

Course Exercises Guide

IBM MQ V9.1 System Administration

Course code WM156 / ZM156 ERC 1.1



April 2020 edition

Notices

This information was developed for products and services offered in the US.

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, MD-NC119
Armonk, NY 10504-1785
United States of America*

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 jurisdictions 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 websites are provided for convenience only and do not in any manner serve as an endorsement of those websites. The materials at those websites are not part of the materials for this IBM product and use of those websites is at your own risk.

IBM may use or distribute any of the information you provide in any way it believes appropriate without incurring any obligation to you.

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.

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 actual people or business enterprises is entirely coincidental.

Trademarks

IBM, the IBM logo, and ibm.com are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the web at "Copyright and trademark information" at www.ibm.com/legal/copytrade.shtml.

© Copyright International Business Machines Corporation 2020.

This document may not be reproduced in whole or in part without the prior written permission of IBM.

US Government Users Restricted Rights - Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

Contents

Trademarks	vii
Exercises description	viii
Exercise objectives	viii
User IDs and passwords	viii
How to follow the exercise instructions	ix
Exercise 1. Getting started with IBM MQ	1-1
Section 1. Explore your installation	1-2
Section 2. Testing the installation	1-5
2.1. Create a queue manager	1-5
2.2. Create a queue	1-7
2.3. Send and retrieve a test message	1-8
Section 3. Setting up the mqweb server for IBM MQ Console	1-10
3.1. Configure basic security for users and groups to access the IBM MQ Console	1-10
3.2. Start the mqweb server	1-11
3.3. Start IBM MQ Console	1-12
3.4. Customize the tab name	1-13
3.5. Create a queue manager	1-14
3.6. Add a widget	1-15
3.7. Delete a queue manager	1-18
3.8. Exercise cleanup	1-18
Exercise 2. Working with IBM MQ administration tools	2-1
Section 1. Create and start a queue manager with IBM MQ control commands	2-2
Section 2. Using MQSC commands	2-3
2.1. Using MQSC commands interactively	2-3
2.2. Running MQSC commands from a text file	2-5
Section 3. Modify local queues	2-7
Section 4. Working with IBM MQ sample programs	2-10
4.1. Putting messages: amqsput	2-10
4.2. Browsing messages: amqsbcg	2-11
4.3. Getting messages: amqsget	2-11
4.4. Manage messages in MQSC mode	2-12
4.5. Clear messages and delete queues	2-13
Section 5. Create queue manager sets	2-14
5.1. Create the production queue managers	2-14
5.2. Define a new queue manager set	2-15
5.3. Create a set by using a filter	2-17
Section 6. Deleting queue manager sets	2-21
Section 7. Exercise cleanup	2-22
Exercise 3. Implementing distributed queuing	3-1
Section 1. Creating channels between queue managers	3-2
1.1. Create the transmission queue	3-2
1.2. Create the sender channel	3-4
1.3. Create the remote queue definition	3-5
1.4. Create the target queue on QMQA1	3-7
1.5. Create the receiver channel on QMQA1	3-8
Section 2. Testing queues in a distributed environment	3-10

2.1. Send some test messages	3-10
2.2. Verify delivery of the messages	3-10
2.3. Start the channel	3-11
2.4. Confirm transmission of the message	3-12
Section 3. Exercise cleanup	3-13
Exercise 4. Connecting an IBM MQ client	4-1
Section 1. Set up the lab environment	4-2
1.1. Configuring the queue manager to act as a server	4-2
1.2. Monitor channel status on IBM MQ Console	4-3
Section 2. Method 1: Client connection through the CCDT	4-5
2.1. Set up a client connection with MQCCDTURL	4-5
2.2. Test the connection	4-5
Section 3. Method 2: Client connection by using the MQSERVER environment variable	4-8
Section 4. Method 3: Client connection by using the client configuration file	4-10
4.1. Edit the mqclient.ini client configuration file	4-10
4.2. Test the connection	4-10
4.3. Exercise cleanup	4-11
Exercise 5. Implementing a basic cluster	5-1
Section 1. Defining the cluster	5-3
1.1. Create the queue managers for the cluster	5-3
1.2. Create the queue manager cluster	5-3
1.3. Add queue managers to the cluster	5-5
1.4. Create cluster queues	5-7
Section 2. Managing the cluster workload	5-9
2.1. Testing the cluster	5-9
2.2. Using a round-robin approach for workload balancing	5-9
2.3. Controlling workload with channel and queue rank	5-11
2.4. Controlling workload with channel priority	5-14
2.5. Using channel weight to control workload	5-15
2.6. Exercise cleanup	5-16
Exercise 6. Configuring publish/subscribe message queuing	6-1
Section 1. Setting up a publish/subscribe cluster	6-2
1.1. Add a partial repository to the cluster	6-2
1.2. Create a topic	6-4
Section 2. Testing the publish/subscribe cluster	6-6
Section 3. Testing cluster publication routing	6-8
3.1. Create a test subscription	6-8
3.2. Validate the proxy subscriptions	6-9
Section 4. Clustered publish/subscribe with topic host routing	6-10
4.1. Test topic-host routing	6-10
4.2. Exercise cleanup	6-11
Exercise 7. Controlling access to IBM MQ	7-1
Section 1. Point-to-point security	7-2
1.1. Set up the lab environment	7-2
1.2. Test user permissions	7-2
1.3. Grant authorization to the queue manager	7-3
1.4. Grant authorization to the queue	7-6
1.5. Validate oamlabuser permissions	7-8
Section 2. Publish and subscribe security	7-10
2.1. Review of publish/subscribe security	7-10
2.2. Validate the permissions	7-11
2.3. Exercise cleanup	7-15

Exercise 8. Securing channels with TLS	8-1
Section 1. Set up the environment	8-2
1.1. Create the queue managers	8-2
1.2. Create the SECQM1 objects	8-3
1.3. Create the SECQM2 objects	8-5
Section 2. Eavesdropping on the connection	8-6
Section 3. Setting up the key repository	8-8
3.1. Creating certificates	8-8
3.2. Update the queue managers with the key database files	8-9
3.3. Create self-signed certificates for the queue managers	8-10
3.4. Update the SSL certificate label properties on the queue managers	8-13
3.5. Exchange certificates	8-14
3.6. Defining your TLS channels	8-16
3.7. Test the TLS channels	8-17
Section 4. TLS with IBM MQ clients	8-20
4.1. Set up the client and server channels	8-20
4.2. Test without TLS	8-21
4.3. Create a keystore for the client connection	8-22
4.4. Test with TLS	8-23
4.5. Test distinguished name matching	8-25
4.6. Exercise cleanup	8-27
Exercise 9. Implementing connection authentication	9-1
Section 1. Setting up permissions	9-2
1.1. Set up the environment	9-2
1.2. Defining user ID and password connection authentication	9-4
Section 2. Checking locally bound connections	9-7
2.1. Testing with the Check locally bound connections option set to None	9-7
2.2. Testing with the Check locally bound connections option set to Optional	9-8
2.3. Testing with the Check locally bound connections option set to Required	9-8
Section 3. Checking client connections (MQ client bindings)	9-11
3.1. Exercise cleanup	9-13
Exercise 10. Running an IBM MQ trace	10-1
Section 1. Starting a trace	10-2
1.1. Start the trace	10-2
1.2. Use the sample programs to put and get messages from a queue	10-3
Section 2. Viewing the trace files	10-5
Section 3. Using the dspmqrt application	10-6
3.1. Creating the queue manager and connections	10-6
3.2. Test the channel definitions	10-7
3.3. Running dspmqrt	10-8
Section 4. Working with the dead-letter queue	10-11
4.1. Viewing messages on the dead-letter queue	10-11
4.2. Using the dead-letter queue handler	10-12
4.3. Exercise cleanup	10-13
Exercise 11. Using a media image to restore a queue	11-1
Section 1. Restarting the queue manager	11-2
1.1. Create a queue manager, local queues, and alias queues	11-2
1.2. Test the queues	11-3
Section 2. Recovering media	11-4
2.1. Damage your queue	11-4
2.2. Re-create your queue	11-4
2.3. Exercise cleanup	11-5

Exercise 12. Backing up and restoring IBM MQ object definitions	12-1
Section 1. Backing up and restoring queue manager configuration	12-2
1.1. Create a backup	12-2
1.2. Restoring the object definitions	12-2
Section 2. Dumping security definitions	12-4
2.1. Write security definitions to a file	12-4
2.2. Exercise cleanup	12-4
Exercise 13. Monitoring IBM MQ for performance	13-1
Section 1. Collecting statistics data	13-3
1.1. Enable statistics monitoring	13-3
1.2. Define the queues for statistics	13-4
1.3. Put some messages to the queue to generate statistics	13-4
1.4. Review the statistics data	13-5
1.5. Disable statistics monitoring	13-5
Section 2. Collecting accounting data	13-6
2.1. Enable accounting monitoring	13-6
2.2. Generate and view accounting data	13-6
2.3. Disable accounting monitoring	13-7
Section 3. Online monitoring	13-8
3.1. Enable online monitoring and generate data	13-8
3.2. Disable online monitoring	13-9
Section 4. Resource monitoring	13-10
4.1. Get a queue status summary	13-10
4.2. Exercise cleanup	13-13
Exercise 14. Monitoring resources with the IBM MQ Console	14-1
Section 1. Configuring IBM MQ Console to monitor MQ objects	14-2
1.1. Sign in to IBM MQ Console	14-2
1.2. Configure the dashboard	14-2
Section 2. Testing security with the basic registry	14-7
2.1. Review the permissions for administrative users	14-7
2.2. Export the dashboard	14-8
2.3. Test mqreader user permissions for MQ objects	14-9
Appendix A. Installing IBM MQ V9.1.0.1	A-1

Trademarks

The reader should recognize that the following terms, which appear in the content of this training document, are official trademarks of IBM or other companies:

IBM, the IBM logo, and ibm.com are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide.

The following are trademarks of International Business Machines Corporation, registered in many jurisdictions worldwide:

AIX®

First Failure Support
Technology™

PureApplication®

Redbooks®

WebSphere®

z/OS®

Intel, Intel Xeon and Xeon are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.

Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both.

Windows is a trademark of Microsoft Corporation in the United States, other countries, or both.

Java™ and all Java-based trademarks and logos are trademarks or registered trademarks of Oracle and/or its affiliates.

UNIX is a registered trademark of The Open Group in the United States and other countries.

VMware is a registered trademark or trademark of VMware, Inc. or its subsidiaries in the United States and/or other jurisdictions.

Other product and service names might be trademarks of IBM or other companies.

Exercises description

Exercise objectives

After completing the exercises, students should be able to:

- Create queue managers and queues
- Use IBM MQ sample programs to test the configuration
- Connect an IBM MQ client
- Implement point-to-point messaging
- Implement a basic cluster
- Implement publish/subscribe messaging
- Manage IBM MQ security by implementing authentication, authorization, and encryption
- Troubleshoot messaging issues
- Back up and restore IBM MQ object definitions
- Monitor IBM MQ performance

In the exercise instructions, you can check off the line before each step as you complete it to track your progress.

Most exercises include required sections, which should always be completed. It might be necessary to complete these sections before you can start later exercises. If you have sufficient time and want an extra challenge, some exercises might also include optional sections that you can complete.



Important

The exercises in this course use a set of lab files that might include scripts, applications, files, and solution files. The course lab files can be found in the following directory:

C:\labfiles

The exercises point you to the lab files as you need them.

User IDs and passwords

The following table lists user ID and password information for this course.

Entry point	User ID	Password
VMware image	administrator	passw0rd
Windows 2012 Server	administrator	passw0rd
IBM MQ Console	mqadmin	mqadmin

IBM MQ Console	mqreader	mqreader
User authentication	mquser	passw0rd
User authentication	oamlabuser	passw0rd
IBM Key Management	--	mqpass

How to follow the exercise instructions

Exercise structure

Each exercise is divided into sections with a series of numbered steps and lettered substeps:

- The numbered steps (1, 2, 3) represent actions to be done.
- The lettered substeps (a, b, c) provide detailed guidance on how to complete the action.



Information

If you already understand how to do the action in the numbered step, you can skip the specific guidance in the lettered substeps.

The following example comes from Exercise 1 of this course.



Example

Excerpt from Exercise 1

- 1. Use the Create Queue Manager wizard to create a queue manager named **QM01**.
 - a. In the **Queue manager name** field, enter: QM01
 - b. In the **Dead-letter queue** field, enter: SYSTEM.DEAD.LETTER.QUEUE
 - c. Click **Finish**.

In this example, the numbered instructions prompt you to create a queue manager. The “a” and “b” substeps provide specific guidance on how to create the queue manager.

Text highlighting in exercises

Different text styles indicate various elements in the exercises.

Words that are highlighted in **bold** represent GUI items that you interact with, such as:

- Menu items
- Field names
- Icons

Words that are highlighted with a `fixed font` include the following items:

- Text that you type or enter as a value
- System messages
- Directory paths
- Code

Tracking your progress

As shown in the example step, you can see that an underscore precedes each numbered step and lettered substep.

You are encouraged to use these markers to track your progress by checking off each step as you complete it. Tracking your progress in this manner might be useful if you are interrupted while working on an exercise.

Required exercise sections

Most exercises include required sections that should always be completed. It might be necessary to complete these sections before you can start subsequent exercises.

Dependencies between exercises are listed in the exercise introduction.



Important

Online course material updates might exist for this course. To check for updates, see the Instructor wiki at <http://ibm.biz/CloudEduCourses>.

Exercise 1. Getting started with IBM MQ

Estimated time

01:00

Overview

In this exercise, you explore your IBM MQ installation. To validate your installation, you create a queue manager and queue, and test messaging. You also set up basic security to use IBM MQ console administration tools.

Objectives

After completing this exercise, you should be able to:

- Explore your IBM MQ installation by creating a queue and testing messaging
- Validate your installation by creating a queue and testing messaging
- Configure the mqweb server for user access to IBM MQ Console

Introduction

This exercise includes these sections:

- [Section 1, "Explore your installation"](#)
- [Section 2, "Testing the installation"](#)
- [Section 3, "Setting up the mqweb server for IBM MQ Console"](#)

Requirements

For this exercise, IBM MQ V9.1 must be installed.

Section 1. Explore your installation

- 1. If you did not yet sign in to the VMware image provided for your lab environment, sign in with these credentials.
 - User name: Administrator
 - Password: passw0rd
- 2. Check your IBM MQ installation information by opening a command prompt and typing:
dspmqver

```
cmd Select Administrator: Command Prompt
Microsoft Windows [Version 10.0.14393]
(c) 2016 Microsoft Corporation. All rights reserved.

C:\Users\Administrator>dspmqver
Name: IBM MQ
Version: 9.1.0.1
Level: p910-001-181108
BuildType: IKAP - (Production)
Platform: IBM MQ for Windows (x64 platform)
Mode: 64-bit
O/S: Windows Server 2016 Server Standard Edition, Build 14393
InstName: Installation1
InstDesc:
Primary: Yes
InstPath: C:\Program Files\IBM\MQ
DataPath: C:\ProgramData\IBM\MQ
MaxCmdLevel: 910
LicenseType: Production
```

- 3. Notice the values for InstPath and DataPath to see the IBM MQ installation path and the path for IBM MQ program data.

```
cmd Select Administrator: Command Prompt
Microsoft Windows [Version 10.0.14393]
(c) 2016 Microsoft Corporation. All rights reserved.

C:\Users\Administrator>dspmqver
Name: IBM MQ
Version: 9.1.0.1
Level: p910-001-181108
BuildType: IKAP - (Production)
Platform: IBM MQ for Windows (x64 platform)
Mode: 64-bit
O/S: Windows Server 2016 Server Standard Edition, Build 14393
InstName: Installation1
InstDesc:
Primary: Yes
InstPath: C:\Program Files\IBM\MQ
DataPath: C:\ProgramData\IBM\MQ
MaxCmdLevel: 910
LicenseType: Production
```

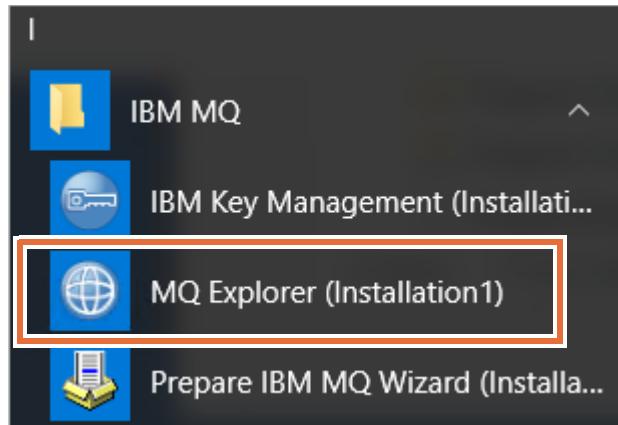
- 4. In Windows Explorer, go to the `InstPath` and `DataPath` directories to view the contents.
You refer to these directories later during the exercises.



Troubleshooting

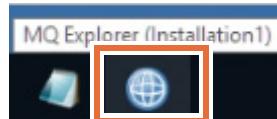
If you cannot see the `C:\ProgramData` folder, make sure that it is not hidden in Windows Explorer.

- 5. Start IBM MQ Explorer from the Windows **Start** menu by expanding the IBM MQ menu and clicking **MQ Explorer (Installation 1)**.

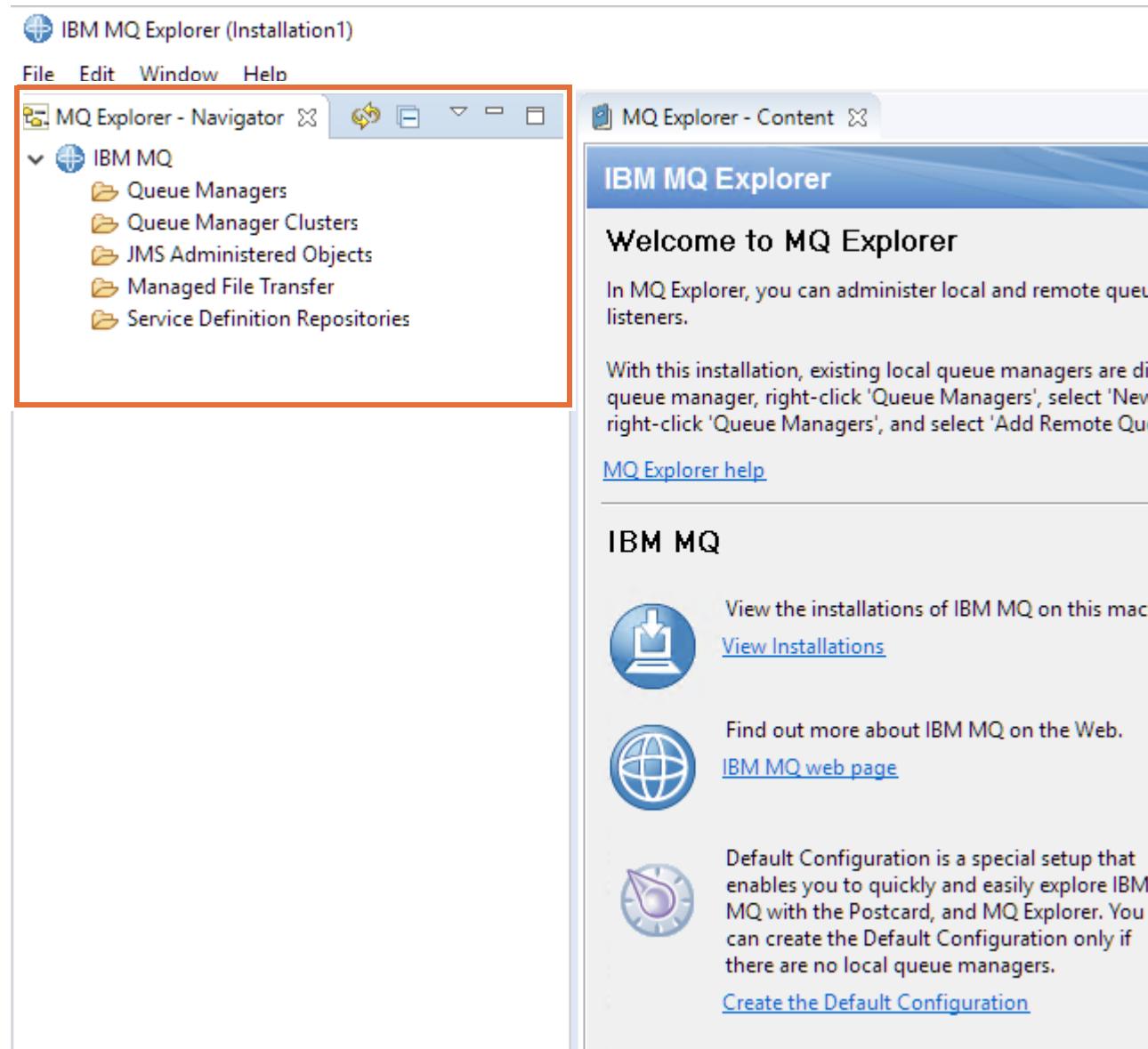


Note

You can also use the shortcut to MQ Explorer on the taskbar.



6. In MQ Explorer, in the **Navigator** pane, expand IBM MQ.



Section 2. Testing the installation

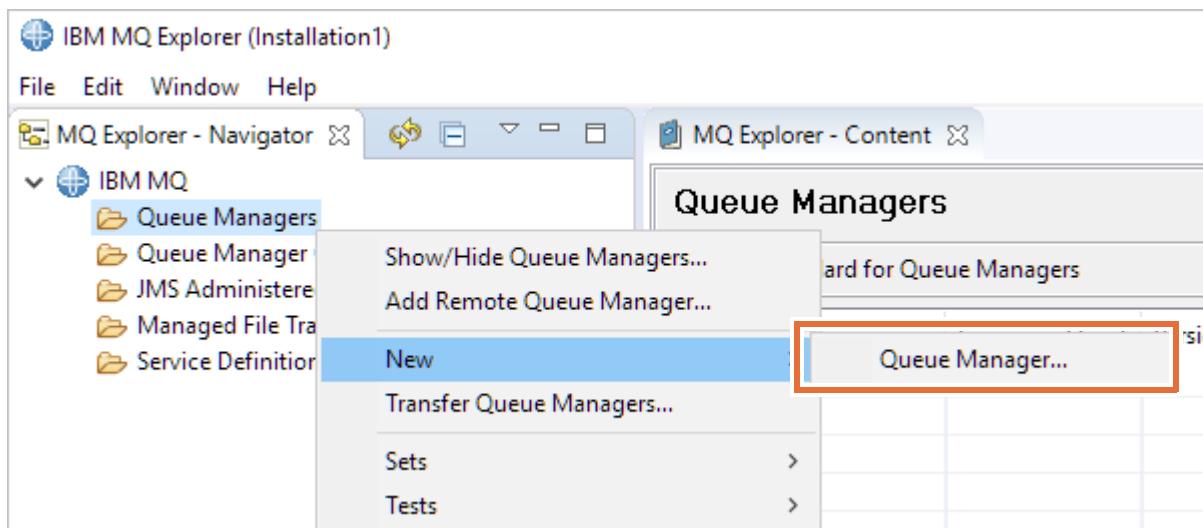
You validate your installation with these steps:

1. Create a queue manager and a queue
2. Send a message to the queue and retrieve it

For this exercise, you use MQ Explorer to test the installation.

2.1. Create a queue manager

- 1. In the **Navigator** pane, right-click **Queue Managers** and click **New > Queue Manager**.



- 2. Use the Create Queue Manager wizard to create a queue manager named **QM01**.

- a. In the **Queue manager name** field, enter: **QM01**
- b. In the **Dead-letter queue** field, enter: **SYSTEM.DEAD.LETTER.QUEUE**
- c. Click **Finish**.

Your newly created queue manager is now listed in the **Navigator** pane and in the **Content** view for Queue Managers.

Queue manager name	Command level	Ver
QM01	910	09

The status icon next to queue name shows an up-pointing arrow. In the Content view, you see the queue manager status is **running**.

3. In the **Navigator** pane, click **QM01** to see its contents in the content view for the queue manager.

You see details about the queue manager and its properties.

The screenshot shows the IBM MQ Explorer interface. The left pane, titled 'MQ Explorer - Navigator', displays a tree structure under 'IBM MQ' with 'Queue Managers' expanded, showing 'QM01'. A red box highlights 'QM01'. The right pane, titled 'MQ Explorer - Content', shows the 'Queue Manager QM01' details. Three sections are visible: 'Connection QuickView', 'Status QuickView', and 'Properties QuickView', each enclosed in a red box. The 'Connection QuickView' table includes:

Connection status	Connected
Connection type	Local
Connection name	
Channel name	
Channel definition table	
Refresh interval	15

The 'Status QuickView' table includes:

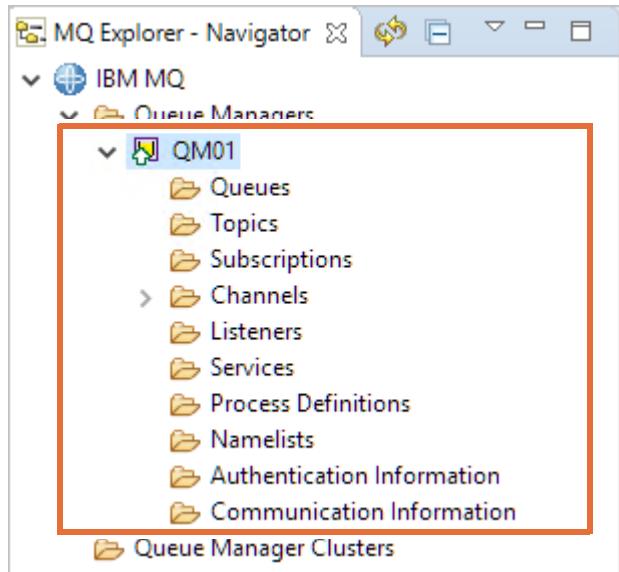
Queue manager status	Running
Command server status	Running
Channel initiator status	Running
Connection count	23
Standby	Not permitted
Start date	Dec 9, 2019
Start time	3:23:05 PM
Installation name	Installation1

The 'Properties QuickView' table includes:

Queue manager name	QM01
Description	
Platform	Windows
Command level	910
Version	09010001
Default transmission queue	

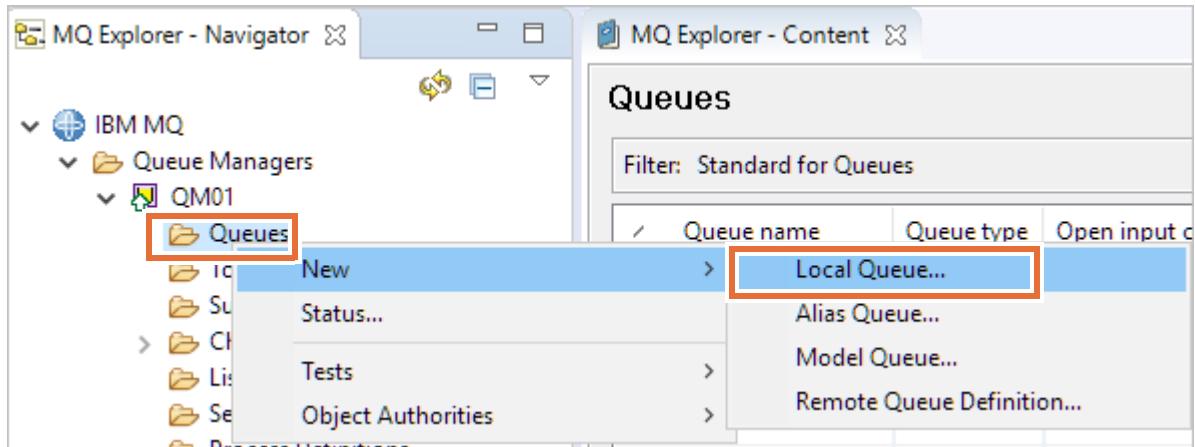
2.2. Create a queue

- 1. In the **Navigator** pane, expand the **QM01** folder to see the navigation tree.



- 2. Create a local queue that is named **myQlocal**.

- a. In the Navigator pane, in the list of folders under **QM01**, right-click **Queues** and click **New > Local Queue**.

