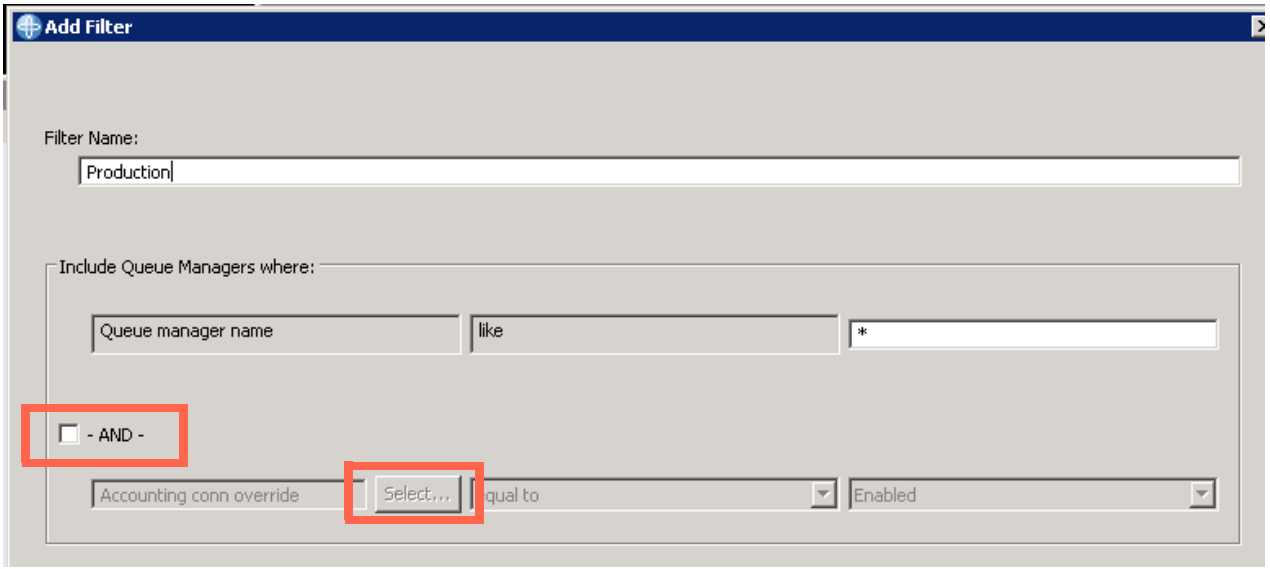
__ c. Click **Manage Filters** to define a new filter that references the queue manager **Description** property.

__ d. Click **Add** in the **Manage Filters** window.

__ e. In the **Add Filter** window, type **`Production`** for the **Filter Name**.

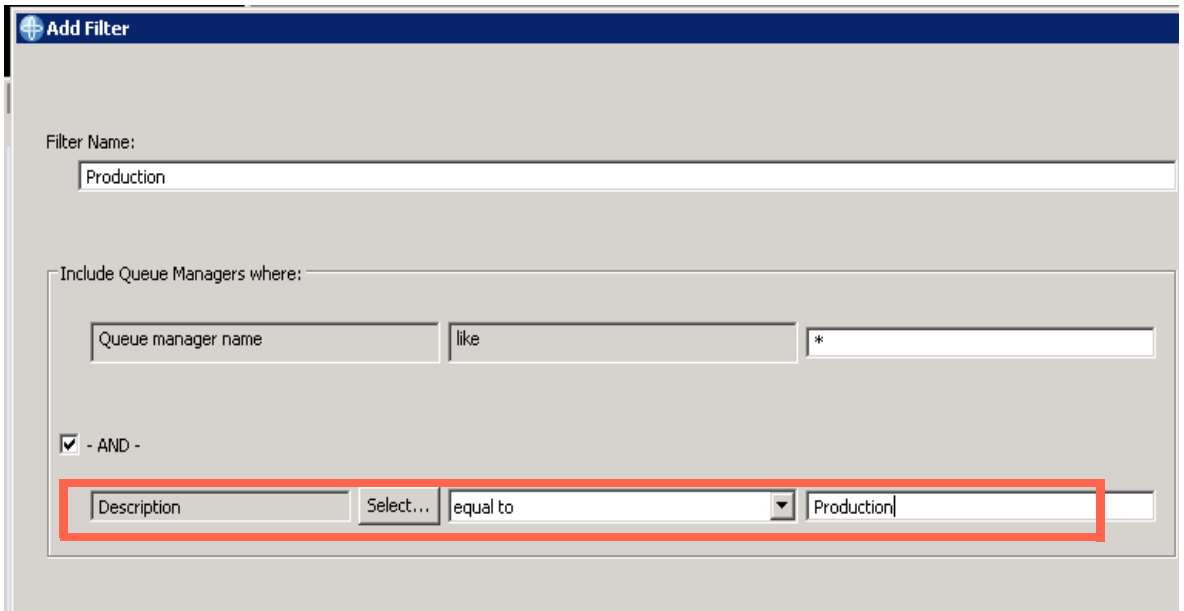
__ f. The first part of the filter currently evaluates all queue managers. This filter should evaluate all queue managers, so no changes are required to the first filter evaluation.