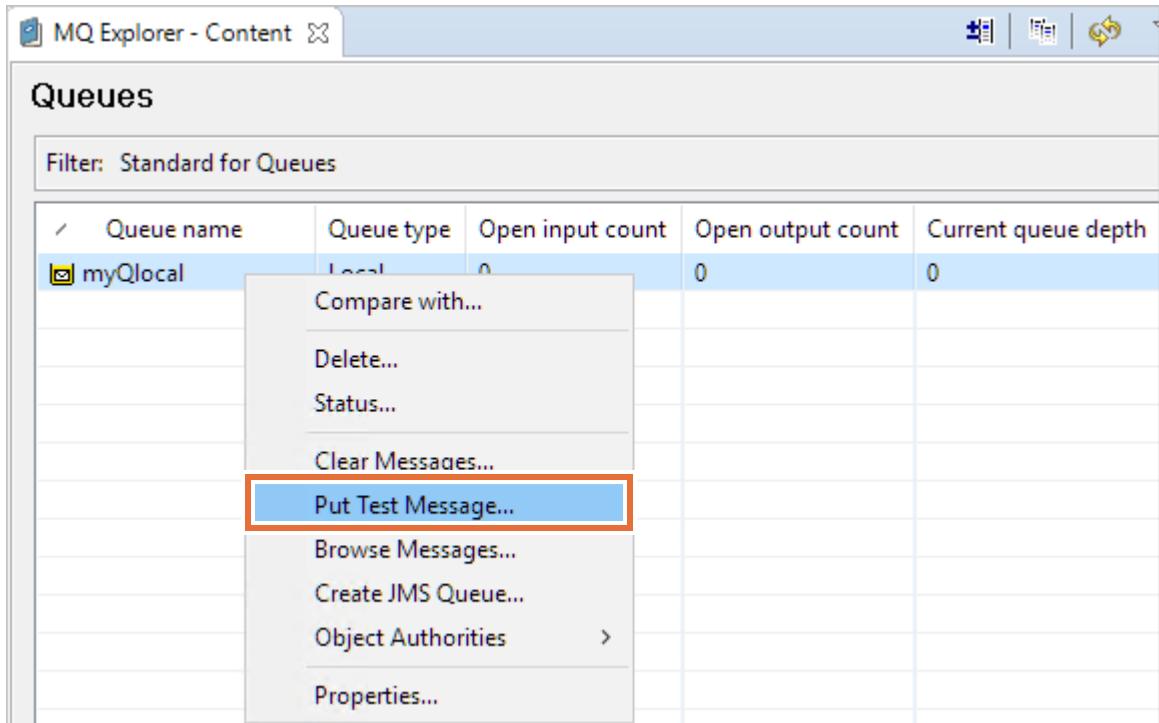


- b. Set **Name** to **myQlocal** and click **Finish**.
— c. Click **OK** to close the success message.

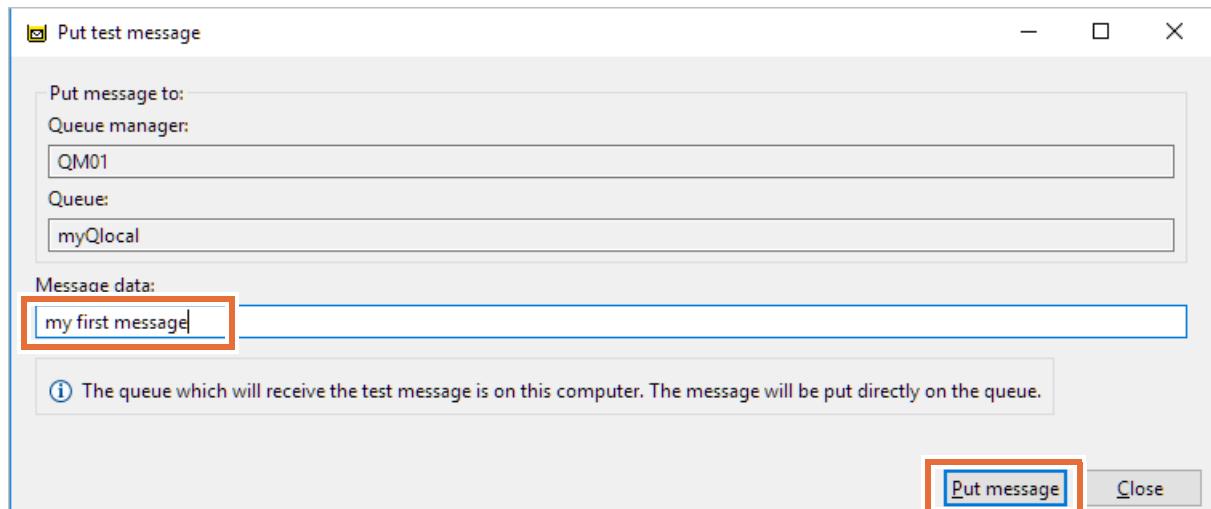
The local queue is now listed in the Queues view.

2.3. Send and retrieve a test message

1. In the Queues view, right-click myQlocal and click Put Test Message.



2. Enter a test message, such as my first message and click Put message.



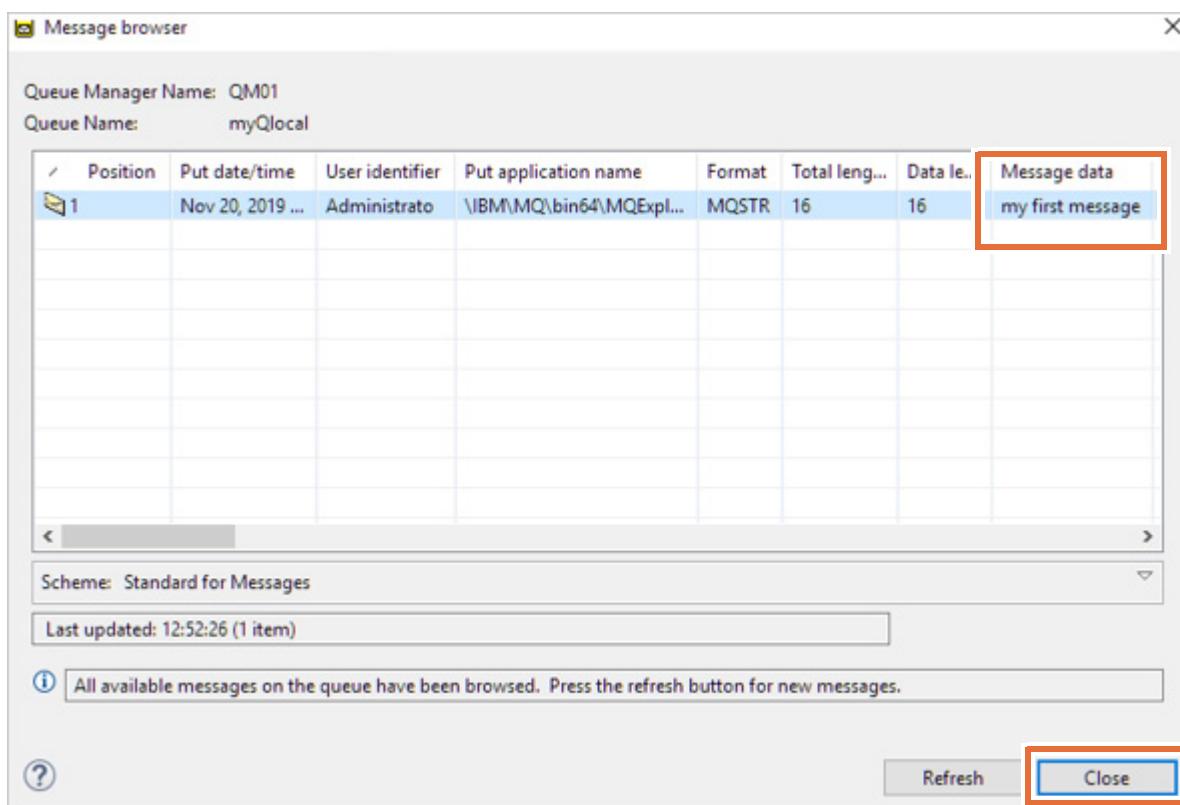
The Put test message dialog box remains open for you to continue putting messages.

3. Click Close to close the Put test message dialog box.

- ___ 4. In the Queues view, notice that the value of the **Current queue depth** property for your queue is **1**.

Queues				
Filter: Standard for Queues				
Queue name	Queue type	Open input count	Open output count	Current queue depth
myQlocal	Local	0	0	1

- ___ 5. To retrieve your message, right-click **myQlocal** and click **Browse Messages**.
The Message browser lists each message on a separate line, along with its properties.
- ___ 6. Check the **Message data** property to confirm that the value matches the message that you sent, and then click **Close**.



The screenshot shows the 'Message browser' window. At the top, it displays 'Queue Manager Name: QM01' and 'Queue Name: myQlocal'. Below this is a table listing messages in the queue:

Position	Put date/time	User identifier	Put application name	Format	Total leng...	Data le...	Message data
1	Nov 20, 2019 ...	Administrato...	\IBM\MQ\bin64\MQEExpl...	MQSTR	16	16	my first message

At the bottom of the window, there is a status bar with the message: 'All available messages on the queue have been browsed. Press the refresh button for new messages.' There are also 'Refresh' and 'Close' buttons at the bottom right.

Section 3. Setting up the mqweb server for IBM MQ Console

In this part of the exercise, you set up the IBM MQ Console for user access. You first configure basic security. Then, you start the mqweb server, connect to the IBM MQ Console, and login as an administrator.

3.1. Configure basic security for users and groups to access the IBM MQ Console

- 1. In Windows Explorer, go to the C:\Program Files\IBM\MQ\web\mq\samp\configuration directory, and explore the `basic_registry.xml` file in Notepad.
- a. Find the definitions for these security roles:
 - MQWebAdmin
 - MQWebAdminRO
 - MQWebUser

```
<!-- Roles for the MQ Console -->
<enterpriseApplication id="com.ibm.mq.console">
    <application-bnd>
        <security-role name="MQWebAdmin">
            <group name="MQWebAdminGroup" realm="defaultRealm"/>
        </security-role>
        <security-role name="MQWebAdminRO">
            <user name="mqreader" realm="defaultRealm"/>
        </security-role>
        <security-role name="MQWebUser">
            <special-subject type="ALL_AUTHENTICATED_USERS"/>
        </security-role>
        <security-role name="MFTWebAdmin">
            <user name="mftadmin" realm="defaultRealm"/>
        </security-role>
        <security-role name="MFTWebAdminRO">
            <user name="mftreader" realm="defaultRealm"/>
        </security-role>
    </application-bnd>
</enterpriseApplication>
```

- __ b. Find the **basicRegistry** section to see the defined users and groups.

```

<!-- Sample Basic Registry -->
<basicRegistry id="basic" realm="defaultRealm">
    <!--
        This sample defines two users with unencoded passwords
        and a group, these are used by the role mappings above.
    -->
    <user name="mqadmin" password="mqadmin"/>
    <user name="mqreader" password="mqreader"/>
    <user name="mftadmin" password="mftadmin"/>
    <user name="mftreader" password="mftreader"/>
    <group name="MQWebAdminGroup">
        <member name="mqadmin"/>
    </group>
</basicRegistry>
```

The `mqadmin` user is defined as a member of the `MQWebAdminGroup` group.

- __ c. Close the `basic_registry.xml` file.
- __ 2. Create a backup of the `mqwebuser.xml` file.
 - __ a. In a separate Windows Explorer window, go to the `C:\ProgramData\IBM\MQ\web\installations\Installation1\servers\mqweb` directory.
 - __ b. Rename the `mqwebuser.xml` file to: `mqwebuser.xml.backup`
- __ 3. Replace the `mqwebuser.xml` file for the `mqweb` server.
 - __ a. Copy the `basic_registry.xml` file from `C:\Program Files\IBM\MQ\web\mq\samp\configuration` to `C:\ProgramData\IBM\MQ\web\installations\Installation1\servers\mqweb`
 - __ b. In the `C:\ProgramData\IBM\MQ\web\installations\Installation1\servers\mqweb` directory, rename the `basic_registry.xml` file as: `mqwebuser.xml`

3.2. Start the `mqweb` server

- __ 1. In a command prompt, start the `mqweb` server:

`strmqweb`

The command returns these messages:

```

Starting server mqweb.
Server mqweb started.
```



Note

You need to run this command each time you restart your system.

- __ 2. Display the URL for the IBM MQ Console by entering the following command.

```
dspmqweb
```

The command should return information about the status of the web server and the URLs for the web console and REST API.

Server mqweb is running.

URLs:

<https://localhost:9443/ibmmq/console/>

<https://localhost:9443/ibmmq/rest/v1/>



Information

The `dspmqweb` command returns URLs for the console and the REST API. The REST API is not used in this exercise.

3.3. Start IBM MQ Console

- __ 1. Connect to IBM MQ console in a browser by entering the console URL you obtained in [Section 3.2, "Start the mqweb server"](#).
- __ a. In Mozilla Firefox, go to this URL:
<https://localhost:9443/ibmmq/console>
- __ b. When prompted about security, click **Advanced** to add the exception and continue.
- __ 2. Sign in to the console with these credentials and click **Login**:
- **User Name:** mqadmin
 - **Password:** mqadmin

The IBM MQ Console displays the default dashboard.

The default dashboard includes a window frame or *widget* that shows any local queue managers on the server. You have one queue manager running in the **Local Queue Managers** widget.

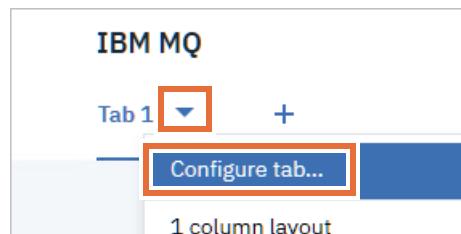
The screenshot shows the IBM MQ dashboard interface. At the top, there's a header bar with the IBM MQ logo, a close button, and a plus sign for adding tabs. Below the header is a browser-style toolbar with icons for back, forward, search, and refresh, followed by the URL https://localhost:9443/ibmmq/console/. To the right of the URL are various browser controls like a menu, a refresh icon, and a star icon for bookmarks.

The main content area is titled "IBM MQ". It features a tab bar with "Tab 1" selected and a plus sign to add more tabs. Below the tabs is a "Local Queue Managers" section with a "Search" input field and a "Configure tab..." button. A blue-bordered box highlights the "Add widget" button in the top right corner of the dashboard area.

The "Local Queue Managers" section contains a table with two columns: "Name" and "Status". There is one entry: QM01, which is listed as "Running" with a green dot icon.

3.4. Customize the tab name

- 1. To customize the name of the tab on your dashboard, click the down arrow next to **Tab 1** and click **Configure tab...**.



- 2. Set **Tab name** to: Overview
- 3. In the **Description** field, type: Lab environment
- 4. Click **Rename**.

3.5. Create a queue manager



Note

Depending on your browser and display settings, you might need to zoom out or expand the window to see the icons that are used in the next steps.

- ___ 1. Create a queue manager that is named QM02 on port 1418.
 - ___ a. Click the **Create (+)** icon on the **Local Queue Managers** widget.

Local Queue Managers	
Search Create +	
▲ Name	Status
QM01	● Running

- ___ b. Set **Queue manager name** to: QM02
- ___ c. Set **Port** to: 1418
- ___ d. Click **Create**.

The new queue manager is created and started.

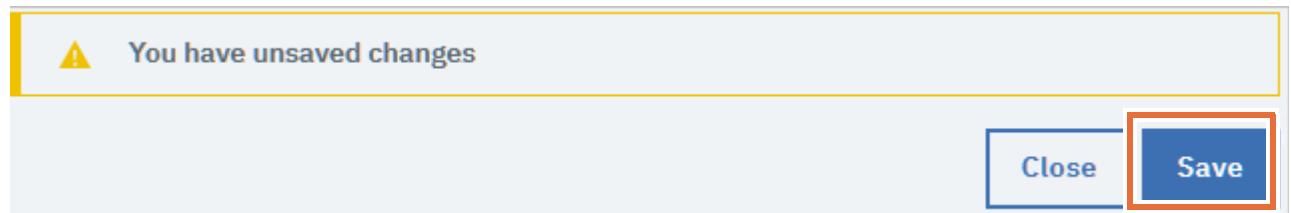
Local Queue Managers	
Search Create +	
▲ Name	Status
QM01	● Running
QM02	● Running

- 2. In the **Local Queue Managers** widget, click **QM02** and click the **Properties** icon to open the queue manager properties.

Name	Status
QM01	Running
QM02	Running

The properties editor contains information that is familiar to administrators who use either MQSC or IBM MQ Explorer to view queue manager properties.

- 3. On the **General** tab, type a description into the **Description** field, such as: Queue manager in Console lab
- 4. Click **Save** to accept the changes, and then click **Close**.



3.6. Add a widget

- 1. Add a widget to manage queues.
- a. In the upper right, click **Add widget**.

- __ b. On the Add a new widget page, select **QM01** in the queue managers list and click **Queues**.

Add a widget to display MQ object information for the specified queue manager

Queue manager:

QM01

Queues Configure destinations for messages

Topics Administrative objects for assigning

Close

- __ 2. Click **myQlocal** and notice the Queue depth column shows 1 message on the queue.

Queues on QM01			
	Name	Queue type	Queue depth
	myQlocal	Local	1

- __ 3. Put a message on the queue.

- __ a. Click the **Put message** icon to put a test message.

Queues on QM01			
	Name	Queue type	Queue depth
	myQlocal	Local	1

- ___ b. Type a message in the **Message** field, such as my second message, and click **Put**.
- ___ 4. View the messages.
- ___ a. Click the **Browse messages** icon to view the contents of the messages.

Queues on QM01

Name	Queue type	Queue depth
myQlocal	Local	1

- ___ b. Click **Close**.
- ___ 5. Delete the messages.
- ___ a. Click the ellipsis (...) menu and click **Clear queue**.

Queues on QM01

Name	Queue type
myQlocal	Local

Ellipsis menu options:

- Clear queue...
- Manage authority records...

- ___ b. When prompted to confirm, click **Clear queue**.
- ___ 6. Delete the queue.
- ___ a. With the queue selected, click the Delete icon.

Queues on QM01

Name	Queue type	Queue depth
myQlocal	Local	1

- ___ b. When prompted to confirm, click **Delete**.

3.7. Delete a queue manager



Attention

Deleting a queue manager is a drastic step because you also delete all resources associated with the queue manager, including all queues and their messages and all object definitions.

- ___ 1. In the **Local Queue Managers** widget, click **QM02** and click the **Stop** icon to stop the queue manager.

Name	Status
QM01	Running
QM02	Running

- ___ 2. When prompted to confirm, click **Stop**.
- ___ 3. After QM02 is stopped, click the **Delete** icon.

Name	Status
QM01	Running
QM02	Stopped

- ___ 4. When prompted to confirm, click **Delete**.

3.8. Exercise cleanup

- ___ 1. Close or minimize the browser running IBM MQ Console. You return to it in later exercises.
- ___ 2. Close Windows Explorer.

___ 3. Close any open command prompts.

You keep MQ Explorer running for the next exercise.

End of exercise

Exercise review and wrap-up

This exercise showed you how to validate your IBM MQ installation by creating a queue manager and queue, and sending test messages. You also set up the mqweb server for IBM MQ Console.

Exercise 2. Working with IBM MQ administration tools

Estimated time

01:00

Overview

In this exercise, you use IBM MQ control commands to create a queue manager, start it, and create queues. You also use MQ script commands and command files. You also use IBM MQ Explorer to create queue manager sets to administer a group of queue managers collectively.

Objectives

After completing this exercise, you should be able to:

- Work with IBM MQ administration tools to create and manage queue managers, queues, and messaging
- Create a queue manager set to administer queue managers as a group

Introduction

This exercise includes these sections:

- [Section 1, "Create and start a queue manager with IBM MQ control commands"](#)
- [Section 2, "Using MQSC commands"](#)
- [Section 3, "Modify local queues"](#)
- [Section 4, "Working with IBM MQ sample programs"](#)
- [Section 5, "Create queue manager sets"](#)
- [Section 6, "Deleting queue manager sets"](#)
- [Section 7, "Exercise cleanup"](#)

Requirements

Make sure that you complete [Exercise 1, "Getting started with IBM MQ"](#) before starting this exercise.

Section 1. Create and start a queue manager with IBM MQ control commands

- ___ 1. Open a command prompt and type the following command to create a queue manager named QM02.

```
crtmqm -p 2415 -u SYSTEM.DEAD.LETTER.QUEUE QM02
```

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



Note

You use this queue manager throughout the exercises in this course. The queue manager and dead-letter queue names are case-sensitive.

- ___ 2. Start the queue manager by typing this command:

```
strmqm QM02
```

You should see the message that the queue manager started.

- ___ 3. Verify that the QM02 queue manager is created and running:

- ___ a. Open IBM MQ Explorer to see that QM02 is listed in the **Navigator** pane with a green arrow, which indicates that it is running.
- ___ b. Open IBM MQ Console in a browser (<https://localhost:9443/ibmmq/console>), sign in with `mqadmin` as the username and password, and note that the status for QM02 is **Running**.

- ___ 4. Minimize IBM MQ Explorer and IBM MQ Console. You return to these tools later.

Section 2. Using MQSC commands

2.1. Using MQSC commands interactively

In this part, you work with MQSC commands interactively on QM02.

- ___ 1. In a command prompt, start an MQSC session for QM02 by typing the following command:

```
runmqsc QM02
```

You see a message that MQSC is starting for the queue manager QM02. You are in MQSC mode until you enter the `END` command.



Note

When MQSC mode starts, you do not see any prompts but you can enter text.

```
C:\Users\Administrator>runmqsc QM01
5724-H72 (C) Copyright IBM Corp. 1994, 2018.
Starting MQSC for queue manager QM01.
```

- ___ 2. Display all the attributes of the queue manager and verify the value of the `DEADQ` attribute.

- ___ a. Type this command to display queue manager attributes:

```
DISPLAY QMGR
```

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

- ___ b. In the list of attributes, verify that the `DEADQ` attribute is set to:

```
SYSTEM.DEAD.LETTER.QUEUE
```

```
display qmgr
  1 : display qmgr
AMQ8408I: Display Queue Manager details.
QMNAME(QM01)                               ACCTCONO(DISABLED)
ACCTINT(1800)                                ACCTMQI(OFF)
ACCTQ(OFF)                                 ACTIVREC(MSG)
ACTVCONO(DISABLED)                           ACTVTRC(OFF)
ADVCAP(DISABLED)                            ALTDATE(2019-08-05)
ALTTIME(23.40.16)                           AMQPCAP(NO)
AUTHOREV(DISABLED)                           CCSID(437)
CERTLBL(ibmwebspheremqqm01)                 CERTVPOL(ANY)
CHAD(DISABLED)                             CHADEV(DISABLED)
CHADEXIT( )                                CHLEV(DISABLED)
CHLAUTH(ENABLED)                           CLWLDATA( )
CLWLEXIT( )                                CLWLLEN(100)
CLWLMRUC(99999999)                           CLWLUSEQ(LOCAL)
CMDEV(DISABLED)                            CMDLEVEL(910)
COMMANDQ(SYSTEM.ADMIN.COMMAND.QUEUE)        CONFIGEV(DISABLED)
CONNAUTH(SYSTEM.DEFAULT.AUTHINFO.IDPWOS)      CRRTIME/22 AG 161
CRDATE(2019-08-05)                           DEADQ(SYSTEM.DEAD.LETTER.QUEUE)
CUSTOM( )                                     DEFAMIL( )
DEFCLXQ(SCTQ)
```

- ___ 3. Display all the queues with queue names that begin with the characters SYSTEM by typing this command:

```
DISPLAY Q(SYSTEM*)
```

- ___ 4. Create a local queue that is named QL.A with a text description of "QL.A TEXT" by typing this command.

```
DEFINE QL(QL.A) REPLACE DESC('QL.A TEXT')
```

- ___ 5. Modify the QL.A attributes.

- ___ a. Display all the attributes of the local queue, named QL.A, and verify that the text that is entered in the previous step is the value for the DESC attribute.

```
DISPLAY QL(QL.A)
```

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

```
ALTER QL(QL.A) MAXDEPTH(1000)
```

- ___ c. Display the queue attributes for QL.A and verify the change to the MAXDEPTH attribute.

```
DISPLAY QL(QL.A)
```



Information

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

- ___ d. 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.

- ___ 6. Define another local queue that is named QL.B and modify its attributes.

- ___ a. Create the local queue with a text description.

```
DEFINE QL(QL.B) REPLACE DESC('QL.B TEXT')
```

- ___ b. Display all the attributes of the queue and verify that the text appears in the DESC attribute.

```
DISPLAY QL(QL.B)
```

- ___ c. 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.

```
DEFINE QL(QL.B) REPLACE MAXDEPTH(2000)
```

- ___ d. Display the queue attributes of QL.B.

```
DISPLAY QL(QL.B)
```

You should see that the MAXDEPTH attribute was modified and that the DESC attribute is now blank.

- ___ 7. Exit the `runmqsc` mode by typing `end` and pressing Enter.

`end`

You see the following message:

```
9 MQSC commands read
No commands have a syntax error.
All valid MQSC commands were processed.
```

- ___ 8. Open IBM MQ Explorer to confirm that the queues were created on QM02.

- ___ a. In the **Navigator** pane, expand **Queue Managers > QM02**.
- ___ b. Click **Queues**.

You see the QL.A and QL.B queues.

2.2. Running MQSC commands from a text file

In this part, you work with MQSC commands in a script file on **QM01**.

- ___ 1. Create a command file to run the MQSC commands.

- ___ a. Open Notepad and enter the following 9 commands:

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

These are the same commands that you entered one at a time in [Section 2.1, "Using MQSC commands interactively"](#).

- ___ b. Save the file to the `C:\labfiles\Lab02-queues` directory, with the name: `CreateQs.mqsc`



Troubleshooting

Make sure that the file is *not* mistakenly saved as a **TXT** file: `CreateQs.mqsc.txt`

- ___ 2. Run the command file and send the results to a file: `report.txt`

- ___ a. In the command window, switch to the directory where you saved the file.

```
cd C:\labfiles\Lab02-queues
```

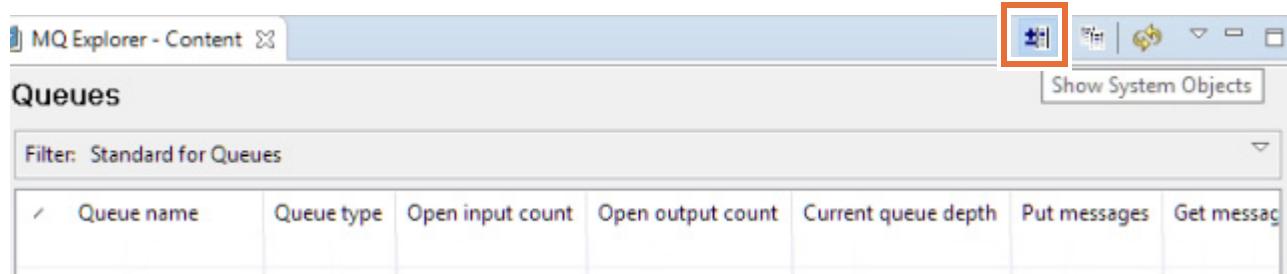
- ___ b. Run the command file by typing:

```
runmqsc QM01 < CreateQs.mqsc > report.txt
```

- ___ 3. Go to the C:\labfiles\Lab02-queues directory, and open the report.txt file to verify that the final message in the file states:
 - 9 MQSC commands read
 - No commands have a syntax error.
 - All valid MQSC commands were processed.
- ___ 4. Open IBM MQ Console to confirm that the queues were created on QM01.
 - ___ a. Open the browser running IBM MQ Console (<https://localhost:9443/ibmmq/console>).
 - ___ b. Sign in with mqadmin/mqadmin.
 - ___ c. You might need to refresh the **Queues on QM01** widget (or refresh the browser).
You see the QL.A and QL.B queues.

Section 3. Modify local queues

- ___ 1. In IBM MQ Explorer, open the queues for QM02.
 - ___ a. In the **Navigator** pane, click **Queue Managers > QM02 > Queues**.
 - ___ b. In the Queues view for QM02, click the **Show System Objects** icon.

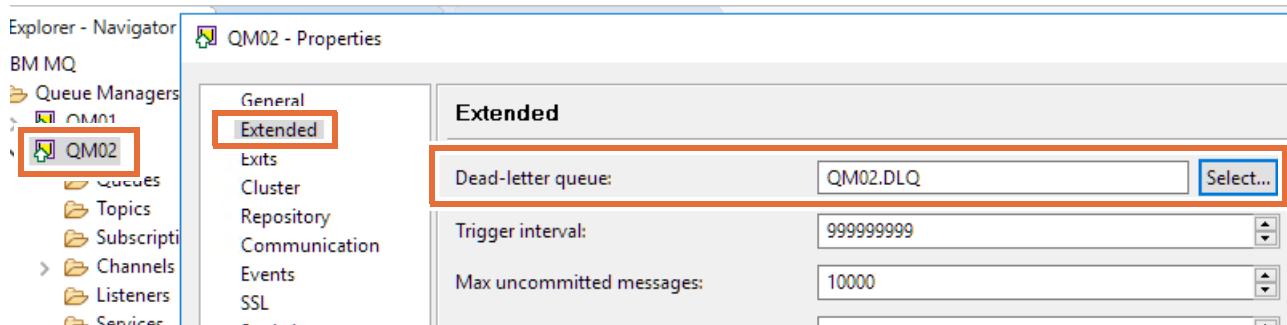


The **Show System Queues** icon is a toggle.

- ___ c. Hide the SYSTEM queues by clicking the **Show System Objects** icon again.
- ___ 2. Create a local queue, named **QM02.DLQ**, to use as the dead-letter queue.
 - ___ a. In the **Navigator** pane, right-click the **QM02 > Queues** folder and click **New > Local Queue**.
 - ___ b. Set **Name** to **QM02.DLQ** and click **Finish**.
 - ___ c. Click **OK** to close the success message.

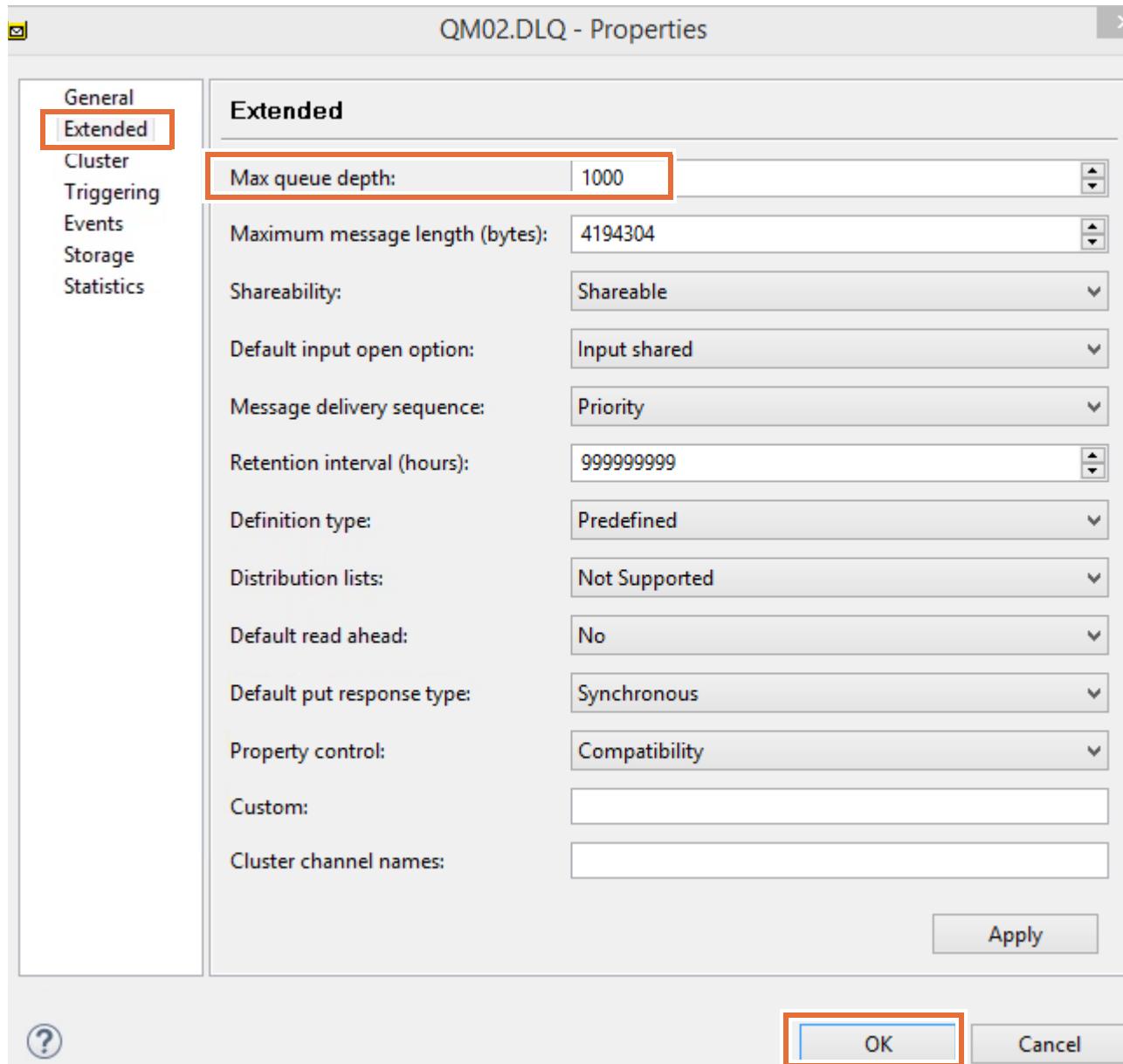
The new queue is listed in the **Queues** view. You can now select this queue to become the dead-letter queue for the queue manager.

 - ___ d. In the Navigator pane, right-click **QM02** and click **Properties**.
 - ___ e. In the Properties dialog box, click **Extended** in the left menu.
 - ___ f. Beside the **Dead-letter queue** field, click **Select**, select **QM02.DLQ** in the list of queues, and click **OK**.



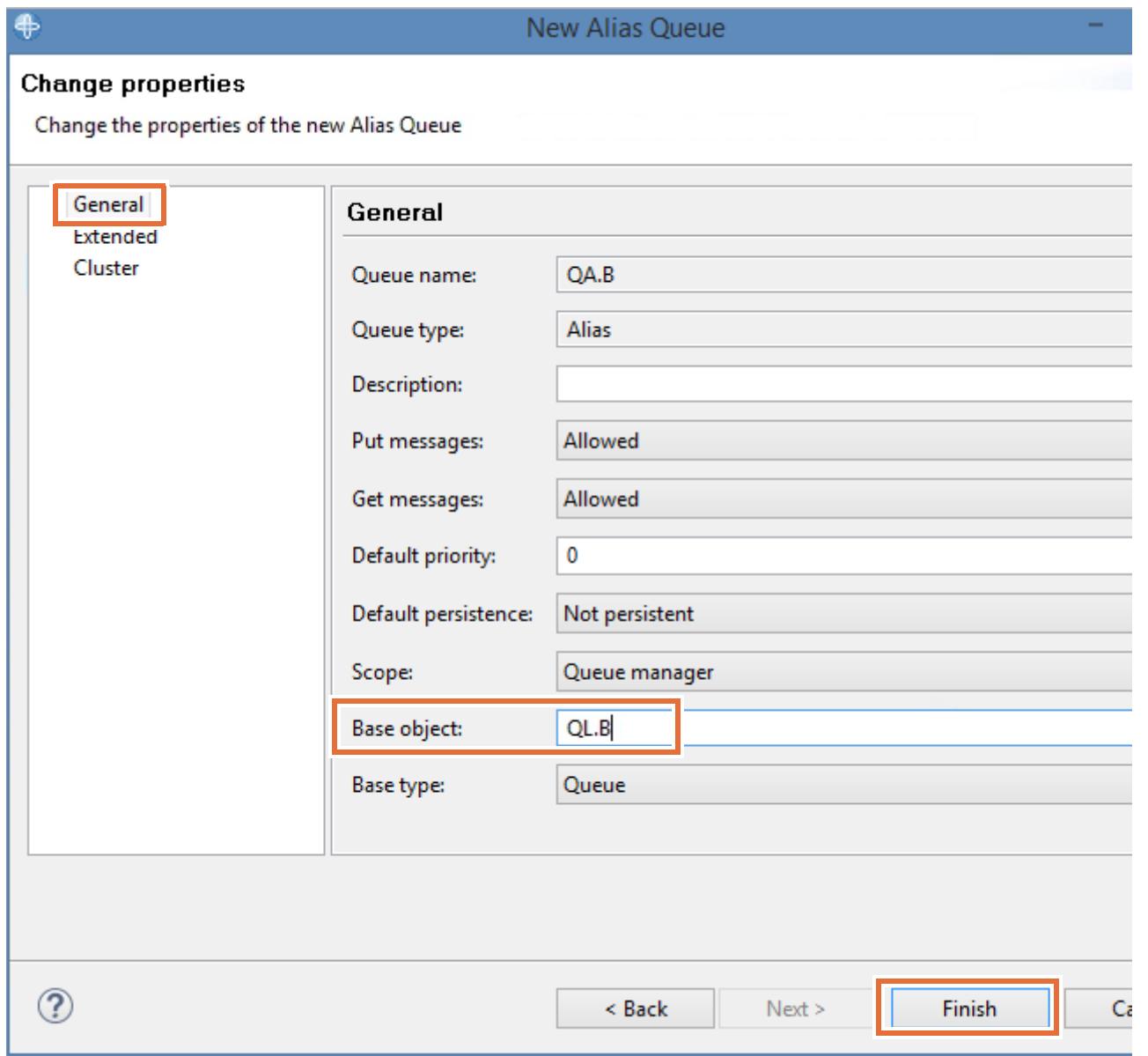
- ___ g. Click **OK** to save this property change.
- ___ 3. Change the maximum number of messages that are allowed on the queue QM02.DLQ to 1000.
 - ___ a. In the Navigator pane, click the **QM02 > Queues** folder to reopen the Queues view.
 - ___ b. In the Queues view, right-click **QM02.DLQ** and click **Properties**.

- ___ c. Click **Extended** in the left menu.
- ___ d. Change the **Max queue depth** property value to: 1000
- ___ e. Click **OK**.



- ___ 4. Change the default persistence on the local queue QL.B to **Persistent**.
 - ___ a. In the **Queues** view, right-click **QL.B** and click **Properties**.
 - ___ b. On the **General** page, change the **Default persistence** property to **Persistent**.
 - ___ c. Click **OK**.
 - ___ d. In the **Queues** view, scroll to the right to verify that the **Default persistence** column now shows **Persistent** for QL.B.

- ___ 5. Create an alias queue on QM02 that is named **QA.B** that resolves to the local queue **QL.B**.
 - ___ a. In the **Navigator** pane, right-click the **QM02 > Queues** folder and click **New > Alias Queue**.
 - ___ b. Set **Name** to **QA.B** and click **Next**.
 - ___ c. On the **General** page, set **Base object** to **QL.B** and click **Finish**.



- ___ d. Click **OK** to close the success message.
- ___ 6. Inhibit PUT requests on the QA.B alias queue on QM02.
 - ___ a. In the **Queues** view, right-click **QA.B** and click **Properties**.
 - ___ b. On the **General** page, change the **Put messages** property to **Inhibited**.
 - ___ c. Click **OK**.
 - ___ d. In the **Queues** view, check that the **Put messages** column for QA.B shows **Inhibited**.

Section 4. Working with IBM MQ sample programs

In this section, you work with the queues on QM02.

4.1. Putting messages: amqsput

In this part of the exercise, you use the IBM MQ sample programs to put, get, and browse messages.

- 1. Open a command prompt and type the following command to use the `amqsput` sample program to put two messages on the local queue QL.A on QM01.

```
amqsput QL.A QM02
```

The program responds with:

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

- 2. Enter some lines of text as data and press Enter after each line. For example:

```
my test data
another test
a third test
```

Each line of data that is entered becomes the data portion of a new message.

- 3. End the `amqsput` sample program by pressing Enter again.



Troubleshooting

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

To interpret the reason code, use the control command `mqrc` at the C: prompt followed by the 4-digit reason code. For example, type: `mqrc 2085`

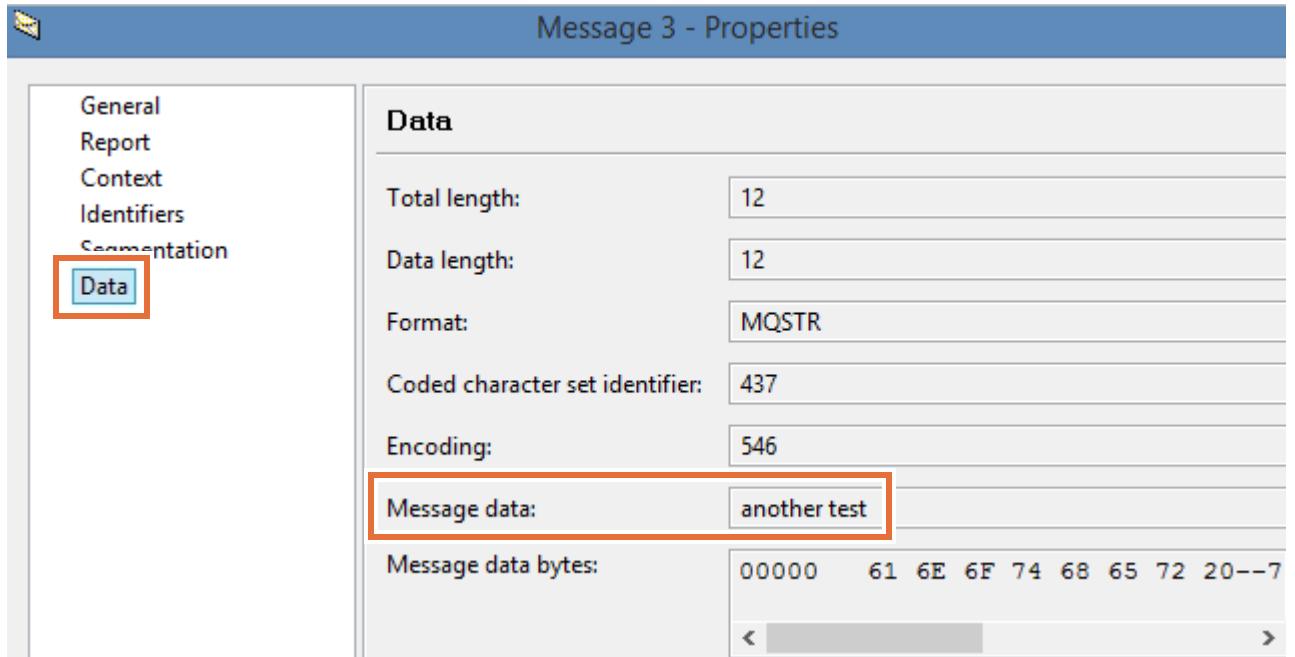
- 4. View the messages in IBM MQ Explorer.

- a. In the **Queues** view for QM02, right-click **QL.A** and click **Browse messages**.

The **Message browser** list shows one message per line for every message that you entered with the `amqsput` sample program.

Message browser				
Queue Manager Name: QM01 Queue Name: QL.A				
Position	/ Put date/time	User identifier	Put application name	Format
1	May 7, 2019 12:36:...	Administrato	les\IBM\MQ\bin64\amqsput.exe	MQSTR
2	May 7, 2019 12:36:...	Administrato	les\IBM\MQ\bin64\amqsput.exe	MQSTR
3	May 7, 2019 12:36:...	Administrato	les\IBM\MQ\bin64\amqsput.exe	MQSTR

- ___ b. Double-click a message from the list of messages to display its properties.
- ___ c. Click **Data** tab to view the message data.



- ___ d. Click **Close** to close the **Properties** window.
- ___ e. Click **Close** to close the message browser window.

4.2. Browsing messages: amqsbcg

- ___ 1. Return to the command prompt and make sure that you are in the `C:\labfiles\Lab02-queues` directory.
`cd C:\labfiles\Lab02-queues`
- ___ 2. Type the following command to use the `amqsbcg` program to send the queue messages to a text file:
`amqsbcg QL.A QM02 > out.txt`
- ___ 3. In Windows Explorer, open the `out.txt` file to view the contents of the messages on QL.A.

4.3. Getting messages: amqsget

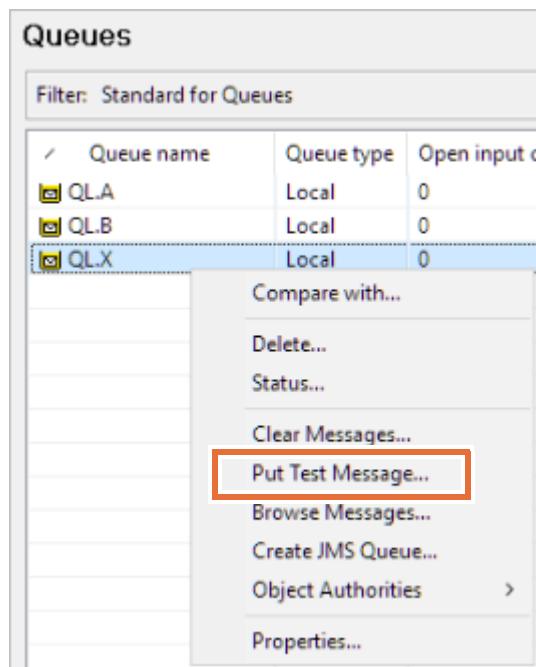
- ___ 1. In the command prompt, type the following command:
`amqsget QL.A QM02`
The `amqsget` program retrieves all the messages on the queue and empties the queue.
- ___ 2. Wait for the following message to know that the program finished running.
`no more messages`
`Sample AMQGET0 end`

- ___ 3. Use the `amqsput` sample program to put three more messages on the queue QL.A:
 - ___ a. In the command prompt, start the `amqsput` sample program:
`amqsput QL.A QM02`
 - ___ b. Enter 3 lines of text, pressing Enter after each line.
 - ___ c. Press Enter again to stop the program.

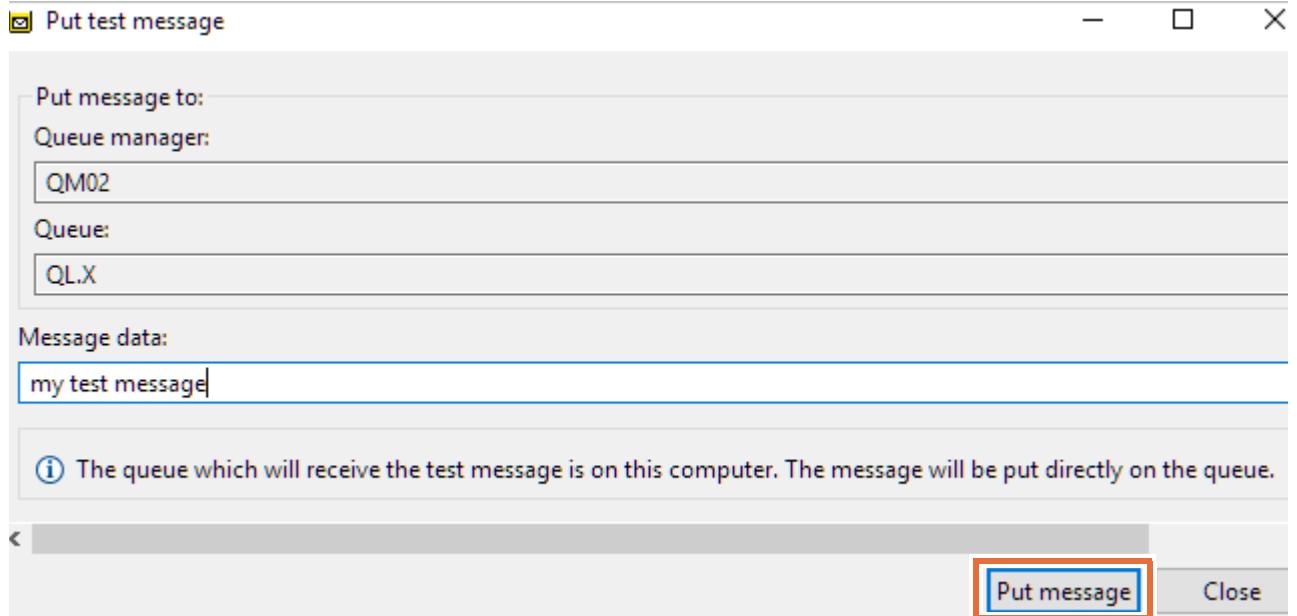
4.4. Manage messages in MQSC mode

- ___ 1. Use an MQSC command to show the number of messages on the queue (`CURDEPTH`) of QL.A and verify that the messages are on the queue.
 - ___ a. In the command prompt, start MQSC mode by typing:
`runmqsc QM02`
 - ___ b. Enter the following command to display the number of messages on the queue:
`DIS Q(QL.A) CURDEPTH`

The response shows that the `CURDEPTH` value is 3.
- ___ 2. Define a new local queue on QM01 that is named `QL.X` with the attributes of the local queue QL.A by typing the following command in the MQSC window:
`DEF QL(QL.X) LIKE (QL.A)`
- ___ 3. Go to IBM MQ Explorer and put some messages on QL.X.
 - ___ a. In the Queues view, right-click **QL.X** and click **Put message**.



- ___ b. Type some text, such as my test message and click **Put Test Message**.



- ___ 4. Click **Close** to close the “Put test message” dialog box, and note that the current queue depth value for QL.X is 1.

4.5. Clear messages and delete queues

- ___ 1. Return to the command prompt that is running MQSC, and type the following command to clear the messages from the local queue QL.A.

```
CLEAR QL(QL.A)
```

- ___ 2. Try to clear the messages from the alias queue QA.A by typing:

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

- ___ 3. Try to delete the local queue QL.X by typing:

```
DELETE QL(QL.X)
```

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

- ___ 4. Delete the QL.X queue and the messages, by typing:

```
DELETE QL(QL.X) PURGE
```

- ___ 5. Stop the MQSC program by typing: end

- ___ 6. Close the command prompt.

Section 5. Create queue manager sets

In this part of the exercise, you learn how to group queue managers into “sets” as a convenient way to view and manage groups of queue managers.



Information

A queue manager can be a member of one set or many sets. A set cannot contain another set.

For this exercise, you create a queue manager set that is named **Production** that includes these queue managers:

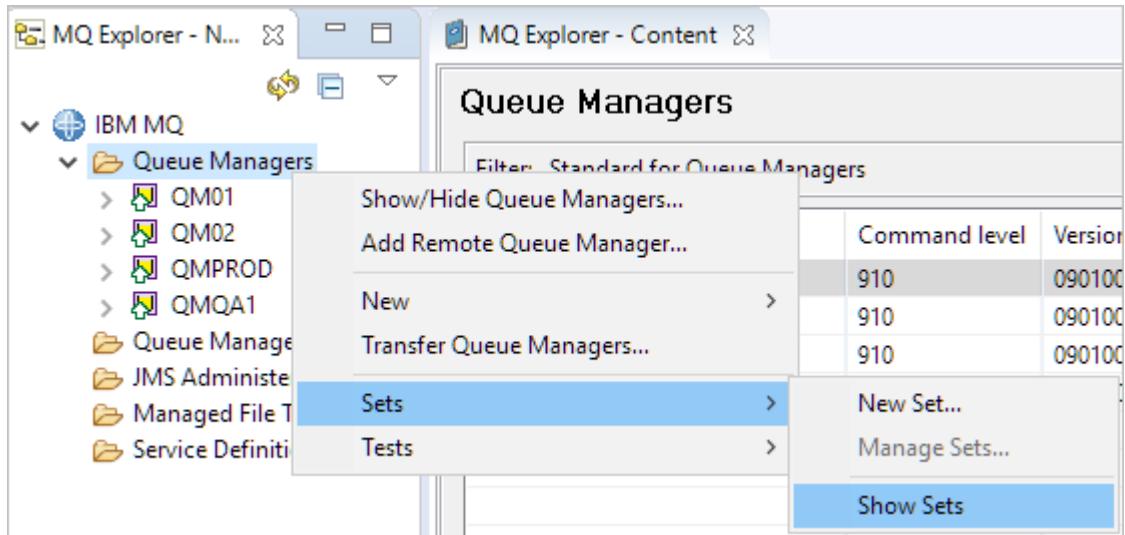
Queue manager	Port
QMPROD	2415
QMQA1	2416

5.1. Create the production queue managers

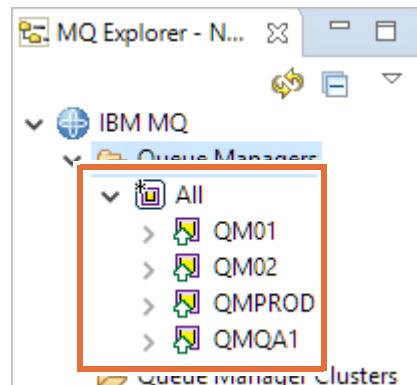
- ___ 1. In IBM MQ Console, create the **QMPROD** queue manager.
 - ___ a. Open the browser that is running IBM MQ Console (<https://localhost:9443/ibmmq/console>), and sign in with `mqadmin` as the username and password.
 - ___ b. In the Local Queue Managers widget, click the **Create (+)** icon.
 - ___ c. Set **Queue manager name** to: **QMPROD**
 - ___ d. Set **Port** to: **2415**
 - ___ e. Click **Create**.
- ___ 2. Create the **QMQA1** queue manager.
 - ___ f. In the Local Queue Managers widget, click the **Create (+)** icon.
 - ___ g. Set **Queue manager name** to: **QMQA1**
 - ___ h. Set **Port** to: **2416**
 - ___ i. Click **Create**.

Note that all the queue managers are running.

- 3. Go to the **Navigator** pane in MQ Explorer, right-click **Queue Managers** and click **Sets > Show Sets**.



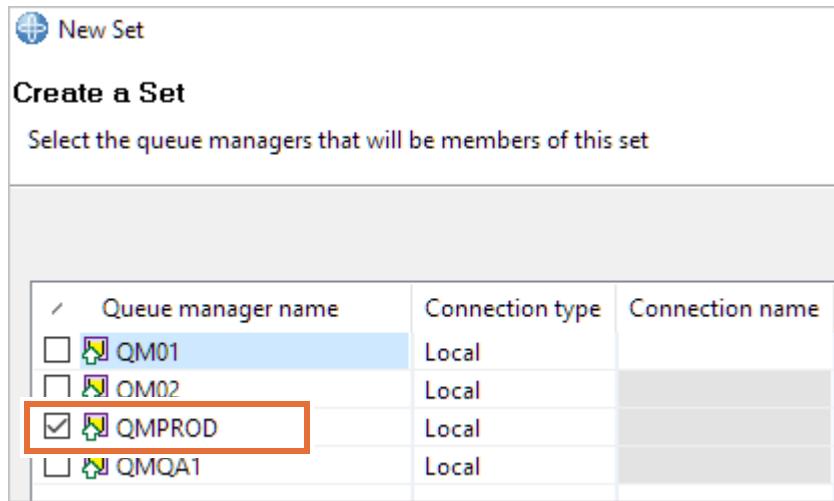
The Queue Manager folder now lists the queue managers as a set called **All**.



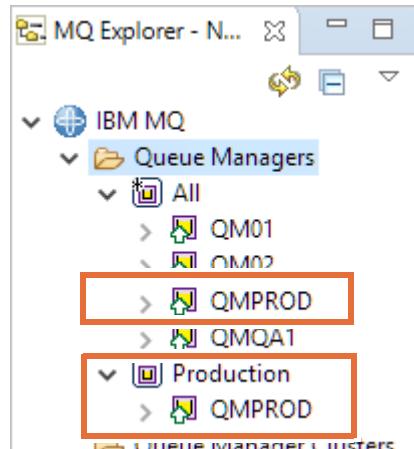
5.2. Define a new queue manager set

- 1. In the **Navigator** pane, right-click **Queue Managers** and click **Sets > New Set**.
- 2. Set **Name** to **Production**, set the type to **Manual**, and click **Next**.

- 3. On the Create a Set page, select **QMPROD** and click **Finish**.



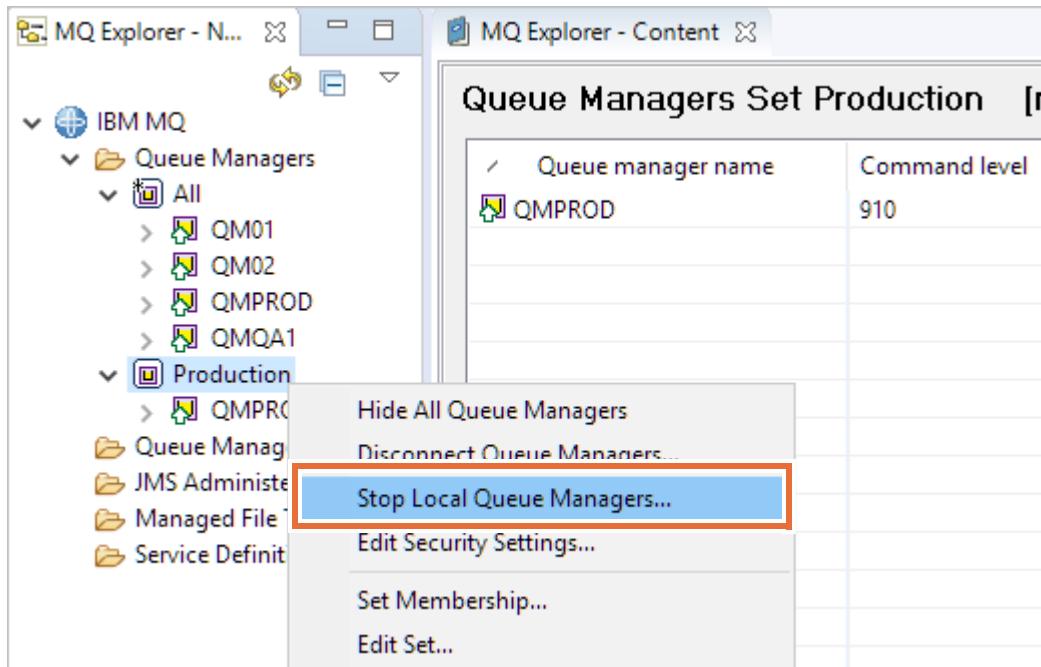
- 4. Expand the **Production** set folder and note that QMPROD is a member of both the All and Production sets.