__ g. Add an AND clause to the filter that tests for the text **`Production`** in the queue manager **Description** property. Select the **AND** check box, and then click **Select**.



__ h. In the **Select Attribute** window, scroll down, select the **Description** attribute, and then click **OK**.

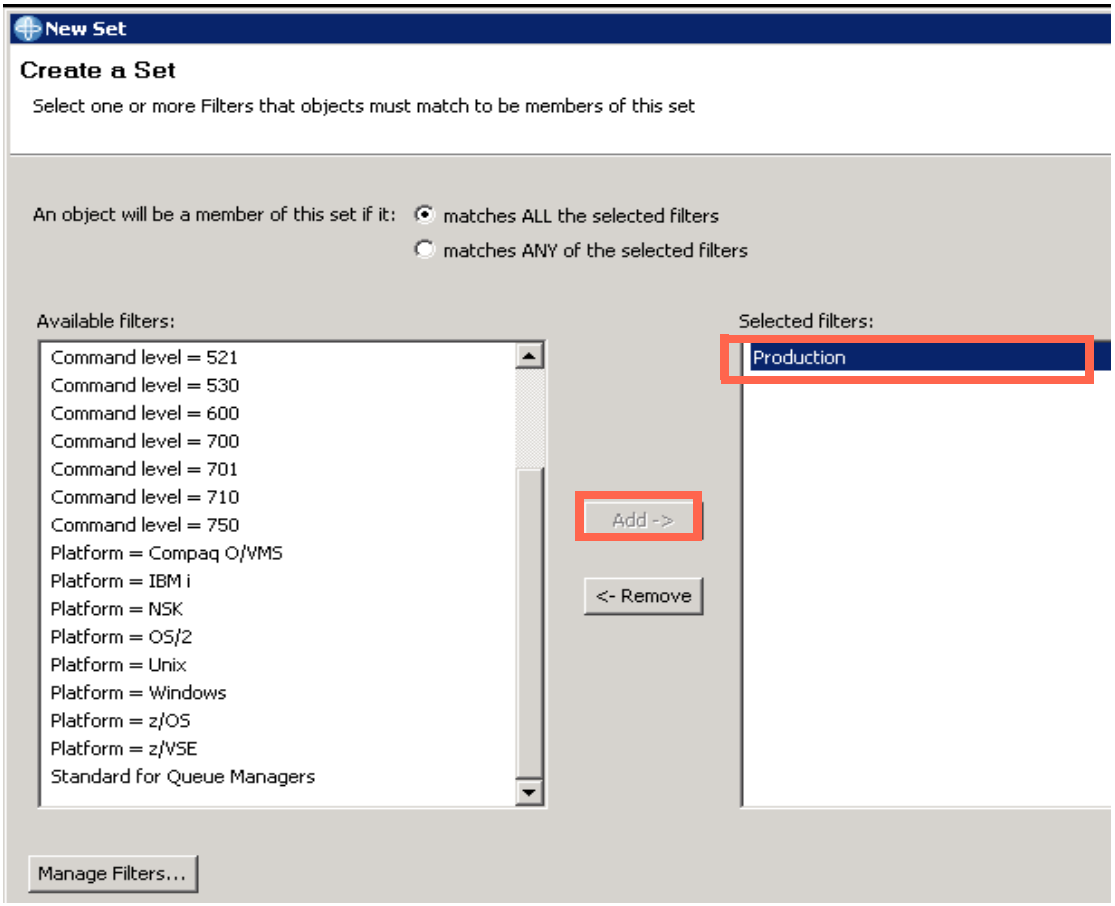
__ i. Complete the AND statement in the filter definition by setting the evaluation option to **equal to** and entering: **`Production`**



__ j. Click **OK** on the **Add Filter** window.

__ k. Click **OK** in the **Manage Filters** window.

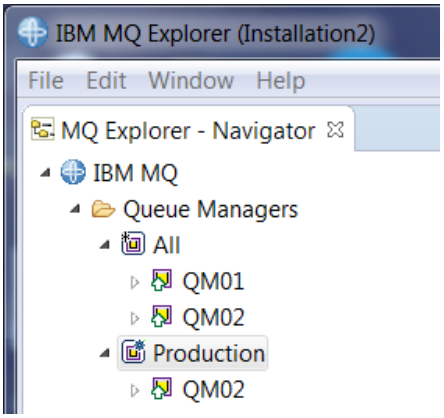
__ l. In the **New Set** window, scroll down the **Available filters** list and then select **Production** from the list of available filters. Click **Add** to move **Production** to the **Selected filters** pane.