Information

After you create a set, you can manage all the queue managers in the set with a single request.

- ___ 5. Right-click **Production**, and click **Stop Local Queue Managers**.



- ___ 6. Leave the stop method as **Controlled** and click **OK**.
 ___ 7. When the Status changes to **Stopped**, click **Close**.

5.3. Create a set by using a filter

- ___ 1. In the **Navigator** pane, right-click **Queue Managers** and click **Sets > New Set**.
 ___ 2. Set **Name** to **QA.QMGR**, set the type to **Automatic**, and click **Next**.

- ___ 3. On the Filters page, click **Manage Filters**.

The screenshot shows the 'Create a Set' window with the following interface:

- New Set** button in the top-left corner.
- Create a Set** title at the top.
- Select one or more Filters that objects must match to be members of this set** instruction below the title.
- An object will be a member of this set if it:** followed by two radio buttons:
 - matches ALL the selected filters
 - matches ANY of the selected filters
- Available filters:** A list box containing 16 filter options, all starting with "Command level =":
 - Command level = 211
 - Command level = 300
 - Command level = 500
 - Command level = 510
 - Command level = 520
 - Command level = 521
 - Command level = 530
 - Command level = 600
 - Command level = 700
 - Command level = 701
 - Command level = 710
 - Command level = 750
 - Command level = 800
 - Command level = 801
 - Command level = 802
- Selected filters:** An empty list box.
- Add ->** and **<- Remove** buttons between the available and selected filter lists.
- Manage Filters...** button at the bottom-left, which is highlighted with a red box.

- ___ 4. Define a new filter that references the queue manager **Name** property.

- ___ a. In the **Manage Filters** window, click **Add**.
- ___ b. In the **Add Filter** window, set the **Filter Name** to: QA queue managers

- ___ c. In the Include Queue Managers where section, set the last field in the line to: **QMQA***

 Edit Filter

Filter Name:
QA queue managers

Include Queue Managers where:

Queue manager name like QMQA*

- AND -

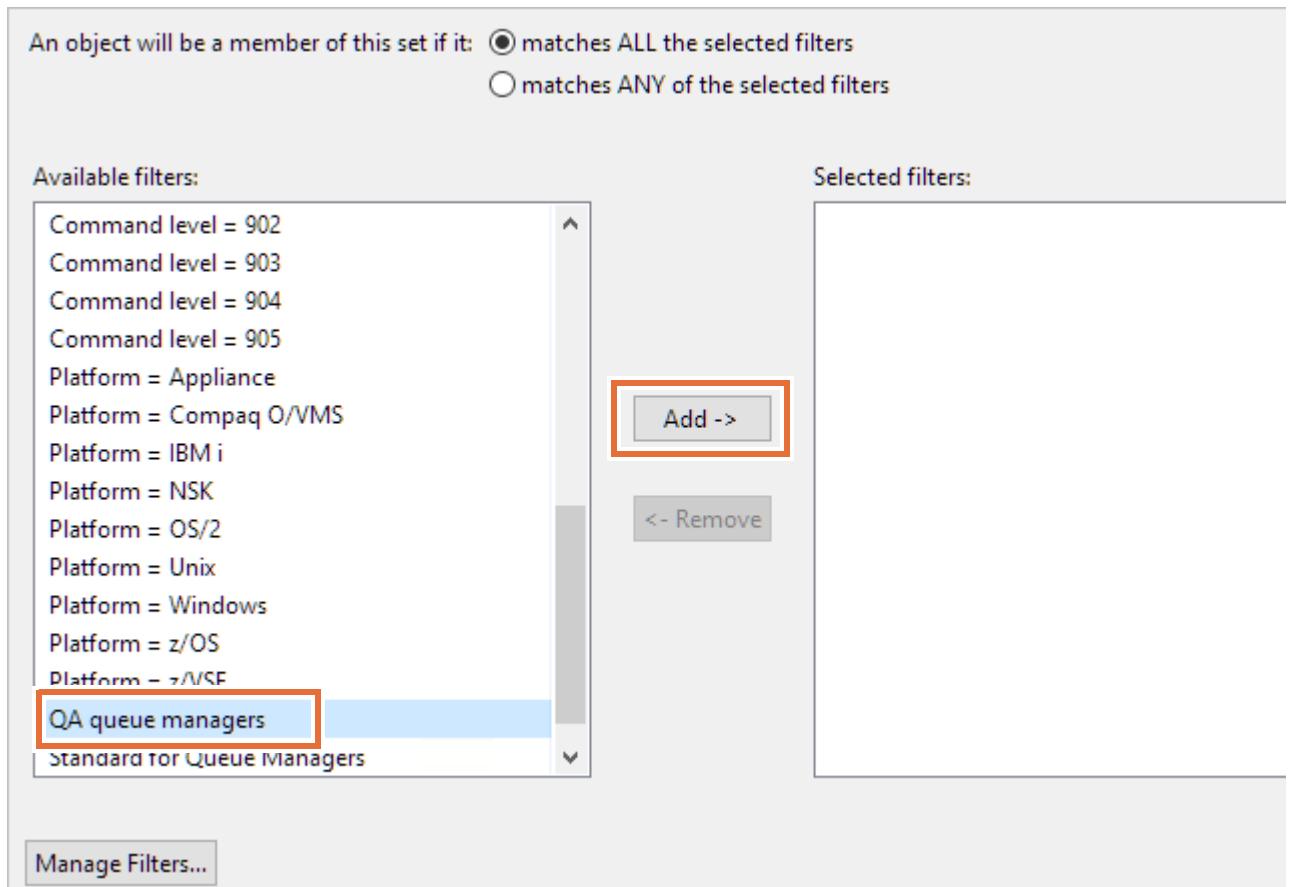
Accounting conn override Select... equal to Enabled

Automatically apply a Column Scheme when this filter is applied
Standard for Queue Managers

- ___ d. Click **OK**.
- ___ e. Click **OK** in the **Manage Filters** window.

- ___ f. In **Available filters** list, scroll down to the newly defined **QA queue managers** filter, and click **Add**, and then click **Finish**.



- ___ 5. In the **Navigator** pane, notice the new **QA.QMGR** entry and expand it to see the queue manager QMQA1.



Section 6. Deleting queue manager sets

You can delete the queue manager sets, as you don't use them in later exercises.

- ___ 1. Delete the queue manager sets.
 - ___ a. In the **Navigator** pane, right-click the **QA.QMGR** set and click **Delete**.
 - ___ b. When prompted to confirm, click **Delete**.
 - ___ c. Right-click the **Production** set and click **Delete**.
 - ___ d. When prompted to confirm, click **Delete**.
- ___ 2. In the **Navigator** pane, right-click **Queue Managers** and click **Sets > Hide Sets**.

Section 7. Exercise cleanup

- ___ 1. In IBM MQ Console, restart **QMPPROD**.
 - ___ a. Open the browser that is running IBM MQ Console (<https://localhost:9443/ibmmq/console>), and sign in with mqadmin/mqadmin.
 - ___ b. In the Local Queue Managers widget, click **QMPPROD** and click the **Start** icon.

Name	Status
QM01	Running
QM02	Running
QMPPROD	Stopped

- ___ 2. Stop queue manager **QM02** and delete it.
 - ___ a. Select **QM02**, click the **Stop** icon, and when prompted to confirm, click **Stop**.

Name	Status
QM01	Running
QM02	Running

- ___ b. When QM02 is stopped, click the **Delete** icon, and when prompted to confirm, click **Delete**.

Name	Status
QM01	Running
QM02	Stopped

**Note**

You should now have two queue managers running.

- ___ 3. Make sure that the queues on QM01 allow PUT requests.
 - ___ a. In the **Queues on QM01** widget, click **QL.A** and click the **Properties** icon.

- ___ b. Confirm that **Enable put** is set to **Allowed**.

- ___ c. Click **Save**.
- ___ d. Repeat these steps for QL.B.
- ___ 4. If QL.A has messages, clear them.
- ___ 5. Minimize IBM MQ Console.
- ___ 6. Close any open command prompts.

End of exercise

Exercise review and wrap-up

The exercise showed you how to use IBM MQ control commands, MQSC commands and command files, and IBM MQ Explorer to manage queues. You also learned how to manage queue manager sets.

Exercise 3. Implementing distributed queuing

Estimated time

00:30

Overview

In this exercise, you learn how to set up a distributed topology for point-to-point message queuing.

Objectives

After completing this exercise, you should be able to:

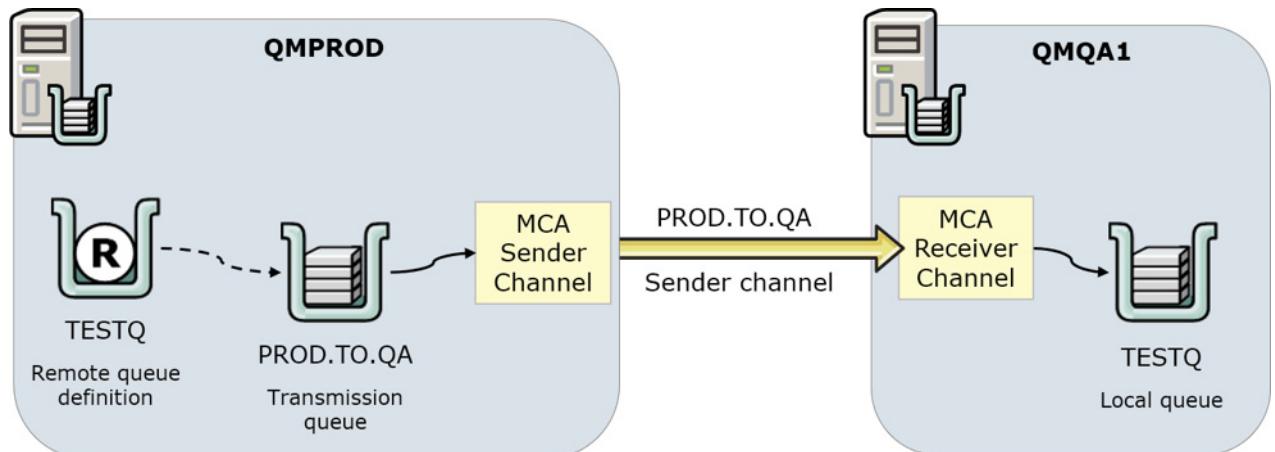
- Set up a distributed topology
- Test point-to-point message queuing

Introduction

This exercise includes these sections:

- [Section 1, "Creating channels between queue managers"](#)
- [Section 2, "Testing queues in a distributed environment"](#)

To simulate a distributed environment, you set up the topology that is depicted here.



Requirements

This exercise uses the queue managers that you created in [Exercise 2, "Working with IBM MQ administration tools"](#).

Section 1. Creating channels between queue managers

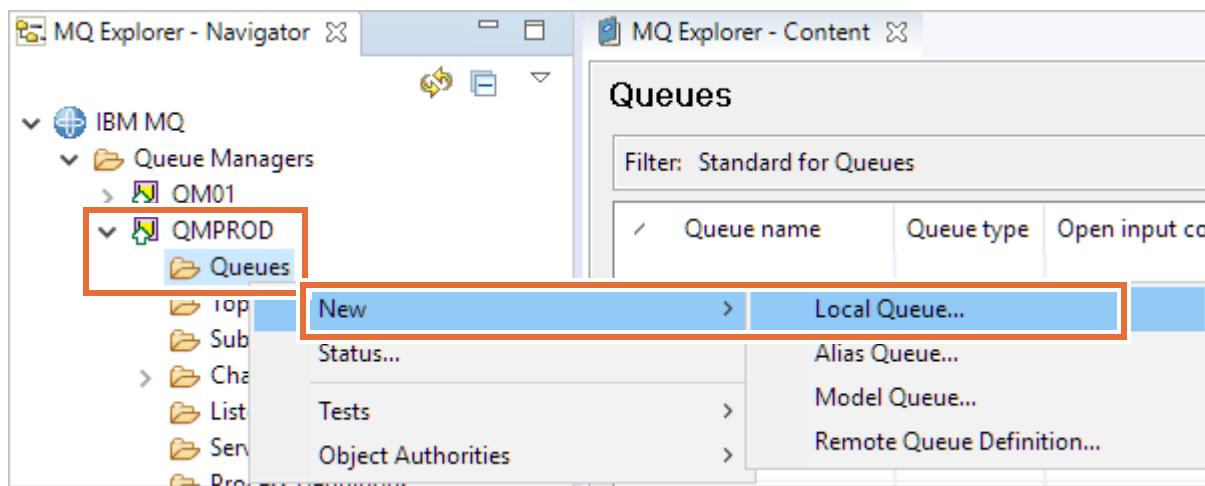
In this part of the exercise, you set up the distributed environment by creating these objects.

Local queue manager	QMPROD
Sender channel	PROD.TO.QA
Transmission queue	PROD.TO.QA
Remote queue definition	TESTQ

Remote queue manager	QMQA1
Receiver channel	PROD.TO.QA
Local queue	TESTQ

1.1. Create the transmission queue

- ___ 1. In MQ Explorer, create a transmission queue on QMPROD.
 - ___ a. In the **Navigator** pane, right-click the **QMPROD > Queues** folder, and select **New > Local Queue**.



Information

A *transmission queue* is a special type of local queue that is associated with a sender channel. The transmission queue holds messages safely until they are transferred to the queue manager on the other end of the channel.

A typical nomenclature for naming transmission queues and channels is:

<sourceqm>.TO.<targetqm>

For this exercise, you name the transmission queue: PROD.TO.QA

-
- ___ b. In the **Name** field, enter **PROD.TO.QA** and click **Next**.

- ___ c. Change the **Usage** property to **Transmission** and click **Finish**.

New Local Queue

Change properties

Change the properties of the new Local Queue

General

General	Queue name: PROD.TO.QA
Extended	Queue type: Local
Cluster	Description:
Triggering	Put messages: Allowed
Events	Get messages: Allowed
Storage	Default priority: 0
Statistics	Default persistence: Not persistent
Scope:	Queue manager
Usage:	Transmission

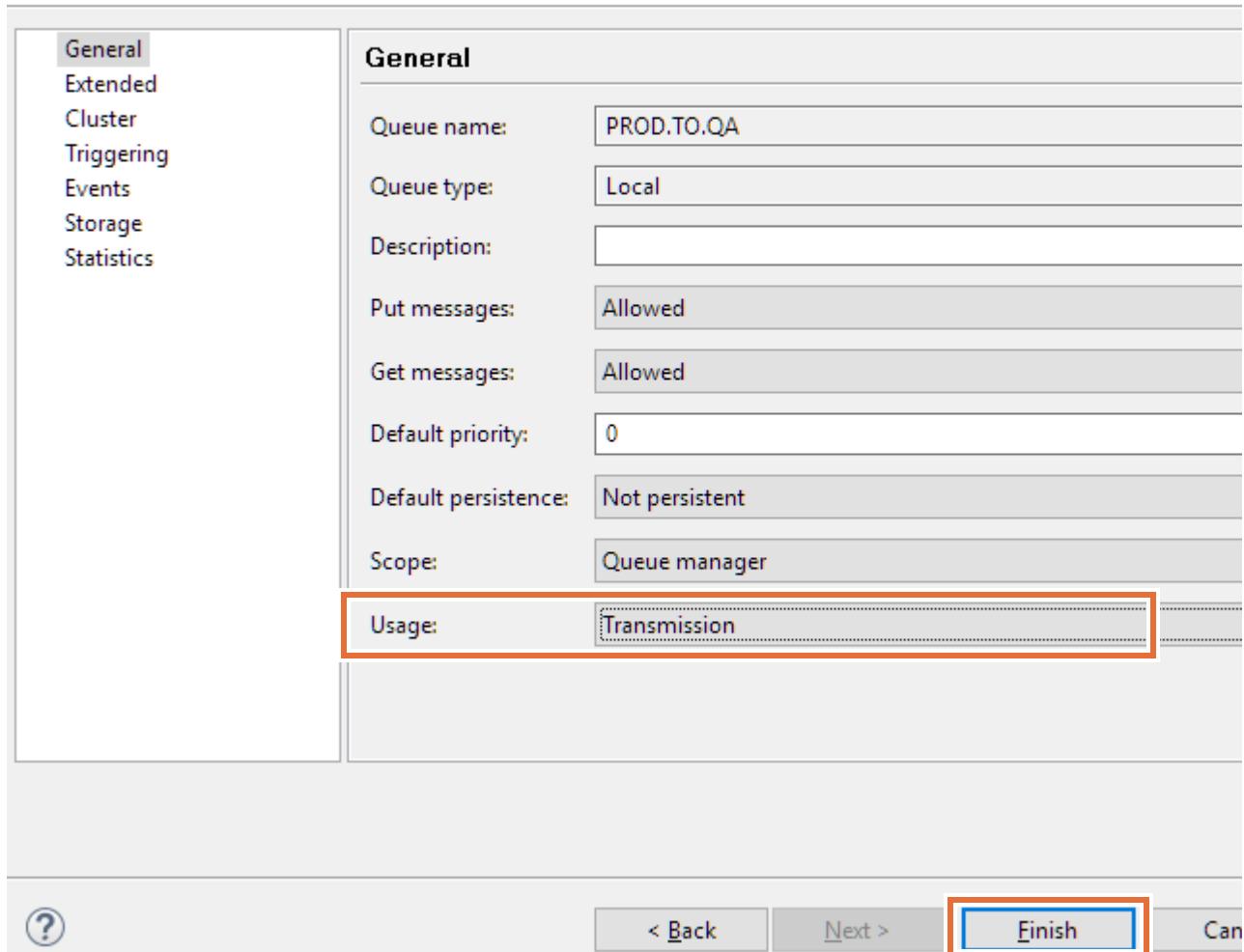
?

< Back

Next >

Finish

Cancel



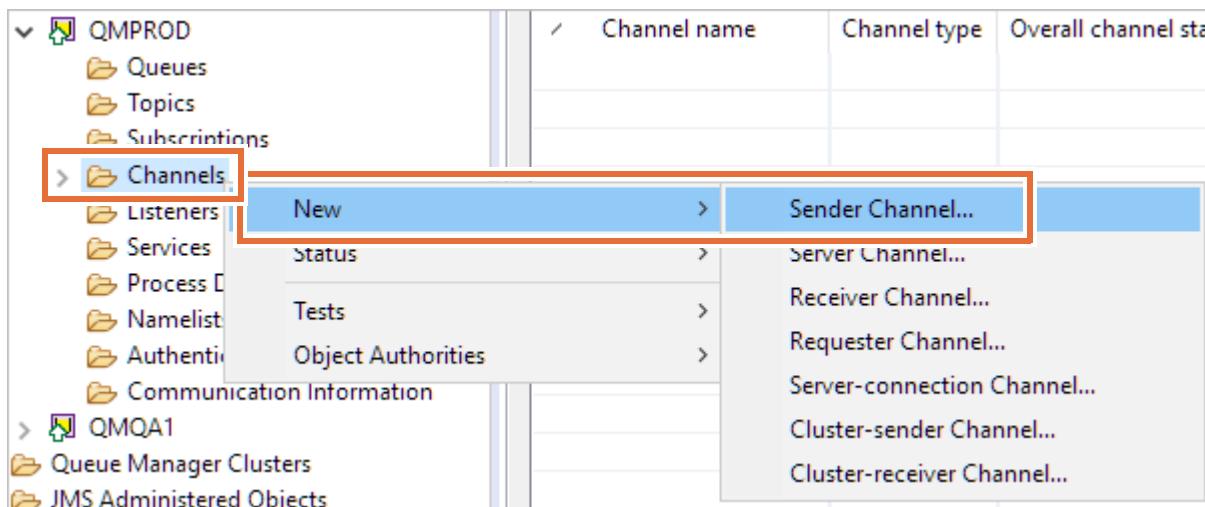
- ___ d. Click **OK** to close the success message.



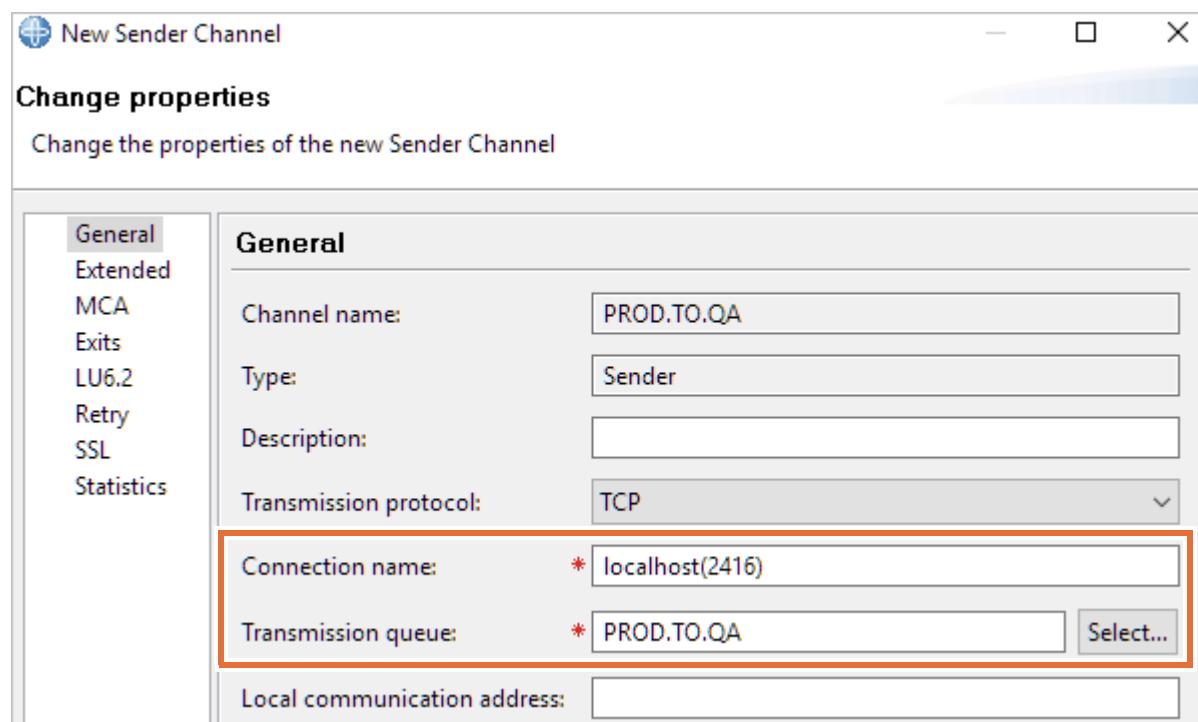
The transmission queue is associated with the sender channel.

1.2. Create the sender channel

- 1. In the Navigator pane, right-click the **QMPPROD > Channels** folder and click **New > Sender Channel**.



- 2. In the **Name** field, enter `PROD.TO.QA` and click **Next**.
- 3. On the “Change properties” page, set the following properties:
- **Connection name:** `localhost(2416)`
This value must match the host and listener port for the queue manager that you want to connect to.
 - **Transmission queue:** `PROD.TO.QA`



- 4. Click **Finish**.

- ___ 5. Click **OK** to close the confirmation message.

In the Channels view, you see the details of the new channel, including the channel type, status, connection host and port, and the name of transmission queue.

Channel name	Channel type	Overall channel status	Conn name	Transmission queue
PROD.TO.QA	Sender	Inactive	localhost(2416)	PROD.TO.QA

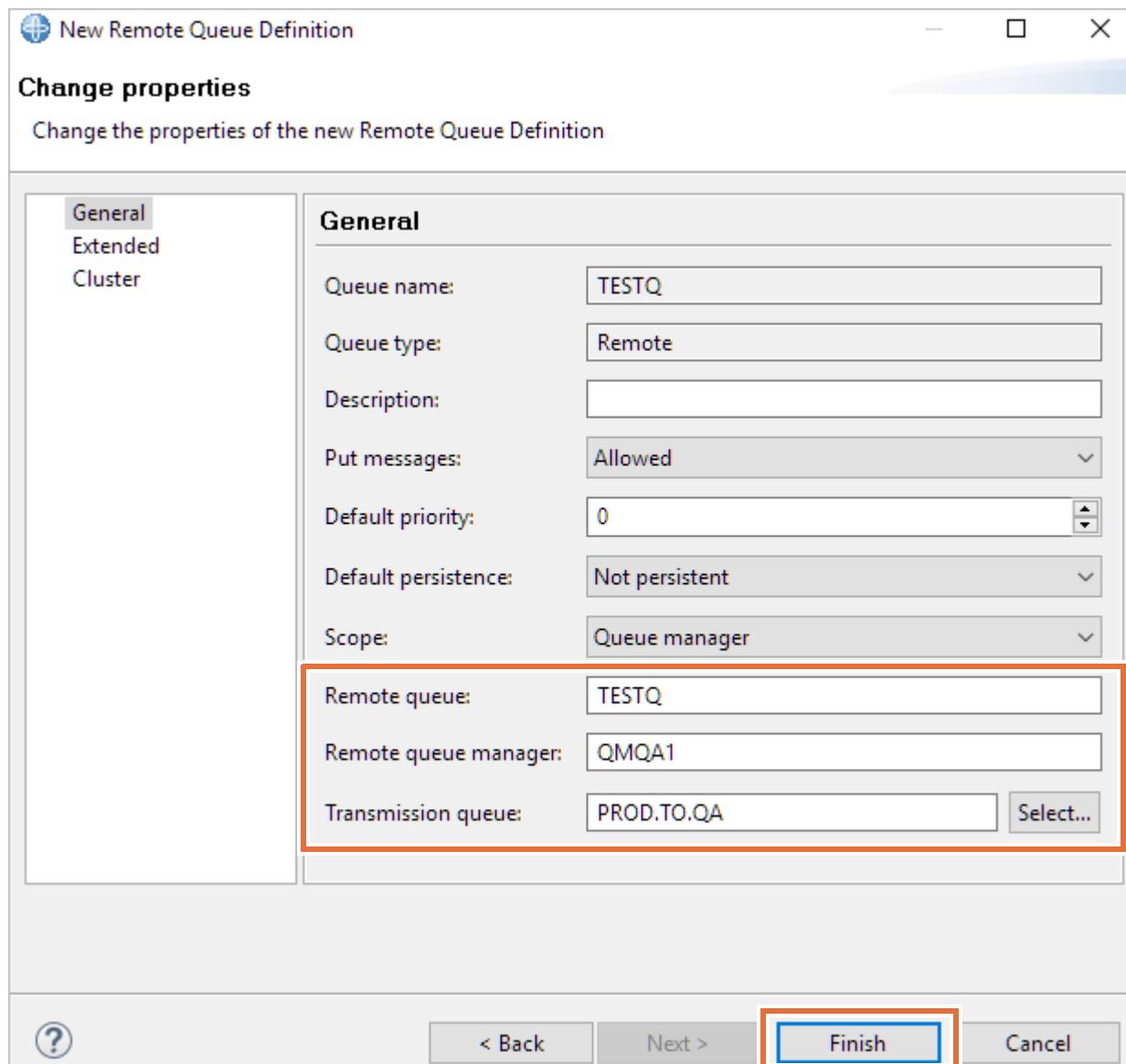
1.3. Create the remote queue definition

The final task on the QMPROD queue manager is to create a remote queue definition. This definition acts as an alias of the actual target queue that you create on the QMQA1 queue manager.

- ___ 1. Right-click the **QMPROD > Queues** folder and click **New > Remote Queue Definition**.

- ___ 2. Set the name of the remote queue definition to **TESTQ** and click **Next**.
 ___ 3. Set the following remote queue properties to specify the target queue, where the target queue resides, and the transmission queue to reach that queue manager:
- **Remote queue:** TESTQ
 - **Remote queue manager:** QMQA1
 - **Transmission queue:** PROD.TO.QA

4. Click **Finish**.



5. Click **OK** to close the confirmation message.

The remote queue definition is listed in the Queues view.

The screenshot shows the MQ Explorer interface with two panes. The left pane is the navigation tree:

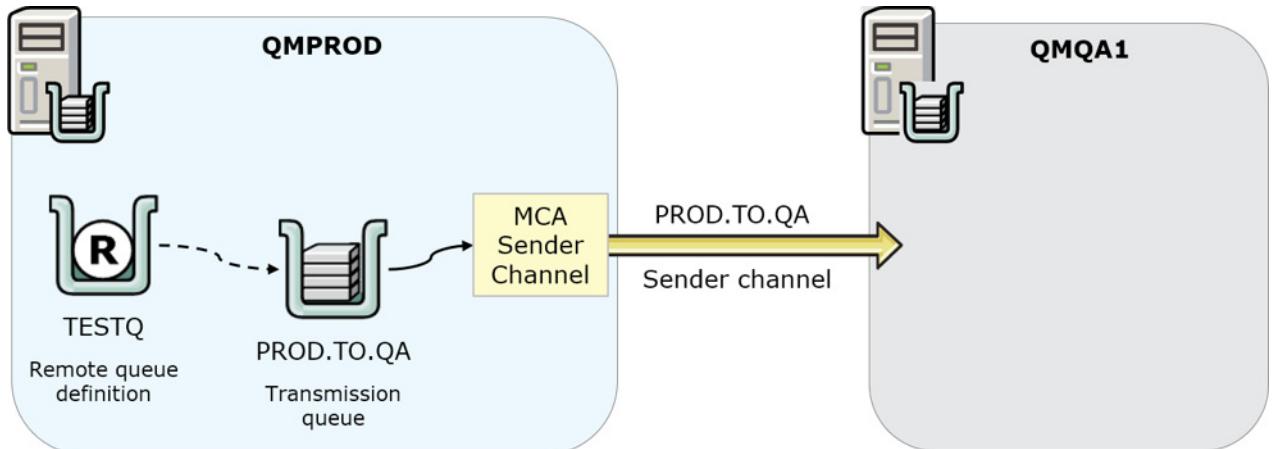
- IBM MQ
 - Queue Managers
 - QM01
 - QMPROD
 - Queues
 - Topics
 - Subscriptions

The right pane is the 'Content' tab, specifically the 'Queues' view. It shows a table with the following data:

Queue name	Queue type	Open input count	Open output count
PROD.TO.QA	Local	0	0
TESTQ	Remote	0	0

A row for 'TESTQ' is highlighted with an orange border.

The local queue manager, QMPROD, is now set up with a remote queue definition, TESTQ. TESTQ points to a transmission queue called PROD.TO.QA, which is associated with a sender channel also called PROD.TO.QA that connects to QMQA1.

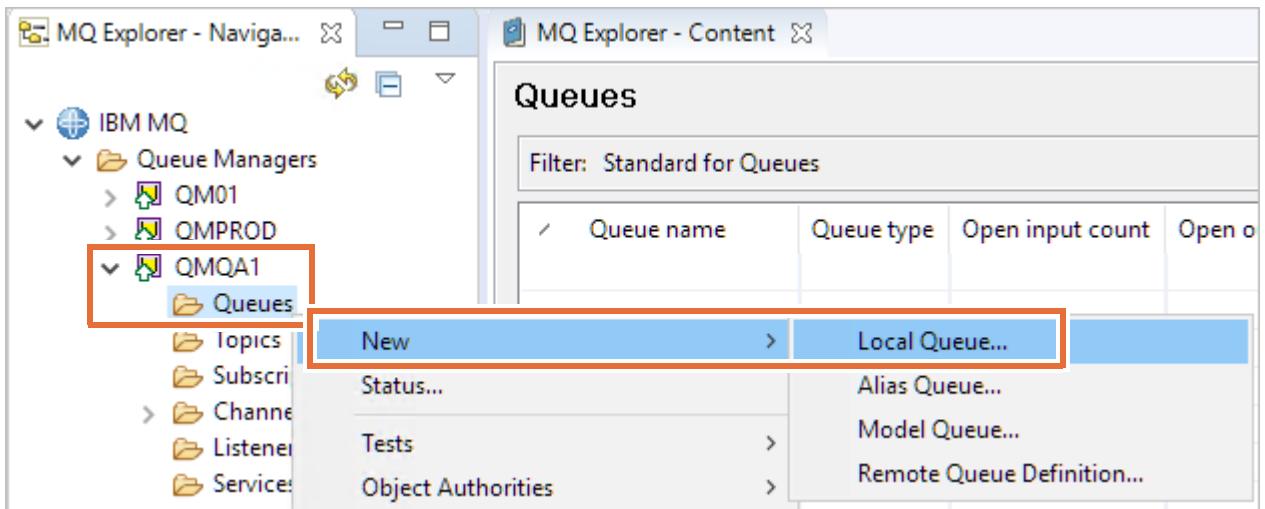


Next, you create the corresponding definitions on QMQA1.

1.4. Create the target queue on QMQA1

The TESTQ remote queue definition that you created is an alias for the actual local queue you are about to create.

- 1. In the **Navigator** pane, right-click the **QM QA1 > Queues** folder and click **New > Local Queue**.

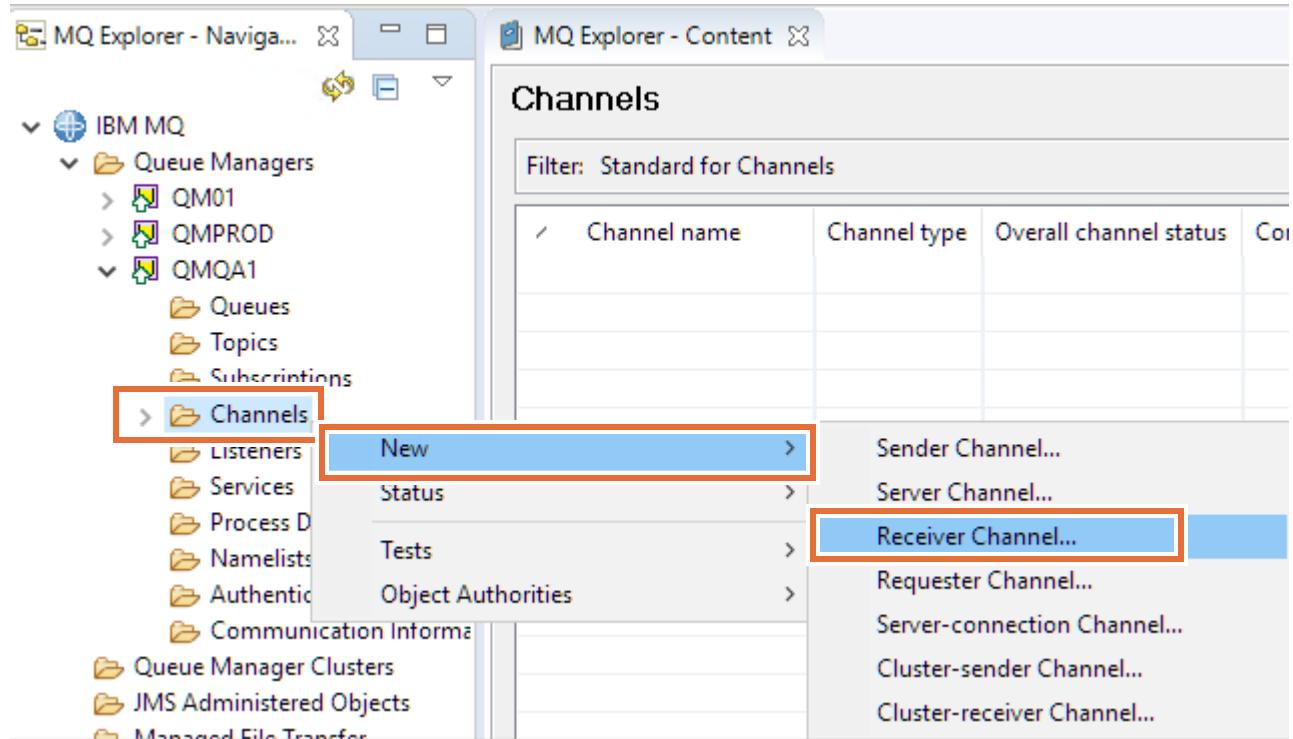


- 2. In the **Name** field, enter **TESTQ** and click **Finish**.
- 3. Click **OK** to close the confirmation message.

1.5. Create the receiver channel on QMQA1

The receiver channel must have the same name as the corresponding sender channel that you created.

- 1. Right-click the **QMQA1 > Channels** folder and click **New > Receiver Channel**.

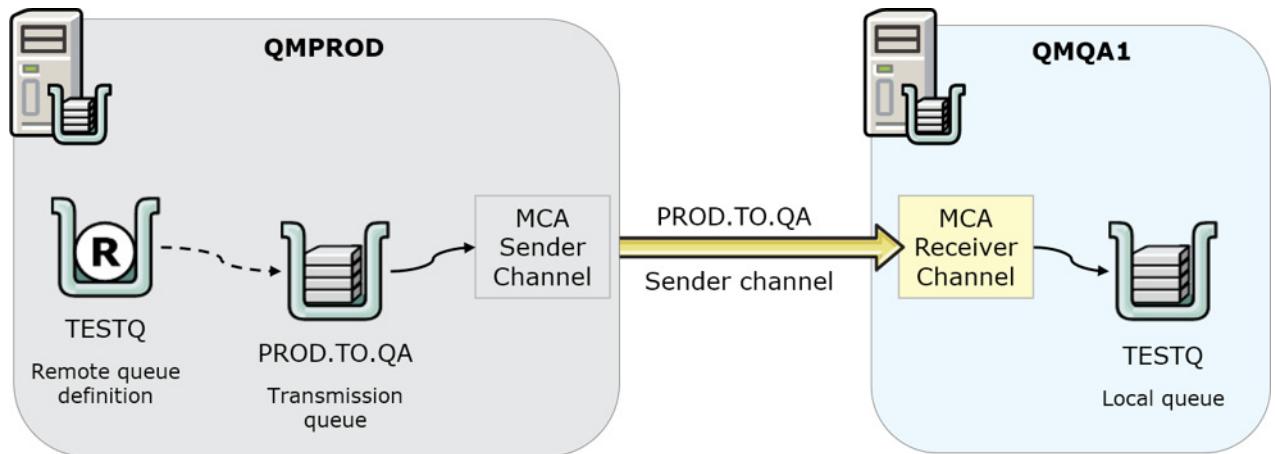


- 2. In the **Name** field, enter `PROD.TO.QA` and click **Finish**.
- 3. Click **OK** to close the confirmation message.

In the Channels view, you see the details of the channel you just created, including the channel type and status.

Channel name	Channel type	Overall channel status	Conn name
<code>PROD.TO.QA</code>	Receiver	Inactive	

The remote queue manager, QMQA1, is now set up with a target queue, TESTQ. You also created a receiver channel, PROD.TO.QA, as a partner for the sender channel on QMPROD.



Next, you test your configuration.

Section 2. Testing queues in a distributed environment

For this test, you put messages to the TESTQ remote queue definition on queue manager QMPROD.

2.1. Send some test messages

- ___ 1. Open a command prompt and start the amqspput sample program.

```
amqspput TESTQ QMPROD
```

- ___ 2. Enter 3 lines of text, such as:

```
test msg1
test msg2
test msg3
```

```
C:\ Administrator: Command Prompt - amqspput TESTQ QMPROD
Microsoft Windows [Version 10.0.14393]
(c) 2016 Microsoft Corporation. All rights reserved.

C:\Users\Administrator>amqspput TESTQ QMPROD
Sample AMQSPUT0 start
target queue is TESTQ
test msg1
test msg2
test msg3
```

- ___ 3. Press Enter to end the program.

2.2. Verify delivery of the messages

- ___ 1. Return to IBM MQ Explorer.
- ___ 2. In the **Navigator** pane, click the **QMPROD > Queues** folder.
- ___ 3. Notice the **Current queue depth** value for TESTQ is empty and disabled.

Queue name	Queue type	Open input count	Open output count	Current queue depth
PROD.TO.QA	Local	0	1	3
TESTQ	Remote			



Questions

Why is the current depth queue value unavailable for TESTQ?

Recall: The TESTQ remote queue definition is not an actual queue. TESTQ is a reference to the transmission queue associated with the channel that points to the queue manager where the actual TESTQ instance resides.



Questions

Why are the 3 messages sitting on the transmission queue? Why were they not transferred to the QMQA1 queue manager?

- 4. Click the **QMPROD > Channels** folder and check the channel status for PROD.TO.QA.

Channel name	Channel type	Overall channel status
PROD.TO.QA	Sender	Inactive

The channel is inactive. In order to move messages, the channel must be running. You can configure the channel to start automatically when a message arrives on its transmission queue. For this exercise, you start the channel manually.

2.3. Start the channel

- 1. In the Channels view, right-click PROD.TO.QA and click **Start**.

- ___ 2. Click **OK** to close the confirmation message.
- ___ 3. Notice that the channel status changes to **Running**.
- ___ 4. With the channel now running, click the **QMPPROD > Queues** folder again to check current queue depth for PROD.TO.QA.

The queue depth is 0.

2.4. Confirm transmission of the message

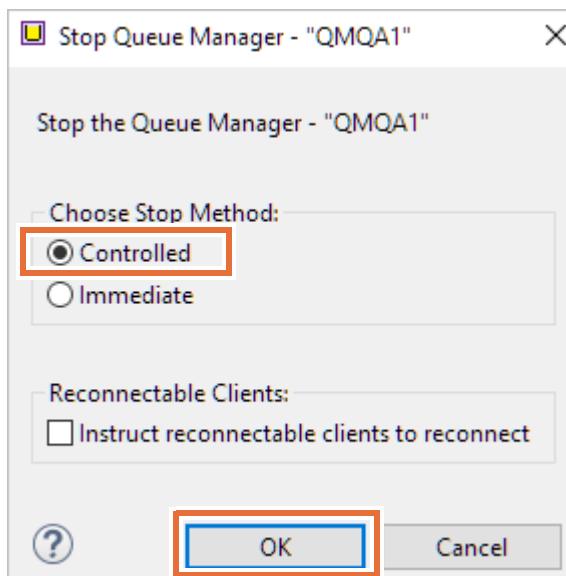
- ___ 1. Check the queue on QMQA1.
 - ___ a. In the **Navigator** pane, click the **QMQA1 > Queues** folder.
 - ___ b. Notice the queue depth value for queue TESTQ is 3.
- To confirm that these are the messages that you put by using amqspput, you use the amqsget sample program.
- ___ 2. Retrieve your messages from the TESTQ local queue on QMQA1 by entering the following command in a command prompt.

```
amqsget TESTQ QMQA1
```

The sample program returns the messages that you put on the queue, and clears the TESTQ queue.

Section 3. Exercise cleanup

- ___ 1. Stop the QMQA1 queue manager.
 - ___ a. In the **Navigator** pane, right-click **QMQA1** and click **Stop > Controlled**.
 - ___ b. Leave the **Controlled** option selected and click **OK**.



- ___ 2. Repeat [Step 1](#) for QMPROD.
- ___ 3. Close all the command prompts.

End of exercise

Exercise review and wrap-up

The first part of the exercise you created channels between two queue managers. You then used the IBM MQ sample programs to test the connection between the queue managers.

Exercise 4. Connecting an IBM MQ client

Estimated time

01:00

Overview

In this exercise, you configure your system to act as a client that is connected to an IBM MQ server. You use various methods to gain experience with the client connectivity methods that are available in IBM MQ.

Objectives

After completing this exercise, you should be able to:

- Create a server connection channel to support client connections
- Use a URL to specify the location of the client connection definition table
- Use the MQSERVER environment variable to specify a client connection channel
- Use a client configuration file to specify a client connection channel

Introduction

This exercise includes these sections:

- [Section 1, "Set up the lab environment"](#)
- [Section 2, "Method 1: Client connection through the CCDT"](#)
- [Section 3, "Method 2: Client connection by using the MQSERVER environment variable"](#)
- [Section 4, "Method 3: Client connection by using the client configuration file"](#)

Requirements

This exercise uses queue manager QM01. Make sure that you complete [Exercise 2, "Working with IBM MQ administration tools"](#) before starting this exercise.

Section 1. Set up the lab environment

1.1. Configuring the queue manager to act as a server

In this exercise, the queue manager QM01 is configured to act as a server for the IBM MQ clients. You use the MQ client sample programs to connect the MQ client to the queue manager QM01.



Important

For this exercise, you disable channel authentication on QM01.

You also use MUSR_MQADMIN for the MCAUSER user ID. In a real-world implementation, you would create an administrative user ID for MQ and use that ID for the MCAUSER. Never use MUSR_MQADMIN or “mqm” in a real-world implementation.

You learn more about security later in this course.

- 1. To eliminate any security errors and security configuration in this exercise, disable the use of channel authentication and connection authentication on QM01.
 - a. Open a command prompt and start an MQSC session for QM01.

```
runmqsc QM01
```

- b. Disable security by entering these commands.

Note: You can copy all these commands together and paste them into the command prompt. Make sure to press Enter after the last line.

```
ALTER QMGR CHLAUTH DISABLED
ALTER AUTHINFO (SYSTEM.DEFAULT.AUTHINFO.IDPWOS) +
  AUTHTYPE (IDPWOS) CHCKCLNT (NONE)
REFRESH SECURITY
```

- 2. On queue manager QM01, define a server-connection (SVRCONN) channel so that MQ clients can connect to the queue manager.

```
DEFINE CHL (QM01_CLNT) CHLTYPE (SVRCONN) TRPTYPE (TCP) +
  MCAUSER (MUSR_MQADMIN)
```

- 3. Verify the channel definition.

```
DIS CHL (QM01_CLNT)
```

- 4. Define the SVRCONN and set the MCAUSER to MUSR_MQADMIN on QM01:

```
DEFINE CHL (FILEURL) CHLTYPE (SVRCONN) TRPTYPE (TCP) +
  MCAUSER (MUSR_MQADMIN)
```

- 5. Verify the channel definition.

```
DIS CHL (FILEURL)
```

- 6. Verify that the QM01 listener is running on port 1414.

```
DIS LSSTATUS (LISTENER.TCP) end
```

If the listener is not running, start the listener by typing:

START LISTENER(LISTENER.TCP)

- ___ 7. End the MQSC session by typing: end

1.2. Monitor channel status on IBM MQ Console

During this exercise, you use IBM MQ Console to confirm that your channels are running.

- ___ 1. Open IBM MQ Console in a browser (<https://localhost:9443/ibmmq/console>) and sign in with mqadmin/mqadmin.
- ___ 2. Add a widget to monitor channels.
- ___ a. In the upper right, click **Add widget**.
- ___ b. On the Add a new widget page, select **QM01** in the queue managers list and click **Channels**.

The screenshot shows the 'Add a new widget' dialog in the IBM MQ Console. At the top, it says 'Add a new widget'. Below that, there are two tabs: 'Local Queue Managers' (selected) and 'Manage local queue managers'. Underneath, there's a 'Chart' tab labeled 'Monitor your MQ platform'. A horizontal line separates this from the main configuration area. The main area has a heading 'Add a widget to display MQ object information for the specified queue manager'. It asks 'Queue manager:' and shows 'QM01' selected in a dropdown menu, which is highlighted with an orange box. Below this, there are several categories: 'Queues' (Configure destinations for messages), 'Topics' (Administrative objects for assigning attributes to topics), 'Listeners' (Configure processes to accept network requests), and 'Channels' (Queue manager communication paths). The 'Channels' category is highlighted with a blue box. At the bottom, there's another horizontal line and the 'Client-connection Channels' (Client connectivity details) section. At the very bottom right of the dialog is a 'Close' button.

The widget opens automatically. You see the channels that you created with a status of **Inactive**.

Name	Type	Overall channel sta
FILEURL	Server-connection	● Inactive
QM01_CLNT	Server-connection	● Inactive

- ___ 3. Keep the IBM MQ Console browser open but minimized or in the background.

Section 2. Method 1: Client connection through the CCDT

In this part of the exercise, you connect to the queue manager QM01 by using a client connection channel that is defined on the client. The client locates the CCDT by using the file URL that is specified in the MQCCDTURL environment variable.

2.1. Set up a client connection with MQCCDTURL

- ___ 1. Open Windows Explorer, and create a folder in the C:\labfiles directory that is named: **ccdturl**

You use this folder to store the client CCDT table.

- ___ 2. On the client CCDT table, define a client connection with the same name as the server connection on QM01 (FILEURL).

- ___ a. Open a separate command prompt and start a client MQSC session:

```
runmqsc -n
```

The command should return:

```
Starting local MQSC for 'AMQCLCHL.TAB'.
```

- ___ b. Verify that the CCDT table on the client is empty by typing:

```
DIS CHL(*)
```

The command should return the message:

```
IBM MQ object * not found.
```

- ___ c. Create a FILEURL client connection channel that connects to QM01:

```
DEF CHL(FILEURL) CHLTYPE(CLNTCONN) TRPTYPE(TCP) +
CONNAME('localhost(1414)')
```

Note: At this point, the CCDT table is created in the C:\ProgramData\IBM\MQ directory.

- ___ d. Verify that the channel is now defined in the CCDT table.

```
DIS CHL(*)
```

This time, the command should return:

CHANNEL(FILEURL)	CHLTYPE(CLNTCONN)
------------------	-------------------

- ___ e. End the MQSC session by typing: end

- ___ 3. Go to Windows Explorer and move the AMQCLCHL.TAB CCDT table file from the C:\ProgramData\IBM\MQ directory to the C:\labfiles\ccdturl directory.

2.2. Test the connection

- ___ 1. In a command prompt, type the following command to set the MQCCDTURL environment variable to point to the CCDT that contains the of FILEURL channel definition.

```
SET MQCCDTURL=file:///C:/labfiles/ccdturl/AMQCLCHL.TAB
```

- __ 2. In the same command prompt, start the `amqsputc` sample program to test your connection.
`amqsputc QL.A`
- Note:** You do not need to specify the queue manager parameter because the `amqsputc` sample connects to the queue manager identified by the client channel definition file.
- __ 3. Keep the `amqsputc` sample running. You put messages after the next step.
- __ 4. Open the IBM MQ Console browser to check that the server-connection channel FILEURL is running. You might need to refresh the widget by clicking the **Refresh** icon in the upper right.

Name	Type	Overall channel status
FILEURL	Server-connection	Running
QM01_CLNT	Server-connection	Inactive

Note: You can also check the channel status in MQ Explorer by right-clicking the channel in the **Channels** view and clicking **Status**.

- __ 5. Return to the command prompt where `amqsputc` is running to put some messages to QL.A on QM01, and end the sample program by pressing Enter twice.
 - __ a. Enter some lines of text, for example:


```
message 1
message 2
message 3
```
 - __ b. After the last message, press Enter twice to end the program.

- ___ 6. In IBM MQ Console, verify that your messages are on the QL.A queue on QM01. (You might need to refresh the widget.)

Queues on QM01		
Name	Queue type	Queue depth
QL.A	Local	3
QL.B	Local	0

- ___ 7. Use the `amqsbcgc` client program to browse the messages on the server queue.
- ___ a. In the same command prompt where you ran `amqsputc` (and where you set the `MQCCDTURL` variable), enter this command:
- ```
amqsbcgc QL.A
```
- \_\_\_ b. Scroll up through the results to find the value of `ReplyToQmgr`, which shows you the queue manager to which the client connected.
- \_\_\_ 8. Use the `amqsgetc` program to get the messages from QL.A and empty the queue.
- ```
amqsgetc QL.A
```
- ___ 9. Clear the `MQCCDTURL` environment variable by typing:
- ```
SET MQCCDTURL=
```

## Section 3. Method 2: Client connection by using the MQSERVER environment variable

In this part of the exercise, you use the MQSERVER environment variable to specify a client connection channel.

- 1. Use the MQSERVER environment variable to provide a client-connection channel definition to connect to the server queue manager (QM01) by using the QM01\_CLNT channel.
- a. In a command prompt, set the MQSERVER environment variable.

```
SET MQSERVER=QM01_CLNT/TCP/localhost (1414)
```

- b. Confirm that the `MQSERVER` variable was set correctly by typing:

```
echo %MQSERVER%
```

The value that is returned matches what you entered:

```
QM01_CLNT/TCP/localhost (1414)
```

- 2. In the same command prompt that you set the `MQSERVER` environment variable, start the `amqsputc` sample program to test the client connection.
- 3. Return to IBM MQ Console and refresh the Channels on QM01 widget to verify that the QM01\_CLNT channel is running on QM01.

| Name      | Type              | Overall channel status |
|-----------|-------------------|------------------------|
| FILEURL   | Server-connection | ● Inactive             |
| QM01_CLNT | Server-connection | ● Running              |



### Note

**Note:** To check the channel status in MQ Explorer, click the **QM01 > Channels** folder and check the **Overall channel status** value in the **Channels** view.

- \_\_\_ 4. In the command prompt where `amqsputc` is running, put some messages to QL.A. For example:

```
MQSERVER message 1
MQSERVER message 2
MQSERVER message 3
```
- \_\_\_ 5. In IBM MQ Console, refresh the Queues on QM01 widget to verify that your messages are on QL.A.
- \_\_\_ 6. Use the `amqsbcgc` client program to browse the messages on the server queue.
  - \_\_\_ a. In the command prompt where you ran `amqsputc`, enter this command:

```
amqsbcgc QL.A
```
  - \_\_\_ b. Scroll up through the results to find the value of **ReplyToQmgr**, which shows you the queue manager (QM01) to which the client connected.
- \_\_\_ 7. Use the `amqsgetc` program to get the messages from QL.A and empty the queue.

```
amqsgetc QL.A
```
- \_\_\_ 8. Clear the MQSERVER environment variable.

```
SET MQSERVER=
```

## Section 4. Method 3: Client connection by using the client configuration file

In this part of the exercise, you use the client configuration file to specify information about the client channels.

For this exercise, the `mqclient.ini` client configuration file is provided for you.

### 4.1. Edit the `mqclient.ini` client configuration file

- 1. Edit the `mqclient.ini` client configuration file in Notepad.
  - a. In Windows Explorer, go to the `C:\labfiles\Lab04-client` directory, and open the `mqclient.ini` in Notepad.
  - b. Set the `ServerConnectionParms` attribute to specify the connection to the `QM01_CLNT` channel on QM01:  
`ServerConnectionParms=QM01_CLNT/TCP/localhost(1414)`
  - c. Save and close the file.

### 4.2. Test the connection

- 1. In a command prompt, set the `MQCLNTCF` environment variable to the location of the client configuration file.  
`SET MQCLNTCF=C:\labfiles\Lab04-client\mqclient.ini`
- 2. Start the `amqsputc` to test the client connection.  
`amqsputc QL.A`
- 3. Return to IBM MQ Console and refresh the **Channels on QM01** widget to verify that the `QM01_CLNT` channel is running on QM01.

The screenshot shows the 'Channels on QM01' widget in the IBM MQ Console. The interface includes a search bar, a 'Create' button with a plus sign, and three icons (refresh, settings, and delete). The main table lists two channels:

| Name      | Type              | Overall channel status                       |
|-----------|-------------------|----------------------------------------------|
| FILEURL   | Server-connection | <span style="color: red;">●</span> Inactive  |
| QM01_CLNT | Server-connection | <span style="color: green;">●</span> Running |

- \_\_\_ 4. In the command prompt where `amqsputc` is running, put some messages to QL.A. For example:

```
mqclient message 1
mqclient message 2
mqclient message 3
```
- \_\_\_ 5. In IBM MQ Console, refresh the **Queues on QM01** widget to verify that your messages are on QL.A.
- \_\_\_ 6. Return to the command prompt (where you ran `amqsputc`) and use the `amqsbcgc` client program to browse the messages on the server queue.

```
amqsbcgc QL.A
```

The value of **ReplyToQmgr** in the MQMD shows you the queue manager (QM01) to which the client connected.

- \_\_\_ 7. Use the `amqsgetc` program to get the messages from QL.A and empty the queue.

```
amqsgetc QL.A
```

- \_\_\_ 8. Clear the MQCLNTCF environment variable.

```
SET MQCLNTCF=
```

### 4.3. Exercise cleanup

- \_\_\_ 1. In IBM MQ Console, stop queue manager QM01.
- \_\_\_ 2. Close IBM MQ Console.
- \_\_\_ 3. Close all the command prompts.

## End of exercise

## Exercise review and wrap-up

The first part of the exercise, you configured your system to act as a client that is connected to an IBM MQ server. You then tried the various client connectivity methods that are available in IBM MQ.