__ m. Click **Finish**.

__ 3. Verify that the **Queue Managers** folder in the **MQ Explorer - Navigator** view now contains two sets: **All** and **Production**.

__ 4. Verify that the queue manager QM02 is a member of the **Production** set.



Exercise cleanup

By using MQ Explorer or MQSC, stop the queue manager QM02.

- To stop the queue manager by using MQ Explorer, right-click the queue manager in the **MQ Explorer - Navigator** view and then click **Stop > Controlled**.
- To stop the queue manager by using a command, type: **endmqm -c QM01**

End of exercise

Exercise review and wrap-up

You should now be able to use IBM MQ Explorer to:

- Create a queue manager and modify queue manager properties
- Create local and alias queues and modify queue properties
- Create a queue manager set that uses a custom filter to automatically assign it to the set

Exercise 3. Using IBM MQ sample programs to test the configuration

Estimated time

00:30

Overview

In this exercise, you use the IBM MQ sample programs to put, get, and browse queues and test the queue manager configuration. You then use IBM MQ Explorer and IBM MQ commands to verify the actions of the sample programs. You also define and test an alias queue.

Objectives

After completing this exercise, you should be able to:

- Use IBM MQ sample programs to put messages onto a queue, browse messages on a queue, and get messages from a queue
- Use IBM MQ Explorer and IBM MQ commands to display queue contents
- Define and test an alias queue that refers to another queue

Introduction

In this exercise, you test the queue manager QM01 and queues that you defined in Exercise 1 by using the MQ sample programs and MQ Explorer to put, get, and browse messages.

Linux

The sample programs are in the **`/opt/mqm/samp/bin`** subdirectory. The path to this directory is included in the **`/etc/environment`** file on the lab image for this course.

Windows

The sample programs are in **`C:\Program Files\IBM\MQ\Tools\c\Samples\Bin64`**. The PATH statement is already set on the Windows lab image.

The IBM MQ sample programs are run in a command window. The **`amqsput`**, **`amqsget`**, and **`amqsbcg`** sample programs that are used in this exercise accept two parameters. The first parameter is required and is the name of a queue. The second parameter is optional and is the name of a queue

manager. If the second parameter is omitted, the default queue manager is assumed. Both parameters are case-sensitive.

A brief description of each program is provided.

- The **amqsput** program connects to the queue manager and opens the queue. It reads lines of text from the standard input device, generates a message from each line of text, and puts the messages on the named queue. After the program reads a null line or the EOF character from the standard input device, it closes the queue, and disconnects from the queue manager.
  The command to start the program is: `amqsput` *`QName QmgrName`*

  Example: **amqsput Q1 QM01**

- The **amqsget** program connects to the queue manager, opens the queue for input, and gets all the messages from the queue. It writes the text within the message to the standard output device, waits 15 seconds (60 seconds if a message does not exist at the start) in case any more messages are put on the queue. The program then closes the queue and disconnects from the queue manager.
  The command to start the program is: `amqsget` *`QName QmgrName`*

  Example: **amqsget Q2 QM01**

- The **amqsbcg** program connects to the queue manager and opens the queue for browsing. It browses all the messages on the queue and writes their contents, in both hexadecimal and character format, to the standard output device. It also returns, in a readable format, the fields in the message descriptor for each message. It then closes the queue and disconnects from the queue manager.
  The command to start the program is: `amqsbcg` *`QName QmgrName`*

  Example: **amqsbcg Q3 QM01**

Requirements

- IBM MQ and IBM MQ Explorer
- Queue manager QM01 and queues that were created in Exercise 1
- The IBM MQ sample programs **amqsput**, **amqsbcg**, and **amqsget**
- A text editor

Exercise instructions

Part 1: Make QM01 is the default queue manager

The default queue manager processes any commands for which a queue manager name is not explicitly specified. In this part of the exercise, you make QM01 the default queue manager so that you do not have to specify the queue manager name when you run the IBM MQ sample programs.