# Exercise 5. Implementing a basic cluster

## Estimated time

02:00

## Overview

In this exercise, you create a cluster of four queue managers. You then test the cluster by using the cluster mechanism to send messages between queues on all queue managers in the cluster.

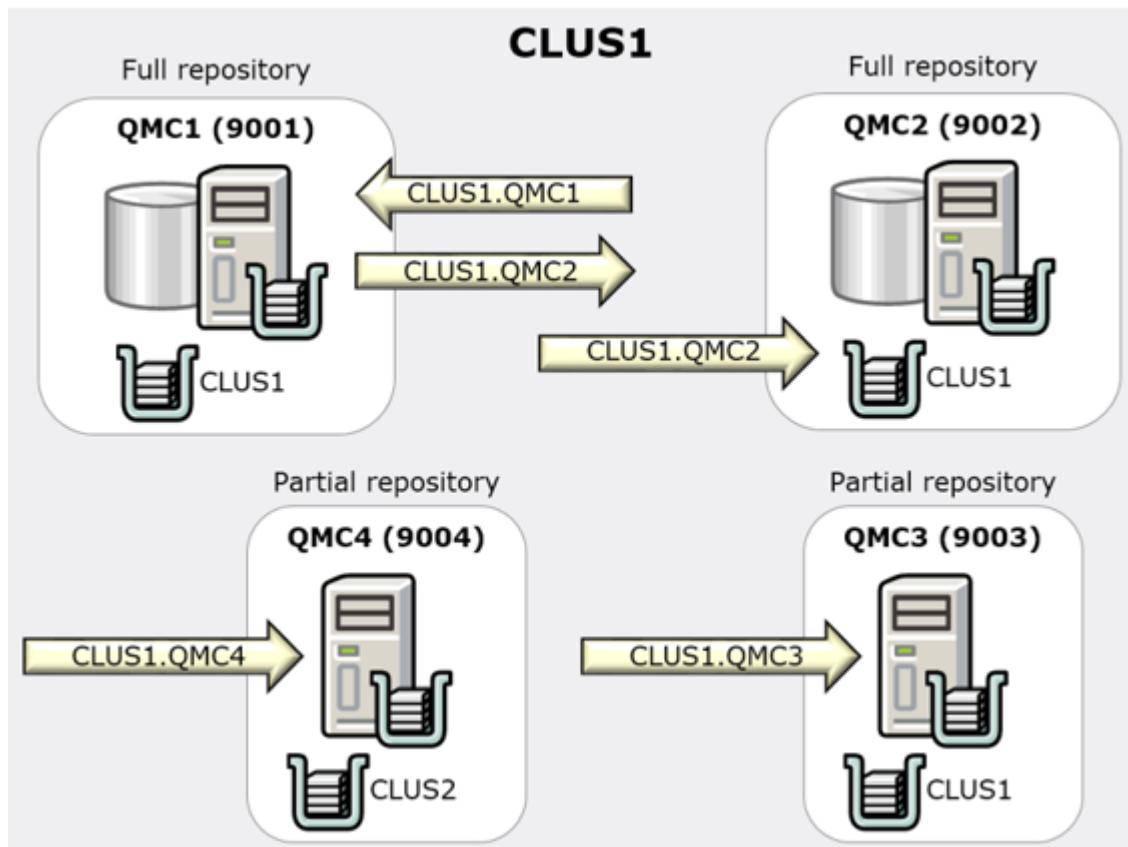
## Objectives

After completing this exercise, you should be able to:

- Create a simple queue manager cluster
- Test the cluster environment

## Introduction

In this exercise, you create a basic cluster that is named CLUS1 with four queue managers:



The cluster is composed of these MQ objects.

| <b>Cluster: CLUS1</b>                                                                                                                                                                                                      |                                                                                                                                                                           |
|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Full repository: QMC1 (9001)</b> <ul style="list-style-type: none"> <li>Cluster-sender channel: <b>CLUSQ1.QMC2</b></li> <li>Cluster-receiver channel: <b>CLUSQ1.QMC1</b></li> <li>Local queue: <b>CLUSQ1</b></li> </ul> | <b>Partial repository: QMC3 (9003)</b> <ul style="list-style-type: none"> <li>Cluster-receiver channel: <b>CLUSQ1.QMC3</b></li> <li>Local queue: <b>CLUSQ1</b></li> </ul> |
| <b>Full repository: QMC2 (9002)</b> <ul style="list-style-type: none"> <li>Cluster-receiver channel: <b>CLUSQ1.QMC2</b></li> <li>Local queue: <b>CLUSQ1</b></li> </ul>                                                     | <b>Partial repository: QMC2 (9004)</b> <ul style="list-style-type: none"> <li>Cluster-receiver channel: <b>CLUSQ1.QMC4</b></li> <li>Local queue: <b>CLUSQ2</b></li> </ul> |

This exercise includes these sections:

- [Section 1, "Defining the cluster"](#)
- [Section 2, "Managing the cluster workload"](#)

## Requirements

This exercise requires that you use the `data.txt` file in the `C:\labfiles\Lab05-cluster` directory.

## Section 1. Defining the cluster

In this part of the exercise, you use IBM MQ Explorer to define the cluster queue managers, channels, and clustered queues. You also use IBM MQ Explorer to verify your configuration.

### 1.1. Create the queue managers for the cluster

- \_\_\_ 1. In MQ Explorer, make sure that all queue managers are stopped.
- \_\_\_ 2. Create the following queue managers as described in the table. Accept the default values for properties that are not listed in the table.

| Queue manager name | Dead-letter queue        | Listener port number |
|--------------------|--------------------------|----------------------|
| QMC1               | SYSTEM.DEAD.LETTER.QUEUE | 9001                 |
| QMC2               | SYSTEM.DEAD.LETTER.QUEUE | 9002                 |
| QMC3               | SYSTEM.DEAD.LETTER.QUEUE | 9003                 |
| QMC4               | SYSTEM.DEAD.LETTER.QUEUE | 9004                 |



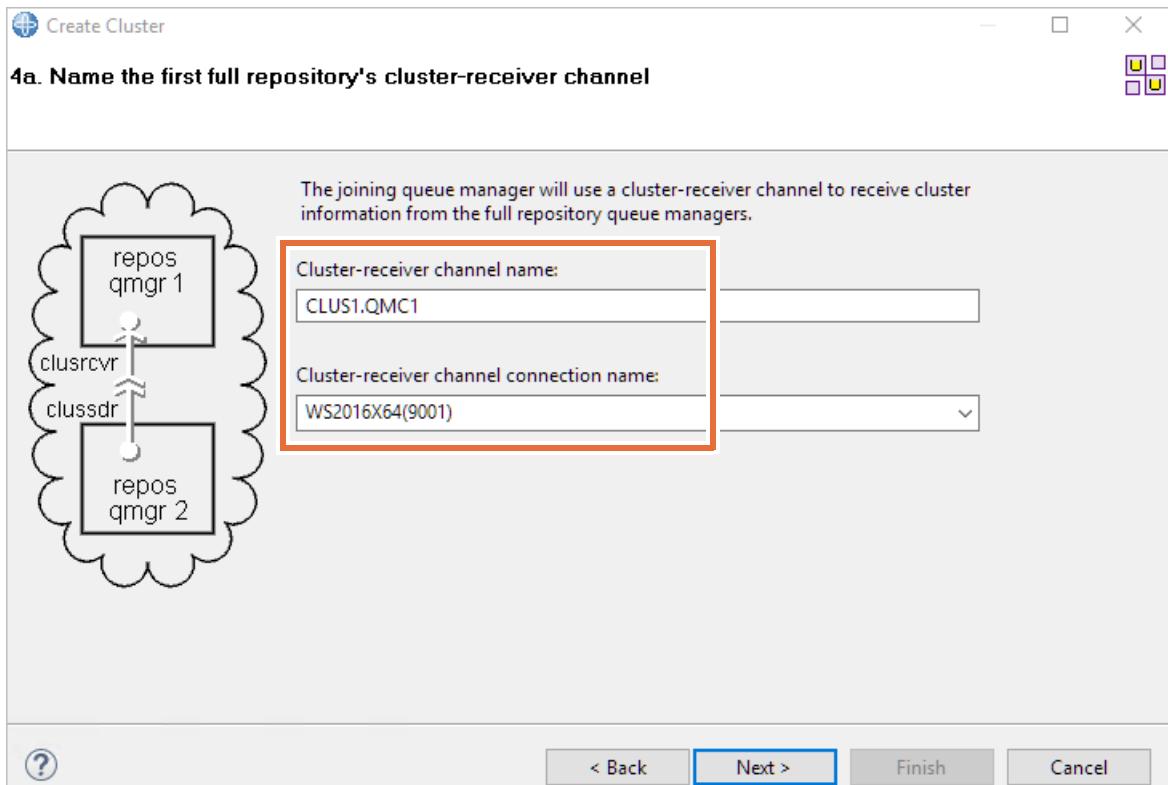
#### Hint

To create a queue manager, see [Section 2.1, "Create a queue manager"](#) in [Exercise 1, "Getting started with IBM MQ"](#).

### 1.2. Create the queue manager cluster

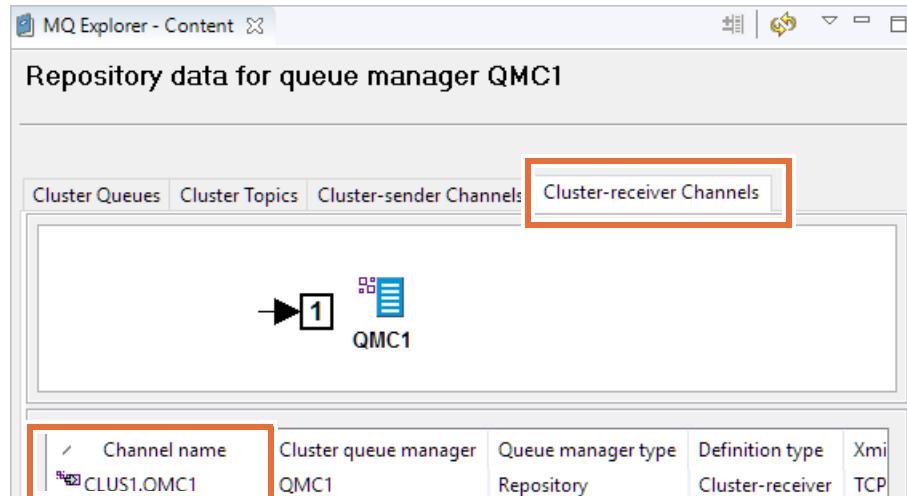
- \_\_\_ 1. In the **Navigator** pane, right-click **Queue Manager Clusters** and click **New > Queue manager cluster**.
- \_\_\_ 2. Create the cluster.
  - \_\_\_ a. For the cluster name, enter `CLUS1` and click **Next**.
  - \_\_\_ b. For the first full repository queue manager, select **QMC1** from the list and click **Next**.
  - \_\_\_ c. For the second full repository queue manager, select **QMC2** and click **Next**.
  - \_\_\_ d. Click **Next** to define the cluster channels.
  - \_\_\_ e. For the first full repository cluster-receiver channel, change **Cluster-receiver channel name** to: `CLUS1.QMC1`

The **Cluster-receiver channel connection name** field is automatically set to the correct name.

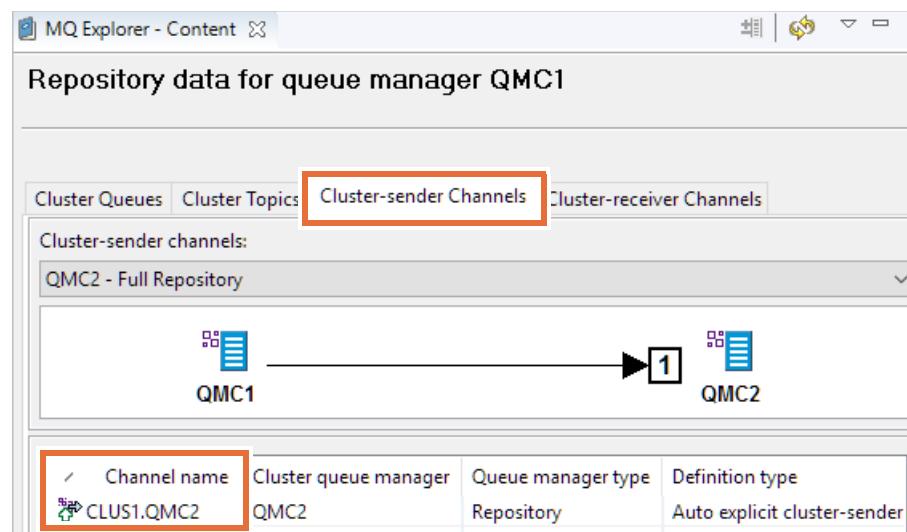


- \_\_\_ f. Click **Next**.
  - \_\_\_ g. For the second full repository cluster-receiver channel, change **Cluster-receiver channel name** to **CLUS1.QMC2** and click **Next**.
- The **Cluster-receiver channel connection name** field is automatically set to the correct name.
- \_\_\_ h. Click **Finish**.
3. In the Navigator pane, expand **Queue Managers Clusters > CLUS 1 > Full Repositories** to see that QMC1 and QMC2 are listed.

- \_\_\_ 4. In the **CLUS1 > Full Repositories** folder, select **QMC1** to open the **Repository data for queue manager QMC1** view
- \_\_\_ 5. Click the **Cluster-receiver Channels** tab and verify that QMC1 has one cluster-receiver channel that is named CLUS1.QMC1.



- \_\_\_ 6. Click the **Cluster-sender Channels** tab and verify that QMC1 has one cluster-sender channel that is named CLUS1.QMC2.



### 1.3. Add queue managers to the cluster

- \_\_\_ 1. Add QMC3 to the cluster as a partial repository.
  - \_\_\_ a. In the **Navigator** pane, right-click **CLUS1** and click **Add Queue Manager to Cluster**.
  - \_\_\_ b. Select **QMC3** from the list and click **Next**.
  - \_\_\_ c. Make sure that **Partial repository** is selected and click **Next**.
  - \_\_\_ d. Change **Cluster-receiver channel name** to **CLUS1.QMC3** and click **Next**.
  - \_\_\_ e. Make sure that **QMC1** is selected as the full repository queue manager and click **Next**.
  - \_\_\_ f. Accept the default to use the cluster-receiver channel **CLUS1.QMC1** and click **Next**.

- g. Click **Finish**.



### Information

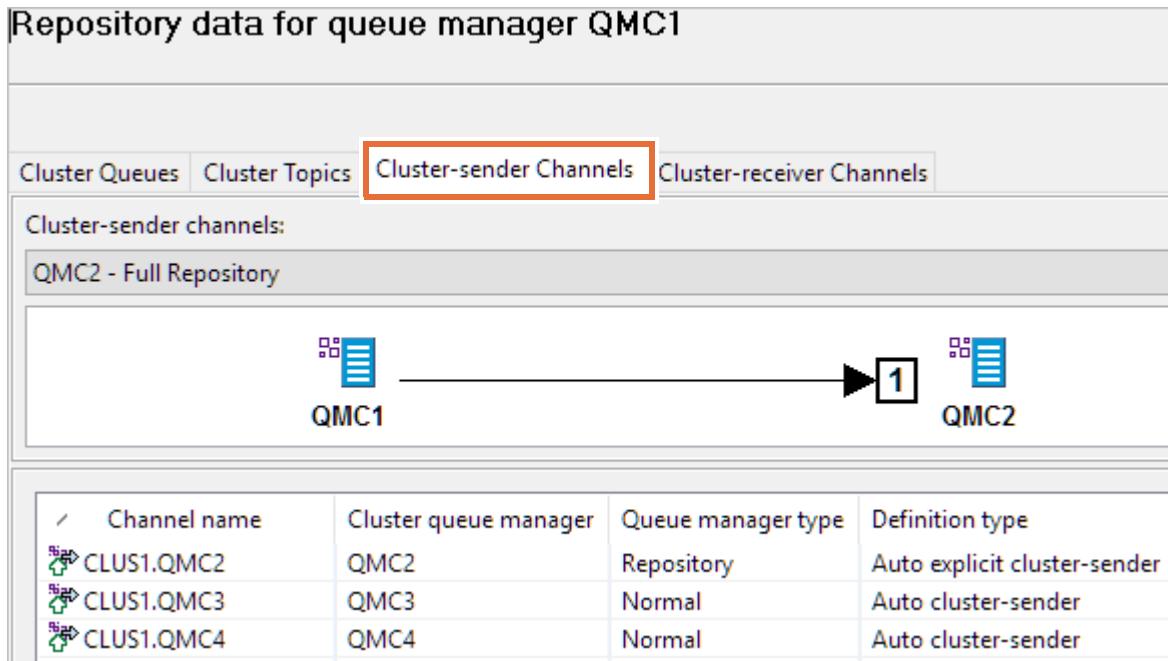
To add a partial repository to a cluster with MQSC, you define a cluster-receiver channel for QMC3, and a cluster-sender channel that points to a full repository queue manager.

The equivalent MQSC commands to add QMC3 to the cluster as a partial repository are:

```
DEFINE CHANNEL(CLUS1.QMC3) CHLTYPE(CLUSRCVR) TRPTYPE(TCP) +
CONNAME('localhost(9003)') CLUSTER(CLUS1)
```

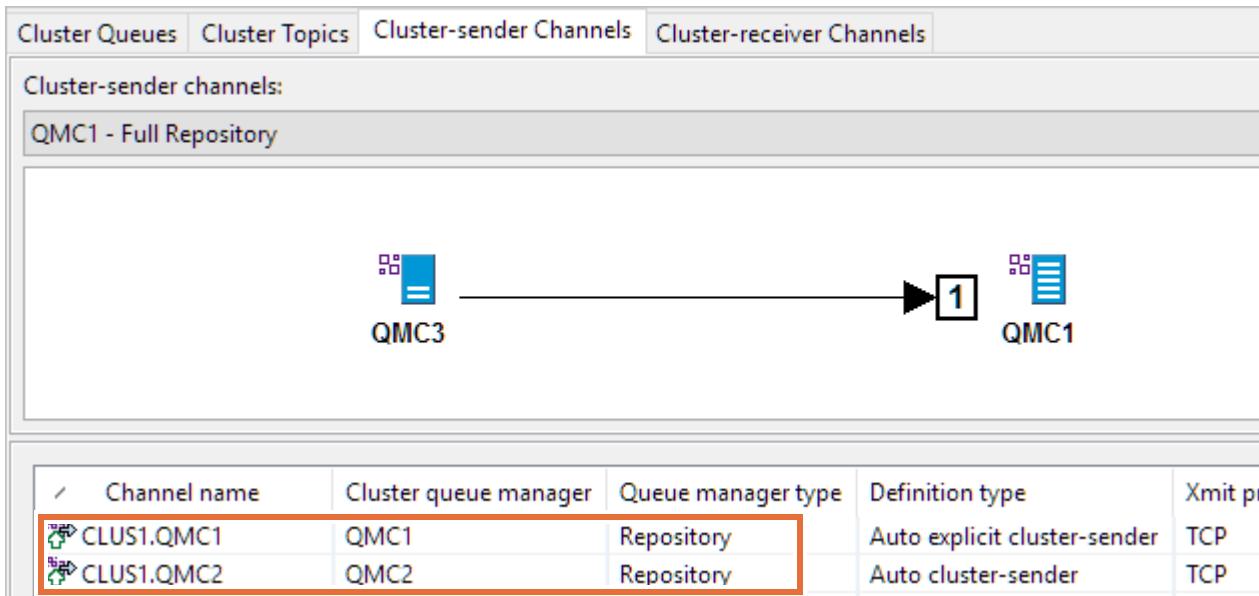
```
DEFINE CHANNEL(CLUS1.QMC1) CHLTYPE(CLUSSDR) TRPTYPE(TCP) +
CONNAME('localhost(9001)') CLUSTER(CLUS1)
```

- 2. Repeat [Step 1](#) to add QMC4 to the cluster as a partial repository with a cluster-receiver channel name of CLUS1.QMC4.
- 3. In the **Navigator** pane, click **CLUS1 > Full Repositories > QMC1** to verify that cluster-sender channels were automatically defined between the partial repositories (QMC3 and QMC4) and the full repository (QMC1).



- 4. In the **Navigator** pane, click **QMC3** under the **Partial Repositories** folder and verify that QMC3 has cluster-sender channels to both full repositories (QMC1 and QMC2).

Notice that the channel to the second full repository (QMC2) was automatically defined.



## 1.4. Create cluster queues

- 1. Create a cluster queue that is named **CLUSQ1** on QMC1.
  - a. In the **Navigator** pane, expand **Queue Managers > QMC1**.
  - b. Right-click **Queues** and click **New > Local Queue**.
  - c. Set **Name** to **CLUSQ1** and click **Next**.
  - d. Click **Cluster** in the left menu pane to open the cluster properties.
  - e. Select **Shared in cluster** and enter **CLUS1** for the cluster name.
  - f. Click **Finish**.
  - g. Click **OK** to close the success message.



### Information

The MQSC command for creating a cluster queue is: **DEF QL(CLUSQ1) CLUSTER(CLUUS1)**.

- 
- 2. Repeat [Step 1](#) to create the following queues:
    - On QMC2, create a cluster queue on CLUS1 that is named **CLUSQ1**.
    - On QMC3, create a cluster queue on CLUS1 that is named **CLUSQ1**.
    - On QMC4, create a cluster queue on CLUS1 that is named **CLUSQ2**.
  - 3. In the **Navigator** pane, click **CLUS1 > Full Repositories > QMC1** and click the **Cluster Queues** tab to verify that you have a cluster queue on each queue manager.

You should have a cluster queue that is named `CLUSQ1` on QMC1, QMC2, and QM3 and a cluster queue that is named `CLUSQ2` on QMC4.

**Repository data for queue manager QMC1**

Cluster Queues Cluster Topics Cluster-sender Channels Cluster-receiver Channels

QMCA QMCB QMC1 QMC2

| Queue name | Queue type | Description | Put messages | Default priority | Default persistence |
|------------|------------|-------------|--------------|------------------|---------------------|
| CLUSQ1     | Cluster    |             | Allowed      | 0                | Not persistent      |
| CLUSQ1     | Cluster    |             | Allowed      | 0                | Not persistent      |
| CLUSQ1     | Cluster    |             | Allowed      | 0                | Not persistent      |
| CLUSQ2     | Cluster    |             | Allowed      | 0                | Not persistent      |

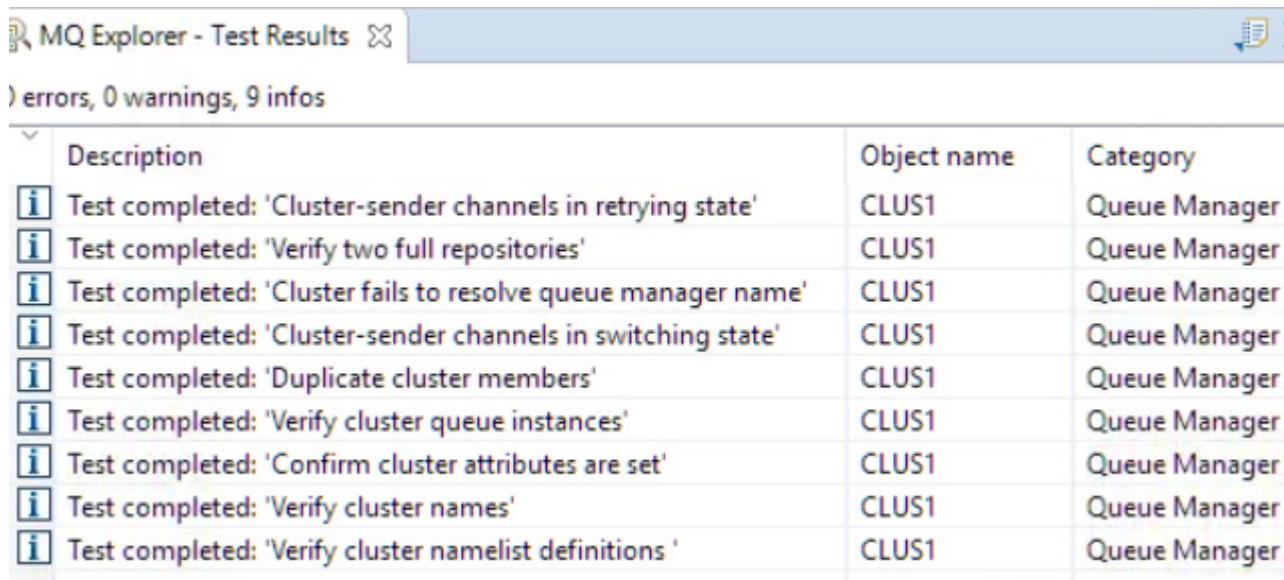
## Section 2. Managing the cluster workload

### 2.1. Testing the cluster

In this scenario, CLUSQ1 is defined locally on three queue managers: QMC1, QMC2, and QMC3.

In the most basic scenario that uses default configuration, all messages that are put to the cluster queue CLUSQ1 are put on the local manager.

- \_\_\_ 1. In the **Navigator** pane, right-click **CLUS1** and click **Tests > Run Default Tests**.
- \_\_\_ 2. Click **OK** to close the confirmation message.
- \_\_\_ 3. Review the test results in the **Test Results** view.



| Description                                                       | Object name | Category      |
|-------------------------------------------------------------------|-------------|---------------|
| [i] Test completed: 'Cluster-sender channels in retrying state'   | CLUS1       | Queue Manager |
| [i] Test completed: 'Verify two full repositories'                | CLUS1       | Queue Manager |
| [i] Test completed: 'Cluster fails to resolve queue manager name' | CLUS1       | Queue Manager |
| [i] Test completed: 'Cluster-sender channels in switching state'  | CLUS1       | Queue Manager |
| [i] Test completed: 'Duplicate cluster members'                   | CLUS1       | Queue Manager |
| [i] Test completed: 'Verify cluster queue instances'              | CLUS1       | Queue Manager |
| [i] Test completed: 'Confirm cluster attributes are set'          | CLUS1       | Queue Manager |
| [i] Test completed: 'Verify cluster names'                        | CLUS1       | Queue Manager |
| [i] Test completed: 'Verify cluster namelist definitions '        | CLUS1       | Queue Manager |

The default tests verify the cluster object definitions and verify that no errors or potential problems exist.

### 2.2. Using a round-robin approach for workload balancing

- \_\_\_ 1. Use the sample program `amqspput` and the `data.txt` lab file to write 15 messages to the cluster queue CLUSQ1.
  - \_\_\_ a. Open a command prompt and switch to the `C:\labfiles\Lab05-cluster\data` directory.  
`cd C:\labfiles\Lab05-cluster\data`
  - \_\_\_ b. Run the `amqspput` program.  
`amqspput CLUSQ1 QMC1 < C:\labfiles\Lab05-cluster\data\data.txt`
- \_\_\_ 2. Return to MQ Explorer and click the **Queues** folder in each of the **QMC1**, **QMC2**, and **QMC3** queue manager folders to view the queue depth of CLUSQ1 queues.



## Questions

How many messages were put to each queue?

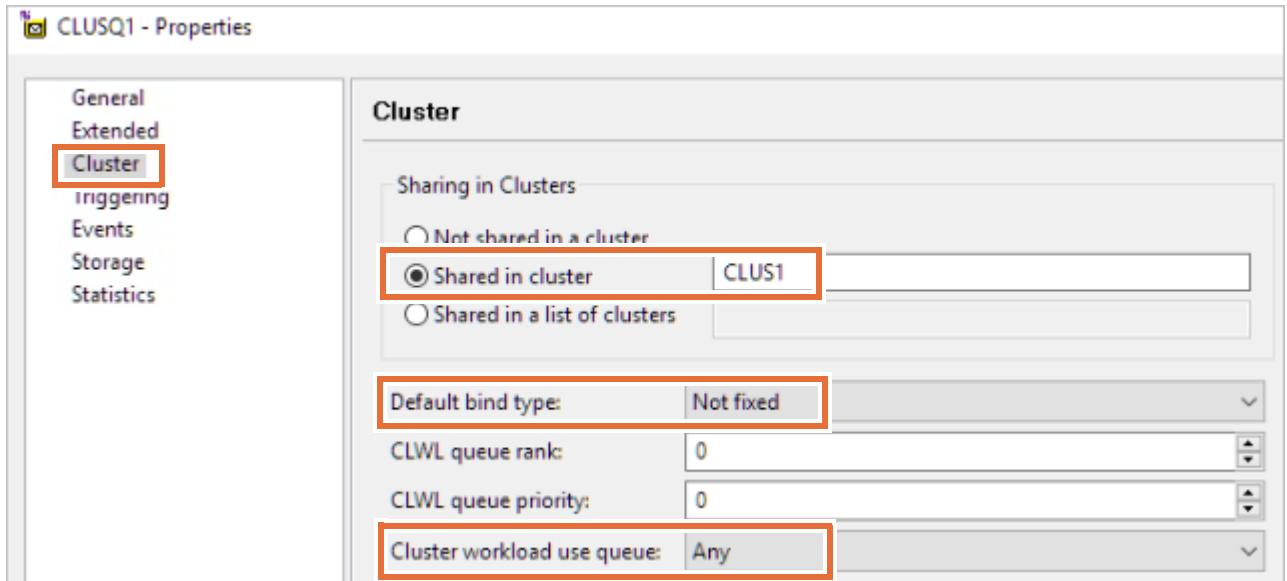
You should see that all 15 messages were put to the queue manager QMC1.

The default queue definition has the **Default bind type** (DEFBIND) set to **Open** and the **Cluster workload use queue** (CLWLUSEQ) set to **Queue manager**. These settings explain why all the messages were placed onto one queue manager.

- DEFBIND(OPEN) binds the queue handle to a specific instance of the cluster queue when the queue is opened.
- CLWLUSEQ(QMGR) means that the CLWLUSEQ attribute of the queue manager definition specifies the behavior. By default, it is LOCAL and the target of an MQPUT is the local queue instance, if one exists.

- 
- \_\_\_ 3. Clear the messages from CLUSQ1 on queue manager QMC1.
    - \_\_\_ a. In the **Navigator** pane, click the **QMC1 > Queues**, and in the **Queue** view, right-click **CLUSQ1**, and then click **Clear Messages**.
    - \_\_\_ b. Select **Queue will be cleared using MQGET API calls** and click **Clear**.
    - \_\_\_ c. Click **OK** to close the success message.
    - \_\_\_ d. Verify that the queue depth is 0.
  - \_\_\_ 4. Change the queue definition for CLUSQ1 queue on the QMC1, QMC2, and QMC3 queue managers.
    - \_\_\_ a. In the **QMC1 > Queues** folder, in the **Queue** view, right-click **CLUSQ1** and click **Properties**.
    - \_\_\_ b. On the **Cluster** page, change **Default bind type** to **Not fixed** so that the queue handle is not bound to any one instance of the cluster queue.

- \_\_\_ c. Change **Cluster workload use queue** to **Any** and click **OK**.



- \_\_\_ d. Repeat these steps for QMC2 and QMC3.

### Information

The MQSC command for changing the cluster queue properties is:

```
ALTER QL(CLUSQ1) DEFBIND(NOTFIXED) CLQLUSEQ(ANY)
```

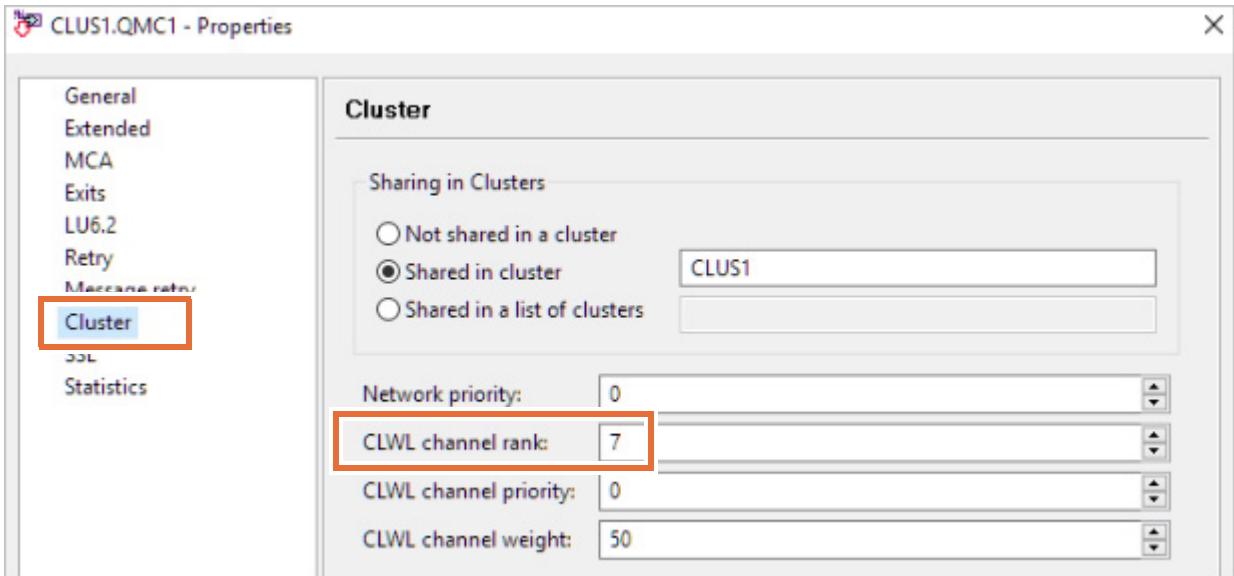
- \_\_\_ 5. Repeat [Step 1](#) to put messages to the cluster queue by running the `amqspput` sample program.  
`amqspput CLUSQ1 QMC1 < C:\labfiles\Lab05-cluster\data\data.txt`
- \_\_\_ 6. View the queue depth of CLUSQ1 on QMC1, QMC2, and QMC3.  
 You should see that the messages are distributed evenly between all three instances of the CLUSQ1 queue on the three queue managers.

## 2.3. Controlling workload with channel and queue rank

In this part of the exercise, you alter the queue and channel definitions to specify a cluster workload rank attribute. The rank attribute directs messages to only two of the three queue managers in the cluster.

- \_\_\_ 1. On **QMC1**, change the **cluster-receiver** channel definition to have a cluster workload channel rank of 7.
  - \_\_\_ a. In the **Navigator** pane, click the **QMC1 > Channels** folder to see the **Channels** content view.
  - \_\_\_ b. In the Channels view, right-click **CLUS1.QMC1** (the *cluster-receiver* channel) and click **Properties**.

- \_\_ c. On the **Cluster** page, set the **CLWL channel rank** property to 7 and click **OK**.

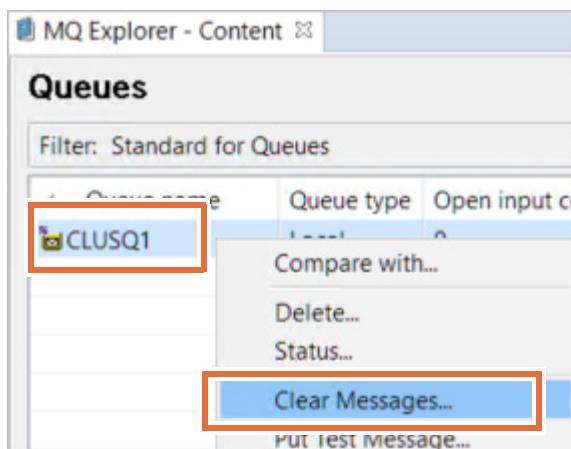


## Information

The MQSC command to change the cluster-receiver channel definition on QMC1 is:

```
ALTER CHANNEL(CLUS1.QMC1) CHLTYPE(CLUSRCVR) CLWLRank(7)
```

- \_\_ 2. On **QMC3**, repeat [Step 1](#) to change the **CLUS1.QMC3** cluster-receiver channel definition to set the cluster workload channel rank to 7.
- \_\_ 3. On each queue manager, clear the messages off the CLUSQ1 queue by using the **Queue will be cleared using MQGET API calls** option.
  - \_\_ a. On QMC1, click the **QMC1 > Queues** folder, right-click **CLUSQ1**, and click **Clear Messages**.



- \_\_ b. When prompted, click **Queue will be cleared using MQGET API calls**.
- \_\_ c. Repeat these steps for QMC2 and QMC3.

**Note:** Clearing the queues is to simplify the math for the number of messages to expect on each queue. This step is not required for the exercise to work.



## Information

If the **Open output count** of the CLUSQ1 on any of the queue managers is greater than 0, an application still has the queue open for output. To clear **Open output count**, right-click the queue in the **Queue** contents view, click **Clear Messages**, and click **Queue will be cleared using MQGET API calls**. This option closes the queue after clearing the messages. You might need to refresh to see the change.

- 4. Run the `amqspput` program on QMC1 with `data.txt` to put 15 messages to the cluster queue CLUSQ1.

```
amqspput CLUSQ1 QMC1 < C:\labfiles\Lab05-cluster\data.txt
```

You should see that the messages are split across QMC1 and QMC3, and QMC2 has 0 messages.

- 5. On **QMC2**, repeat [Step 1](#) to change the *cluster-receiver* channel definition for **CLUS1.QMC2** to set the cluster workload channel rank to 9.
  - 6. Put more messages by running the `amqspput` program again.
- ```
amqspput CLUSQ1 QMC1 < C:\labfiles\Lab05-cluster\data.txt
```
- All 15 messages should now be directed to QMC2.
- 7. For the *cluster-receiver* channel **CLUS1.QMC2** on **QMC2**, set the cluster workload channel rank back to 0.
 - 8. Change the cluster workload **queue rank** (CLWLRANK) property for the queue CLUSQ1 on QMC3 to 6.
- a. Click **QMC3 > Queues**, and in the **Queue** content view, right-click **CLUSQ1** and click **Properties**.
 - b. Click **Cluster** to display the **Cluster** properties.
 - c. Change **CLWL queue rank** to 6.
 - d. Click **OK**.



Information

The MQSC command to alter the queue definition of CLUSQ1 on QMC3 is:

```
ALTER QL(CLUSQ1) CLWLRANK(6)
```

- 9. Use `amqspput` again to put messages to the cluster queue CLUSQ1 and notice that all messages go to QMC3 this time.

```
amqspput CLUSQ1 QMC1 < C:\labfiles\Lab05-cluster\data.txt
```

2.4. Controlling workload with channel priority

In this part of the exercise, you use the channel priority property to control the distribution of messages on the instances of the cluster queue.

For this test, you set the following cluster workload properties on these *cluster-receiver* channels:

- ___ 1. Clear the messages from the cluster queues: CLUSQ1 on QMC1, QMC2, and QMC3.
 - ___ a. For QMC1, click the **QMC1 > Queues** folder, right-click **CLUSQ1**, and click **Clear Messages**.
 - ___ b. When prompted, click **Queue will be cleared using MQGET API calls**.
 - ___ c. Repeat these steps for QMC2 and QMC3.
- ___ 2. Reset the cluster workload channel rank (CLWLRANK) for QMC1 and QMC3
 - On QMC1: **CLUS1.QMC1**, set CLWL channel **rank** to: 0
 - On QMC3: **CLUS1.QMC3**, set CLWL channel **rank** to: 0



Hint

To set cluster workload properties, click the **Channels** folder under the queue manager, and in the **Channels** content view, right-click the *cluster-receiver* channel and click **Properties**.

-
- ___ 3. Set the *cluster-receiver* channel **priority** (CLWPRTY) for QMC2 and QMC3:
 - On QMC2: **CLUS1.QMC2**, set CLWL channel **priority** to: 3
 - On QMC3: **CLUS1.QMC3**, set CLWL channel **priority** to: 1
 - ___ 4. Put messages by using the **amqspput** program.

```
amqspput CLUSQ1 QMC1 < C:\labfiles\Lab05-cluster\data.txt
```

 Why are all the messages put on QMC3's CLUSQ1 queue, even though QMC2 has a higher channel priority? Because the CLUSQ1 on QMC3 still has the CLWL *queue rank* set to 6. The cluster workload algorithm uses *channel* and *queue rank* over *priority*.
 - ___ 5. On the QMC3 CLUSQ1 **queue**, set the CLWL **queue rank** (CLWLRANK) back to 0.
 - ___ a. Click **QMC3 > Queues**, and in the **Queue** content view, right-click **CLUSQ1** and click **Properties**.
 - ___ b. Click **Cluster**, change **CLWL queue rank** to 0, and click **OK**.
 - ___ 6. Put messages by running the **amqspput** program.

```
amqspput CLUSQ1 QMC1 < C:\labfiles\Lab05-cluster\data.txt
```

 Now, all 15 messages are directed to QMC2, which is the highest priority queue manager.
 - ___ 7. On QMC2, stop the **CLUS1.QMC2** cluster-receiver channel.
 You might need to wait a few moments for it to stop.
 - ___ 8. Put messages by running the **amqspput** program.

```
amqspput CLUSQ1 QMC1 < C:\labfiles\Lab05-cluster\data.txt
```

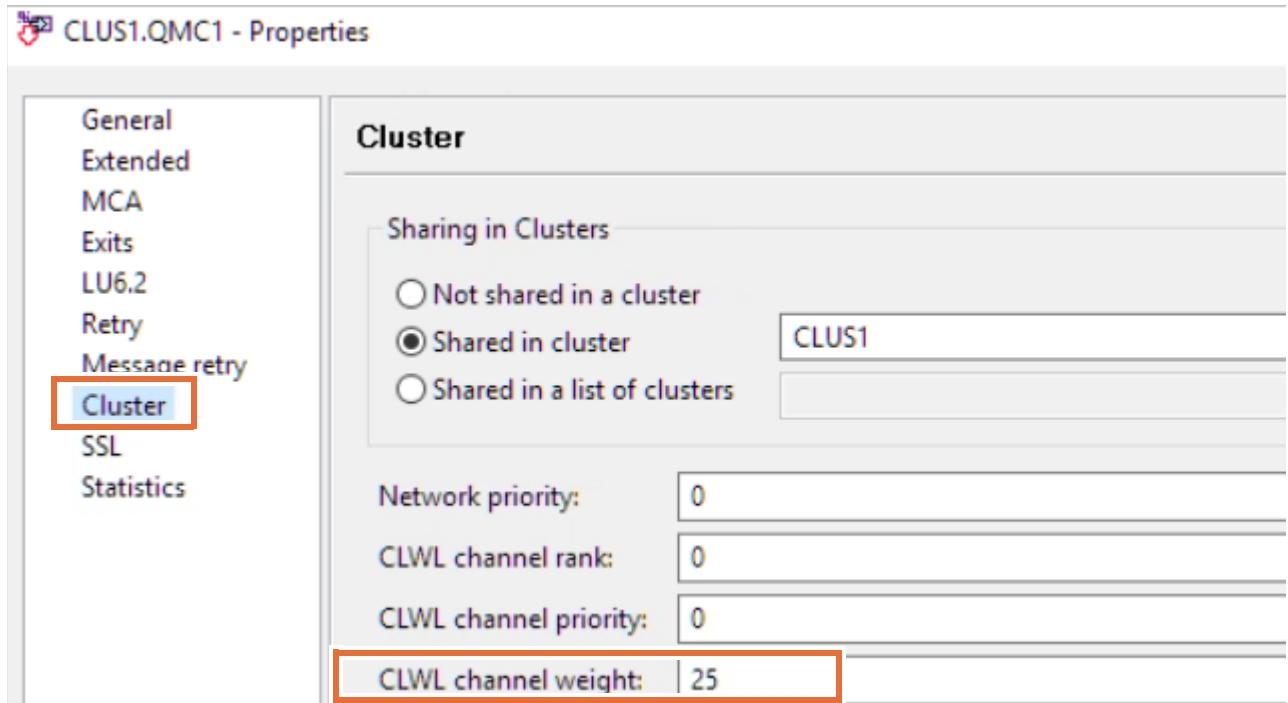
You should see that all the messages go to QMC3, which is the highest priority queue manager.

- ___ 9. On QMC2, restart the **CLUS1.QMC2** cluster-receiver channel.

2.5. Using channel weight to control workload

Suppose now that QMC3 has greater processing power than any of the other queue managers in this cluster. Channel weighting can be used to direct workload to the most powerful queue manager.

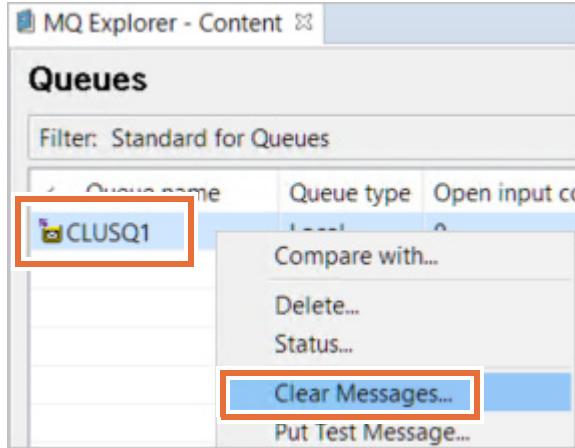
- ___ 1. Reset the cluster workload cluster priority (CLWLPRTY) of the QMC2 and QMC3 cluster-receiver channels to 0.
 - For **CLUS1.QMC2** on QMC2, set **CLWL channel priority** to: 0
 - For **CLUS1.QMC3** on QMC3, set **CLWL channel priority** to: 0
- ___ 2. Assuming that QMC3 has twice the processing power of QMC1 and QMC2, set the following channel weights on the cluster-receiver channels:
 - **CLUS1.QMC1** on **QMC1**: Set **CLWL channel weight** to: 25
 - **CLUS1.QMC2** on **QMC2**: Set **CLWL channel weight** to: 25
 - ___ a. In the **Navigator** pane, click the **QMC1 > Channels** folder.
 - ___ b. Right-click **CLUS1.QMC1** (the *cluster-receiver channel*) and click **Properties**.
 - ___ c. On the **Cluster** page, set the **CLWL channel weight** property to: 25



- ___ d. Click **OK**.
- ___ e. Repeat these steps for **CLUS1.QMC2** on QMC2.

Note: QMC3 keeps the default **CLWL channel weight**: 50

- ___ 3. Clear all messages from the CLUSQ1 on QMC1, QMC2, and QMC3.
 - ___ a. For QMC1, click the **QMC1 > Queues** folder, right-click **CLUSQ1**, and click **Clear Messages**.



- ___ b. When prompted, click **Clear**.
- ___ c. Repeat these steps for QMC2 and QMC3.
- ___ 4. Put new messages on the queues.


```
amqsput CLUSQ1 QMC1 < C:\labfiles\Lab05-cluster\data.txt
```

You should see that 50% of the messages went to QMC3, 25% of the messages went to QMC1 and QMC2 each.
- ___ 5. Reset the following channel weights on the cluster-receiver channels:
 - QMC1: Set **CLWL channel weight** to: 50
 - QMC2: Set **CLWL channel weight** to: 50

2.6. Exercise cleanup

- ___ 1. Clear all messages from the CLUSQ1 on QMC1, QMC2, and QMC3.
- ___ 2. Keep the cluster queue managers running.
- ___ 3. Close any command prompts.

End of exercise

Exercise review and wrap-up

The first part of the exercise demonstrated how to create a cluster of queue managers. You then test the cluster by sending messages between queues on all queue managers in the cluster.

Exercise 6. Configuring publish/subscribe message queuing

Estimated time

01:00

Overview

In this exercise, you define and test an IBM MQ publish/subscribe network by using a direct cluster and a topic host cluster. You also use the IBM MQ sample programs and IBM MQ Explorer to test the cluster and the IBM MQ display route command to show the message route through the publish/subscribe cluster.

Objectives

After completing this exercise, you should be able to:

- Define a direct route publish/subscribe cluster
- Define a topic host route publish/subscribe cluster
- Test the publish/subscribe cluster
- Use the IBM MQ display route (dspmq rte) command to verify the route that the message takes through the publish/subscribe cluster

Introduction

In this exercise, you work with a publish/subscribe cluster. You send messages through the cluster and use the IBM MQ display route command (`dspmq rte`) to verify the route that the message takes through the cluster. You also modify the publish/subscribe cluster to use topic host routing.

This exercise includes these sections:

- [Section 1, "Setting up a publish/subscribe cluster"](#)
- [Section 2, "Testing the publish/subscribe cluster"](#)
- [Section 3, "Testing cluster publication routing"](#)
- [Section 4, "Clustered publish/subscribe with topic host routing"](#)

Requirements

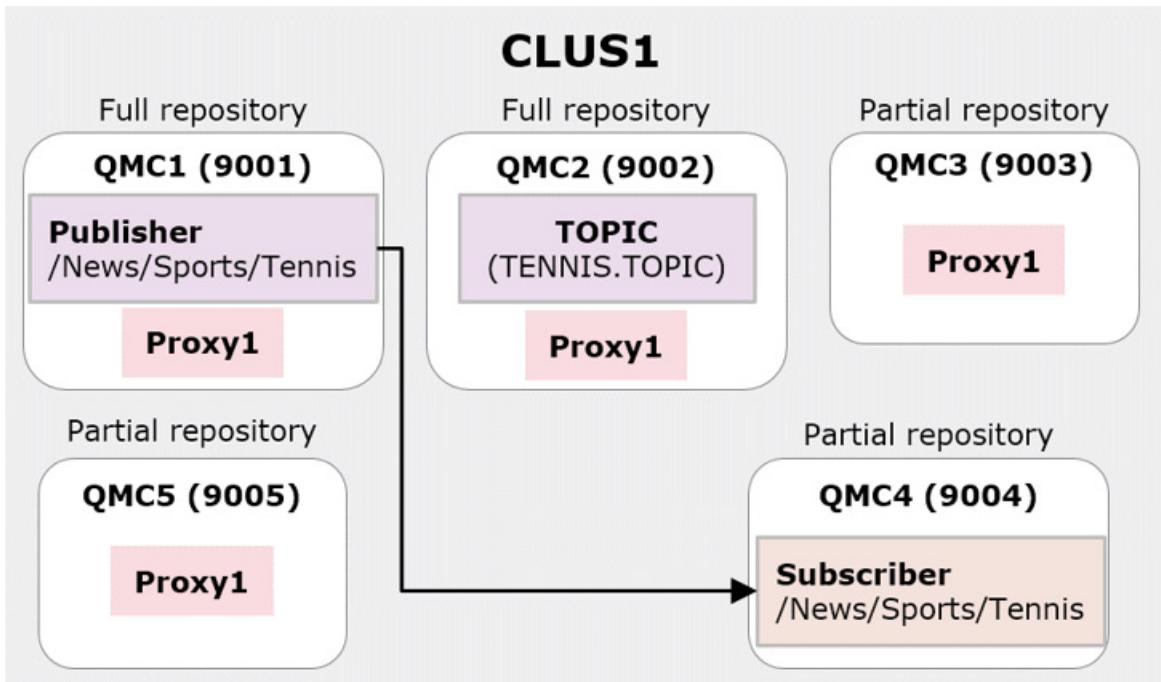
This exercise uses the cluster that you created in [Exercise 5, "Implementing a basic cluster"](#).

Section 1. Setting up a publish/subscribe cluster

Direct routing is the simplest way to enable a publish/subscribe network. In this configuration, all queue managers in the cluster become aware of all other queue managers in the cluster. A proxy subscription is sent to each queue manager so that any queue manager in the cluster that receives a publication can connect to a subscriber's queue manager.

In this part of the exercise, you use a cluster with five queue managers (QMC1 to QMC5). An administered topic object that is named TENNIS.TOPIC, which has CLROUTE(DIRECT), is declared on the QMC2 queue manager. When the subscriber application runs, it attaches to the QMC4 queue manager through the topic string /News/Sports/Tennis. Proxy subscriptions are then propagated to every member of the cluster.

A publication that is published on QMC1 is routed directly to the subscriber on QMC4 by using the cluster channels between QMC1 and QMC4. The arrows in the figure show the path that the message follows when the publish/subscribe cluster uses direct routing.



1.1. Add a partial repository to the cluster

1. In IBM MQ Explorer, make sure that the cluster queue managers that you created in [Exercise 5, "Implementing a basic cluster"](#) are running: QMC1, QMC2, QMC3, and QMC4. The cluster that you created in [Exercise 5, "Implementing a basic cluster"](#) has two partial repository queue managers (QMC3 and QMC4). For this exercise, you need another partial repository queue manager.

- ___ 2. In MQ Explorer, create a queue manager with these values.
 - Name: **QMC5**
 - Dead-letter queue: **DLQ**
 - Port: **9005**
- ___ 3. Add QMC5 to the cluster as a partial repository.
 - ___ a. In the **Navigator** pane, right-click **Queue Manager Clusters > CLUS1** and click **Add Queue Manager to Cluster**.
 - ___ b. Select **QMC5** from the list and click **Next**.
 - ___ c. Keep Partial repository selected and click **Next**.
 - ___ d. Change **Cluster-receiver channel name** to **CLUS1.QMC5** and click **Next**.

Note that the **Cluster-receiver channel connection name** field is automatically set to the correct name.
 - ___ e. Keep **QMC1** selected as the full repository queue manager and click **Next**.
 - ___ f. Keep **CLUS1.QMC1** as the cluster-receiver channel and click **Next**.
 - ___ g. Click **Finish**.
- ___ 4. Run the MQSC DISPLAY CLUSQMGR command with the QMTYPE option on QMC1 to display the CLUS1 cluster and verify the components of the cluster.
 - ___ a. Open a command prompt and start MQSC for QMC1.
`runmqsc QMC1`
 - ___ b. Display the cluster and components.
`DISPLAY CLUSQMGR(*) QMTYPE`

The command should return results similar to the following example:

AMQ8441I: Display Cluster Queue Manager details.	
CLUSQMGR(QMC1)	CHANNEL(CLUS1.QMC1)
CLUSTER(CLUS1)	QMTYPE(REPOS)
AMQ8441I: Display Cluster Queue Manager details.	
CLUSQMGR(QMC2)	CHANNEL(CLUS1.QMC2)
CLUSTER(CLUS1)	QMTYPE(REPOS)
AMQ8441I: Display Cluster Queue Manager details.	
CLUSQMGR(QMC3)	CHANNEL(CLUS1.QMC3)
CLUSTER(CLUS1)	QMTYPE(NORMAL)
AMQ8441I: Display Cluster Queue Manager details.	
CLUSQMGR(QMC4)	CHANNEL(CLUS1.QMC4)
CLUSTER(CLUS1)	QMTYPE(NORMAL)
AMQ8441I: Display Cluster Queue Manager details.	
CLUSQMGR(QMC5)	CHANNEL(CLUS1.QMC5)
CLUSTER(CLUS1)	QMTYPE(NORMAL)
 - ___ c. Close MQSC mode by typing: `end`

1.2. Create a topic

- 1. Create a topic on QMC2 that is called TENNIS.TOPIC with a topic string of **/News/Sports/Tennis**.

- a. In a command prompt, start MQSC for QMC2:

```
runmqsc QMC2
```

- b. Define the topic.

```
DEFINE TOPIC(TENNIS.TOPIC) TOPICSTR('/News/Sports/Tennis') CLUSTER(CLUS1)
```

- c. Verify the topic definition on QMC2 by displaying the topic status.

```
DISPLAY TCLUSTER(TENNIS.TOPIC) TOPICSTR CLUSTER CLROUTE CLSTATE
```

The command should return results similar to the following example:

AMQ8633I: Display topic details.

TOPIC(TENNIS.TOPIC)	TYPE(CLUSTER)
TOPICSTR(/News/Sports/Tennis)	CLUSTER(CLUS1)
CLROUTE(DIRECT)	CLSTATE(ACTIVE)

- d. Verify the topic tree definition on QMC2.

```
DISPLAY TPSTATUS('/#') TYPE(TOPIC) CLUSTER CLROUTE SUBCOUNT ADMIN
end
```

— e. Note the following values in the command results:

- TOPICSTR is the topic string for each tree node.
- ADMIN contains the name of the administrative topic object.
- CLUSTER is the name of the cluster (CLUS1) and appears on the clustered topic only.
- CLROUTE is set to NONE except for the TENNIS.TOPIC where it is set to DIRECT, which indicates direct routing.
- SUBCOUNT is the number of subscriptions currently aware of the topic.

```

AMQ8690I: IBM MQ topic created.
DISPLAY TCLUSTER(TENNIS.TOPIC) TOPICSTR CLUSTER CLRROUTE CLSTATE
      2 : DISPLAY TCLUSTER(TENNIS.TOPIC) TOPICSTR CLUSTER CLRROUTE CLSTATE
AMQ8633I: Display topic details.
          TOPIC(TENNIS.TOPIC)                      TYPE(CLUSTER)
          TOPICSTR(/News/Sports/Tennis)            CLUSTER(CLUS1)
          CLRROUTE(DIRECT)                         CLSTATE(ACTIVE)
DISPLAY TPSTATUS('/#') TYPE(TOPIC) CLUSTER CLRROUTE SUBCOUNT ADMIN
      3 : DISPLAY TPSTATUS('/#') TYPE(TOPIC) CLUSTER CLRROUTE SUBCOUNT ADMIN
AMQ8754I: Display topic status details.
          TOPICSTR()                           ADMIN(SYSTEM.BASE.TOPIC)
          CLUSTER()                            CLRROUTE(NONE)
          SUBCOUNT(0)

AMQ8754I: Display topic status details.
          TOPICSTR(/News)                     ADMIN( )
          CLUSTER()                          CLRROUTE(NONE)
          SUBCOUNT(0)

AMQ8754I: Display topic status details.
          TOPICSTR(/News/Sports/Tennis)        ADMIN(TENNIS.TOPIC)
          CLUSTER(CLUS1)                   CLRROUTE(DIRECT)
          SUBCOUNT(0)

AMQ8754I: Display topic status details.
          TOPICSTR(/News/Sports)           ADMIN( )
          CLUSTER()                        CLRROUTE(NONE)
          SUBCOUNT(0)

```

The topic knowledge is spread to all cluster members. The publish/subscribe engine can match its publications to subscriptions in the same queue manager by using the topic tree.