__ 1. Open the MQ Explorer unless it is already open.

__ 2. Right-click **IBM MQ**, and then select **Properties**. The properties for MQ are displayed.

__ 3. In the **Default queue manager** name, type: **QM01**.

__ 4. Click **Apply** and then click **OK**.

Part 2: Put, get, and browse messages

In this part of the exercise, you use the queues that you created in Exercise 1 and the IBM MQ sample programs to put, get, and browse messages.

__ 1. Use a sample program to put two messages on the local queue QL.A on QM01.

__ a. In a terminal window, type: **amqsput QL.A**

The program responds with:

```
Sample AMQSPUT0 start
target queue is QL.A
```

__ b. Enter some data and then press Enter. Each line of data that is entered becomes the data portion of a new MQ message.

__ c. To end the sample program, press Enter on a blank line.

Troubleshooting

If the program does not run successfully, an MQ reason code is returned.

To interpret the reason code, use the control command `mqrc` at the **C:** or **$** prompt followed by the four-digit reason code. For example, type: `mqrc 2085`

__ 2. Use MQ Explorer to display the queue contents and browse the messages.

__ a. Select the **Queue** folder under QM01 in the **MQ Explorer - Navigator** view to display the **Queues** view.

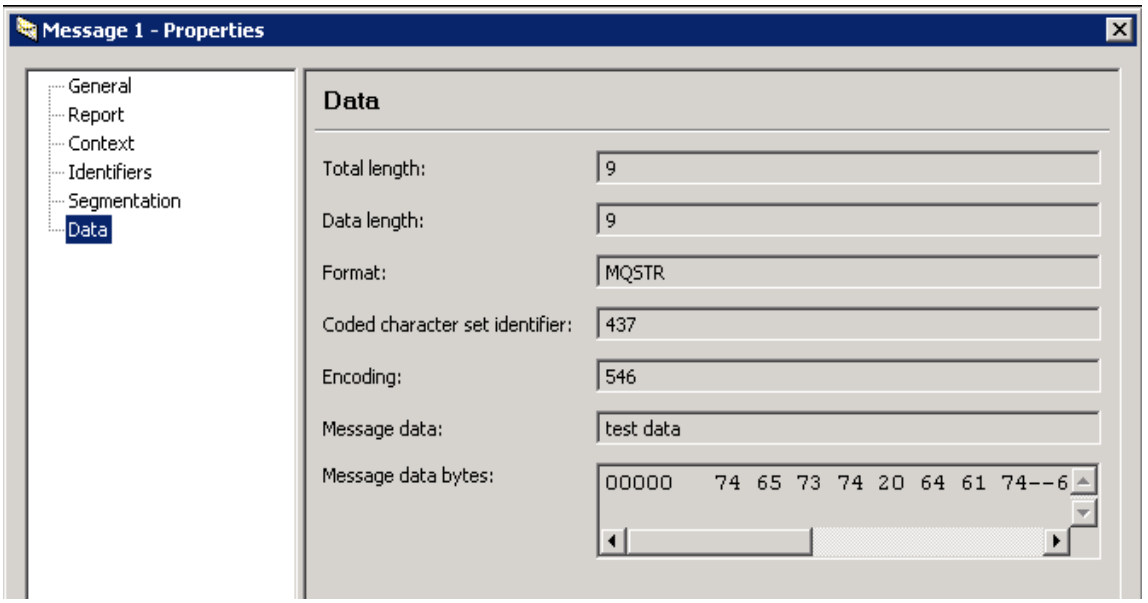
__ b. Right-click **QL.A** in the **Queues** view for QM01 and then click **Browse messages**. The **Message browser** list contains one message for every message that you entered with the **amqsput** sample program.

__ c. Double-click a message from the list of messages to display its properties.

__ d. Click the **Data** properties tab to view the message data.

__ e. Click **Close** to close the **Properties** window.

__ f. Click **Close** to close the message browser window.



__ 3. Use a sample program to browse the messages on the queue QL.A. Direct the command results to a file and view the results.

In the terminal window, type: **amqsbcg QL.A > out.txt**

If the program does not run successfully, can you determine the reason for the failure?

__ 4. Use a sample program to get the messages from the queue and empty the queue.

Type: **amqsget QL.A**

The **amqsget** sample program can take a few seconds to run. It is finished when you see the message:

```
no more messages
Sample AMQGET0 end
```

If the program does not run successfully, can you determine the reason for the failure?

__ 5. Use the **amqsput** sample program to put three messages on the queue QL.A.

__ 6. Use an MSQC command to show the current depth (CURDEPTH) of QL.A (the number of messages on the queue) and verify that the messages are on the queue.

__ a. Open a new terminal window.

__ b. Type: **runmqsc QM01**

__ c. Type: **DIS Q(QL.A) CURDEPTH**

__ 7. Define a new local queue on QM01 that is named **QL.X** with the attributes of the local queue QL.A.

In the MQSC window, type: **DEF QL(QL.X) LIKE(QL.A)**

__ 8. Use a sample program or MQ Explorer to put some messages on QL.X.

To use the sample program, type **amqsput QL.X** and then enter some messages.

To use MQ Explorer, right-click QL.X in the **Queues** view and then click **Put Test Message**.

__ 9. Clear the messages from the local queue QL.A.

In the window that is running MQSC, type: **CLEAR QL(QL.A)**

__ 10. Try to clear the messages from the alias queue QA.A.

In the MQSC window, type: **CLEAR QA(QA.A)**

You should receive a syntax error that indicates that the CLEAR command is valid for local queues and topic strings only and not alias queues (QALIAS).

You cannot clear messages from an alias queue because it is a pointer to another queue and does not hold messages.

__ 11. Delete the local queue QL.X.

In the MQSC window, type: **DELETE QL(QL.X)**

You should see that you cannot delete the queue because it is not empty.

To delete the queue and the messages, type:

DELETE QL(QL.X) PURGE

Part 3: Work with alias queues

In this part of the exercise, you use the alias queues that you created on QM01 in Exercise 1.

In Exercise 1, you created an alias queue that is named QA.A that points to the local queue QL.A and an alias queue that is named QA.B that points to local queue QL.B.

__ 1. Using MQ Explorer or MQSC, determine whether the **PUT messages** attribute is set to **allowed** (enabled) or **inhibited** (disabled) for the alias queue QA.A.

To use MQ Explorer, click the **Queues** folder under QM01 to display the **Queues** view. The **Put messages** column shows the current state of the PUT attribute.

To use MQSC to display the PUT attribute, type: **DIS QA(QA.A)**

__ 2. If the PUT requests on the alias queue QA.A are allowed (enabled), change the queue PUT attribute to inhibit them.

In window that is running MQSC, type: **ALTER QA(QA.A) PUT(DISABLED)**

__ 3. Use MQ Explorer or MQSC to determine whether the **PUT messages** attribute is set to **allowed** (enabled) or **inhibited** (disabled) for the local queue QL.B.

If the PUT requests on the local queue QL.B are allowed, change the queue PUT attribute to inhibit them.

In the window that is running MQSC, type: **ALTER QL(QL.B) PUT(DISABLED)**

__ 4. Use the **amqsput** sample program to try to put messages on both the alias and local queues.

Run the sample program in the command window.

- For QL.A, type: **amqsput QL.A**
- For QA.A, type: **amqsput QA.A**
- For QL.B, type: **amqsput QL.B**
- For QA.B, type: **amqsput QA.B**

The PUT request for QL.A succeeds but the other requests fail with reason code 2051 because PUT is inhibited.

QA.A and QL.B fail because PUT is inhibited on those queues.

QA.B fails because it is an alias queue for the local queue QL.B, which has PUT inhibited.

Exercise cleanup

__ 1. Change the local queue QL.B to allow PUT requests.

In the MQSC window, type: **`ALTER QL(QL.B) PUT(ENABLED)`**

__ 2. Change the alias queue QA.A to allow PUT requests.

In the MQSC window, type: **`ALTER QA(QA.A) PUT(ENABLED)`**

__ 3. Use MQ Explorer to clear any messages on the local queues QL.A and QL.B.

__ a. In the **Queues** view, right-click QL.A and then click **Clear Messages**.

__ b. Select **Clear will be used using CLEAR command** and then click **CLEAR**.

__ c. Click **OK** in the verification window.

__ d. Verify that the **Current queue depth** column value for QL.A is zero.

__ e. If any messages are on QL.B, repeat steps a – d to clear QL.B.

Note

You must use MQGET instead of CLEAR to empty a queue if:

- Uncommitted messages are on the queue and they were put on the queue under sync point
- An application currently has the queue open

End of exercise

Exercise review and wrap-up

You should now be able to:

- Use MQ sample programs to put messages onto a queue, browse messages on a queue, and get messages from a queue
- Use MQ Explorer and MQSC to display queue contents
- Use MQSC to change a queue to inhibit or allow PUTs
- Use MQ Explorer and MQSC clear and delete queues