— 2. Use the following MQSC commands to repeat [Step 1](#) for each of the other queue managers: QMC3, QMC4, QMC5.

```

runmqsc QMC3 | runmqsc QMC4 | runmqsc QMC5

DISPLAY TCLUSTER(TENNIS.TOPIC) TOPICSTR CLUSTER CLRROUTE CLSTATE
DISPLAY TPSTATUS('/#') TYPE(TOPIC) CLUSTER CLRROUTE SUBCOUNT ADMIN
end

```

— 3. Close the command prompt.

Section 2. Testing the publish/subscribe cluster

- ___ 1. Open three command windows side by side.

You use the first command prompt as the ***publisher***. The other 2 command prompts as the ***subscribers***. However, the subscription has a timeout of 30 seconds before it automatically disconnects. So before the subscription starts running, you need to prepare the message that you want to publish.

- ___ 2. In the first command prompt, set up the *publisher* on QMC1 by entering the following command:

```
amqspub /News/Sports/Tennis QMC1
```

- ___ 3. In the *publisher* command prompt, type the following message, but *DO NOT* press Enter yet.

Tennis starts today

- ___ 4. In the other command prompts, set up the *subscribers* on QMC3 and QMC4 by entering the following commands.

```
amqssub /News/Sports/Tennis QMC3
```

```
amqssub /News/Sports/Tennis QMC4
```

- ___ 5. In the first command prompt (*publisher*), press Enter to publish the message.

The message is displayed in both the subscribing command prompt windows.

```
Administrator: Command Prompt - amqspub /News/Sports/Tennis QMC1
Microsoft Windows [Version 10.0.14393]
(c) 2016 Microsoft Corporation. All rights reserved.

C:\Users\Administrator>amqspub /News/Sports/Tennis QMC1
Sample AMQSPUBA start
target topic is /News/Sports/Tennis
Tennis starts today
```

The image shows two separate command prompt windows. Both windows have a title bar 'Administrator: Command Prompt'. The left window's title bar has a red border. The right window's title bar is standard. Both windows show identical command-line output. The output includes the Windows version (10.0.14393), copyright notice, command (amqspub), target topic (/News/Sports/Tennis), and the published message ('Tennis starts today'). The message is highlighted with a red box in both windows. The right window's title bar is standard, while the left window's title bar has a red border.

```
Administrator: Command Prompt
Microsoft Windows [Version 10.0.14393]
(c) 2016 Microsoft Corporation. All rights reserved.

C:\Users\Administrator>amqssub /News/Sports/Tennis QMC3
Sample AMQSSUBA start
Calling MQGET : 30 seconds wait time
message <Tennis starts today>
Calling MQGET : 30 seconds wait time
no more messages
Sample AMQSSUBA end

C:\Users\Administrator>

Administrator: Command Prompt
Microsoft Windows [Version 10
(c) 2016 Microsoft Corporatio

C:\Users\Administrator>amqssu
Sample AMQSSUBA start
Calling MQGET : 30 seconds wa
message <Tennis starts today>
Calling MQGET : 30 seconds wa
no more messages
Sample AMQSSUBA end

C:\Users\Administrator>
```

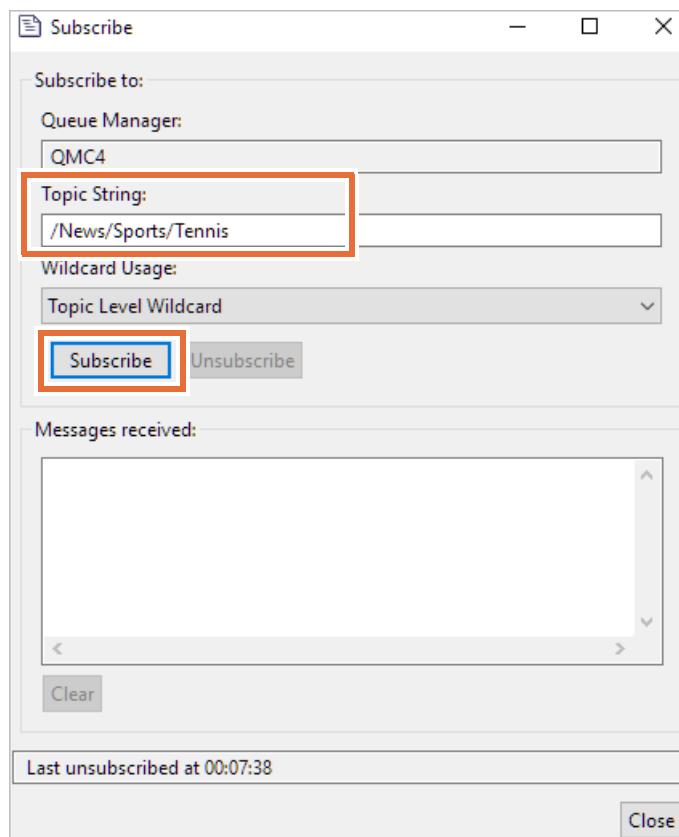
6. Keep the publisher command prompt running, but close the subscriber command prompts.

Section 3. Testing cluster publication routing

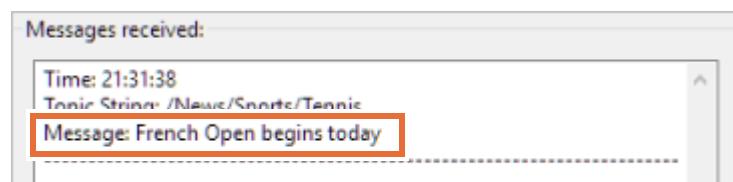
In this part of the exercise, you use IBM MQ Explorer to attach subscribers. You use the IBM MQ display route application (`dspmqrte`) to display the route of the message through the cluster.

3.1. Create a test subscription

- ___ 1. Create a test subscription on QMC4 in MQ Explorer.
 - ___ a. In the Navigator pane, expand **Queue Managers** > **QMC4**, right-click **Topics** and click **Test Subscription**.
 - ___ b. In the **Topic String** field, type: /News/Sports/Tennis
 - ___ c. Click **Subscribe**. Do not click **Close**.



- ___ 2. Return to the publisher command prompt and enter a message, such as:
French Open begins today
- ___ 3. Return to the Subscribe window in MQ Explorer to see the message. Do not close the Subscribe window. You use it for the next steps.



- ___ 4. Repeat [Step 1](#) to create a test subscription on QMC5 and do not close the Subscribe window.

3.2. Validate the proxy subscriptions

- ___ 1. In the **pane**, in the **Queue Managers > QMC5** folder, right-click **Topics** and click **Status**.
- ___ 2. In the **Topic string** column, expand the topic tree from the **[Empty]** node to Tennis, and scroll to the right to see the **Sub count** column.

The proxy subscription count is provided in the **Sub count** column.

QMCS - Topic Status

Queue Manager: QMC5

Topic status:

Topic string	Type	Admin topic name	Sub count	Pub count	Retained publication	NPM de
▼ [Empty]		SYSTEM.BASE.TOPIC	0	0	No	To all av
▼ News			0	0	No	To all av
▼ Sports			0	0	No	To all av
Tennis		TENNIS.TOPIC	2	0	No	To all av
SSYS			0	0	No	To all av

- ___ 3. Right-click the **Tennis** node line and then select **Topic Status - Subscribers** to display more information about the proxy subscription such as the subscription ID.

/News/Sports/Tennis - Status

Queue Manager: QMC5

Topic status - subscribers for the topic "/News/Sports/Tennis":

Filter: [Not Available]

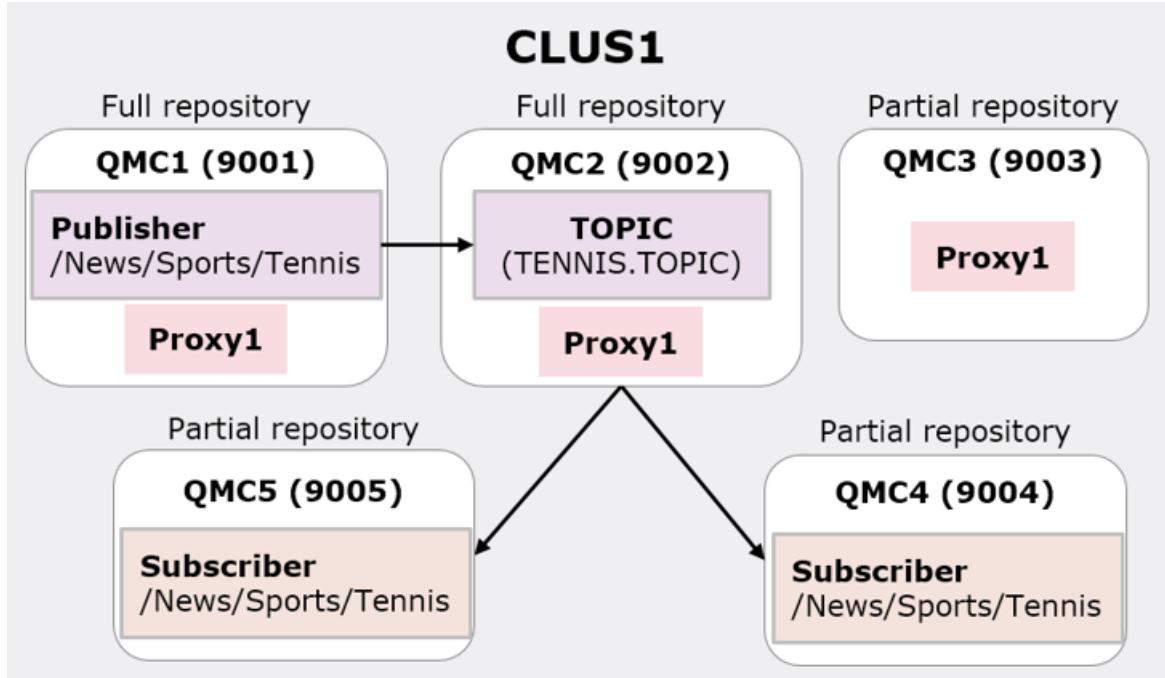
Topic string	Subscription ID	User	Durable	Type	Connection ID
/News/Sports/Tennis	414D5120514D43352020...	Administrato	No	API	414D5143514D
/News/Sports/Tennis	414D5120514D43352020...	MUSR_MQADMIN	Yes	Proxy	0000000000000000

- ___ 4. Click **Close**.
- ___ 5. Close the Subscribe windows for QMC4 and QMC5.

Section 4. Clustered publish/subscribe with topic host routing

In this part of the exercise, you change the topic on QMC2 from direct routing to topic-host routing.

When you modify the topic to use topic-host routing, messages are routed to subscribers through QMC2, which is the topic host queue manager.



4.1. Test topic-host routing

- ___ 1. Replace the existing DIRECT routed topic on QMC2 with a TOPICHOST routed topic.
 - ___ a. In a command prompt, run MQSC to remove the existing TENNIS.TOPIC topic that has routing set to DIRECT from QMC2.


```
runmqsc QMC2
ALTER TOPIC(TENNIS.TOPIC) CLUSTER('')
```



Important

The topic object must be removed from the cluster by clearing the CLUSTER attribute before the CLROUTE attribute is changed in the next step. If you do not remove the topic object first, you receive a system error that warns you to do so.

-
- ___ b. Add the TENNIS.TOPIC to QMC2 (the topic host) with routing set to TOPICHOST by entering:


```
ALTER TOPIC(TENNIS.TOPIC) CLRROUTE(TOPICHOST) CLUSTER(CLUS1)
```
 - ___ c. Verify the topic definition by displaying the topic status:


```
DISPLAY TCLUSTER(TENNIS.TOPIC) TOPICSTR CLUSTER CLRROUTE CLSTATE
```

The command should return the following status:

```
AMQ8633I: Display topic details.
    TOPIC(TENNIS.TOPIC)                      TYPE(CLUSTER)
    TOPICSTR(/News/Sports/Tennis)             CLUSTER(CLUS1)
    CLRROUTE(TOPICHOST)                   CLSTATE(ACTIVE)
```

- ___ 2. Verify that the topic definition is on the other queue managers (QMC1, QMC3, QMC4, and QMC5) by displaying the topic status.

In MQSC for each queue manager, type:

```
DISPLAY TCLUSTER(TENNIS.TOPIC) TOPICSTR CLUSTER CLRROUTE CLSTATE
```

The topic status should match the status returned by QMC2 in [Step 1](#).

- ___ 3. Test the topic host routing by creating a test subscription on QMC4 in MQ Explorer.
 - ___ a. In the **Navigator** pane, expand the queue manager QMC4 under the **Queue Managers** folder.
 - ___ b. Right-click **Topics**, and then click **Test Subscription**.
 - ___ c. In the **Topic String** field, type: /News/Sports/Tennis
 - ___ d. Click **Subscribe**. Do not click **Close**.
- ___ 4. Return to the publisher command prompt to publish a message on QMC1.
 - ___ a. If you closed the command prompt window that was running the **amqspub** sample program, open a new command prompt window and enter:


```
amqspub /News/Sports/Tennis QMC1
```
 - ___ b. Enter a message.
- ___ 5. In another command prompt, display the route to verify that the message went through the topic host.


```
dspmqrte -ts /News/Sports/Tennis -ac -d yes -v outline activity -w 3 -m QMC1
```

 The command should return results that show the message is now routed through QMC2 (the topic host).
- ___ 6. Repeat [Step 3](#) to add a subscription to the QMC5 queue manager.
- ___ 7. Use MQSC to validate the cluster proxy subscriptions on QMC2.

```
runmqsc QMC2
DISPLAY SUB(*) SUBTYPE(PROXY)
```

You should see two proxy subscriptions.

```
AMQ8096I: IBM MQ subscription inquired.
SUBID(414D5120514D433220202020202020D9D4825D225F5C2E)
SUB(SYSTEM.PROXY_2.TENNIS.TOPIC CLUS1 QMC5 /News/Sports/Tennis)
SUBTYPE(PROXY)
```

4.2. Exercise cleanup

- ___ 1. Stop all the queue managers.

__ 2. Close all command prompts.

End of exercise

Exercise review and wrap-up

The first part of the exercise demonstrated how to define and test an IBM MQ publish/subscribe network by using a direct cluster and a topic host cluster.

Exercise 7. Controlling access to IBM MQ

Estimated time

01:00

Overview

In this exercise, you use the IBM MQ OAM commands to set access control on a queue, and then use the IBM MQ sample programs to see the effect of attempting to breach security.

Objectives

After completing this exercise, you should be able to:

- Define and display access control on a queue
- Manage authority records
- Enable and monitor authority events
- Test security

Introduction

This exercise includes these sections:

- [Section 1, "Point-to-point security"](#)
- [Section 2, "Publish and subscribe security"](#)

Requirements

This exercise requires that the Windows local user and group repository include:

- **oamlabuser**
- **mquser**, member of the **mqusers** group

Section 1. Point-to-point security

By default, a new IBM MQ queue manager uses the server's operating system for authentication. It uses the IBM MQ Object Authority Manager (OAM) to enforce authorization. You can configure a queue manager to use alternative options, such as an LDAP server.

For this exercise, you use the Windows local user and group repository.

1.1. Set up the lab environment

- ___ 1. In a command prompt, create and start a new queue manager named OAMQM.

```
crtmqm -p 1420 -u SYSTEM.DEAD.LETTER.QUEUE OAMQM
strmqm OAMQM
```

- ___ 2. Start MQSC mode for OAMQM.

```
runmqsc OAMQM
```

- ___ 3. Create two local queues on OAMQM.

```
DEFINE QLOCAL(Q1) REPLACE
DEFINE QLOCAL(Q2) REPLACE
```

- ___ 4. Run the following commands to create topic nodes in the topic tree.

```
DEFINE TOPIC(SPORT) TOPICSTR('sport') REPLACE
DEFINE TOPIC(SPORT.FOOTBALL) TOPICSTR('sport/football') REPLACE
DEFINE TOPIC(SPORT.FOOTBALL.PLAYERS) TOPICSTR('sport/football/players')
REPLACE
DEFINE TOPIC(SPORT.FOOTBALL.PLAYERS.HURSLEY)
TOPICSTR('sport/football/players/hursley') REPLACE
DEFINE TOPIC(SPORT.FOOTBALL.PLAYERS.WINCHESTER)
TOPICSTR('sport/football/players/winchester') REPLACE
```

- ___ 5. Exit MQSC mode by typing: end

1.2. Test user permissions

In these steps, you compare permissions for an administrative user and an unauthorized user (oamlabuser).

- ___ 1. In the command prompt (which by default, runs under the administrator role), type in the following command:

```
amqspput Q1 OAMQM
```

- ___ 2. Type in some text and then press the Enter twice to send the message and exit the amqspput program.

Your message is stored on the Q1 queue.

- ___ 3. Start a command prompt as the unauthorized user, oamlabuser.

- ___ a. Open a new command prompt and enter the following command:

```
runas /user:oamlabuser cmd
```

- ___ b. When prompted for a password, enter: passw0rd
- ___ 4. In the unauthorized user (`oamlabuser`) command window, run the `amqspput` sample program to attempt to put a message to queue QL.A on the queue manager QM01. Type:


```
amqspput Q1 OAMQM
```

 You should receive an error message with a reason code 2035 (MQRC_NOT_AUTHORIZED).
- ___ 5. Start a command prompt as the unauthorized user, `mquser`.
 - ___ a. Open a new command prompt and enter the following command:


```
runas /user:mquser cmd
```
 - ___ b. When prompted for a password, enter: passw0rd
 You can keep the command prompts open and return to them during the exercise.



Information

The `oamlabuser` user was not granted explicit authorization to connect to the OAMQM queue manager. Authorization to connect to an individual queue manager is required before a user can access any of the objects that are managed by the queue manager.

The `oamlabuser` also requires authorization to write messages to the Q1 queue.

1.3. Grant authorization to the queue manager

Next, you grant authorization to connect to the **OAMQM** queue manager and grant authorization to access the Q1 queue for PUT operations.

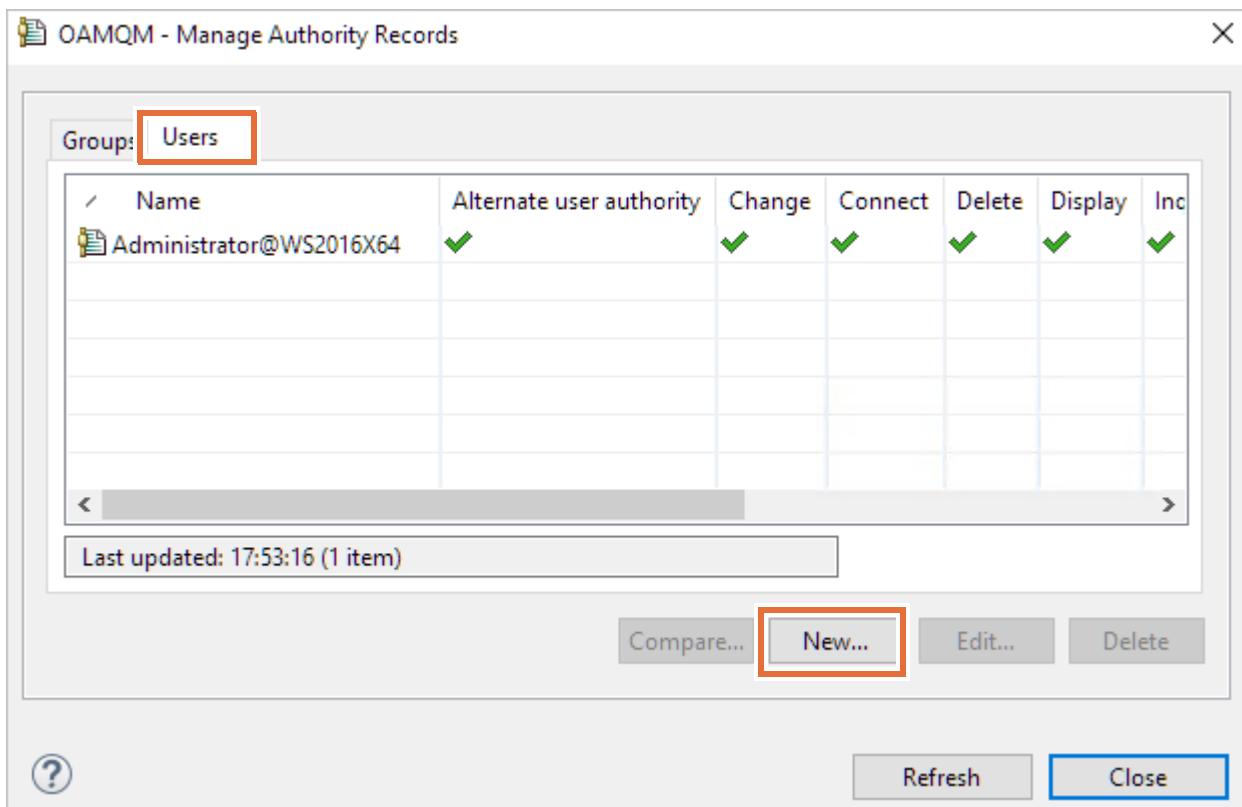


Important

Configuration of security is specific to a queue manager. It is possible to have different security configurations set up for each queue manager when there are multiple queue managers configured on the same server.

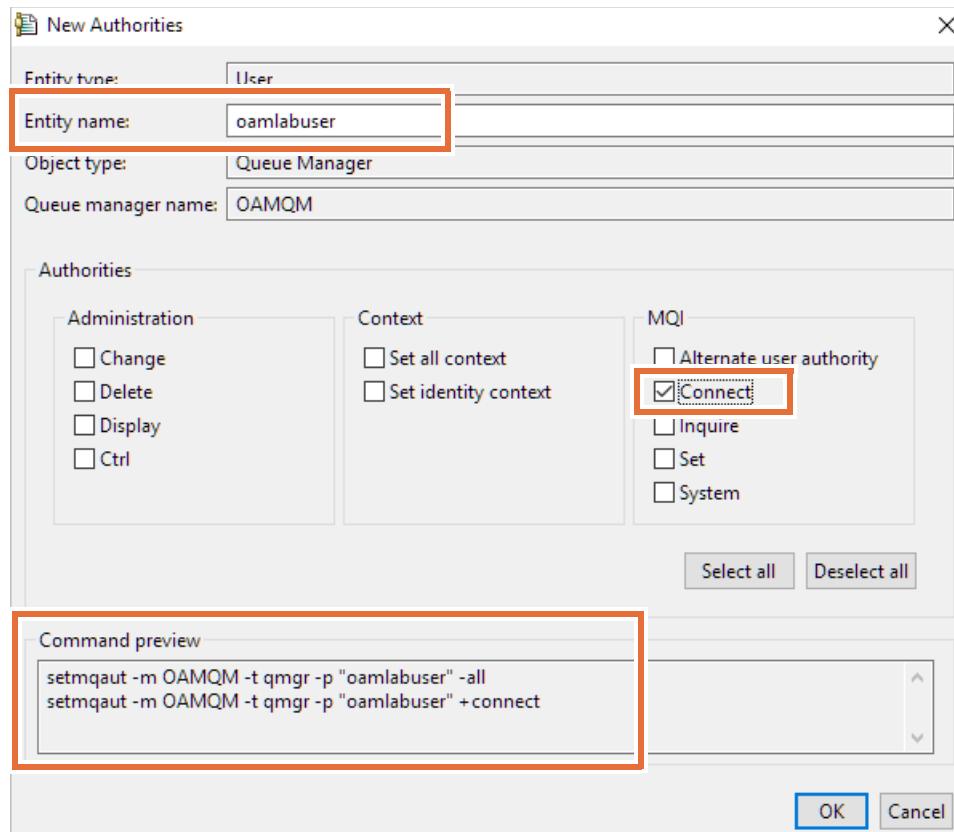
- ___ 1. In MQ Explorer, in the **Navigator** pane, right-click **OAMQM** and click **Object Authorities > Manage Queue Manager Authority Records**.

2. Click the **Users** tab to see which users are authorized to access the queue manager, and click **New**.



3. In the **Entity name** field, type oamlabuser

- 4. In the **MQI** section, select **Connect** and notice the **Command preview** section, which shows the equivalent command line commands for this action.



Information

You can set a number of authorization options by using the OAM. For more information, see the `setmqaut` command description in the IBM Knowledge Center.

- 5. Click **OK**.
- 6. Click **OK** to close the confirmation window.
- 7. Confirm that `oamlabuser` is correctly added to the list.

OAMQM - Manage Authority Records							
	Name	Alternate user authority	Change	Connect	Delete	Display	Inc
	Administrator@WS2016X64	✓	✓	✓	✓	✓	✓
	oamlabuser@WS2016X64			✓			

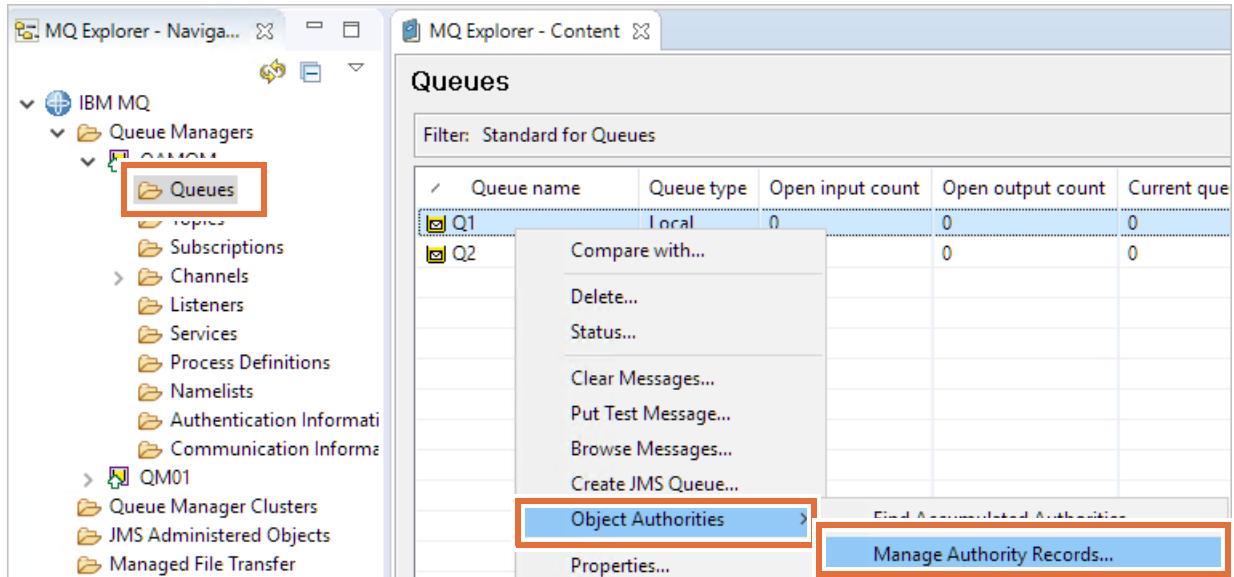
Next, you grant authorization for the `mqusers` group to connect to the **OAMQM** queue manager.

- ___ 8. Add authorization for the `mqusers` group.
 - ___ a. Click the **Groups** tab and click **New**.
 - ___ b. In the **Entity name** field, type `mqusers`
 - ___ c. In the **MQI** section, select **Connect**.
 - ___ d. Click **OK**.
 - ___ e. Click **OK** to close the confirmation window
 - ___ f. Verify that the `mqusers` group is correctly added to the list and click **Close**.

1.4. Grant authorization to the queue

Next, you grant authorization to the Q1 queue.

- ___ 1. In the Navigator pane in MQ Explorer, expand **OAMQM** and click the **Queues** folder.
- ___ 2. Right-click the **Q1** queue and select **Object Authorities > Manage Authority Records**.

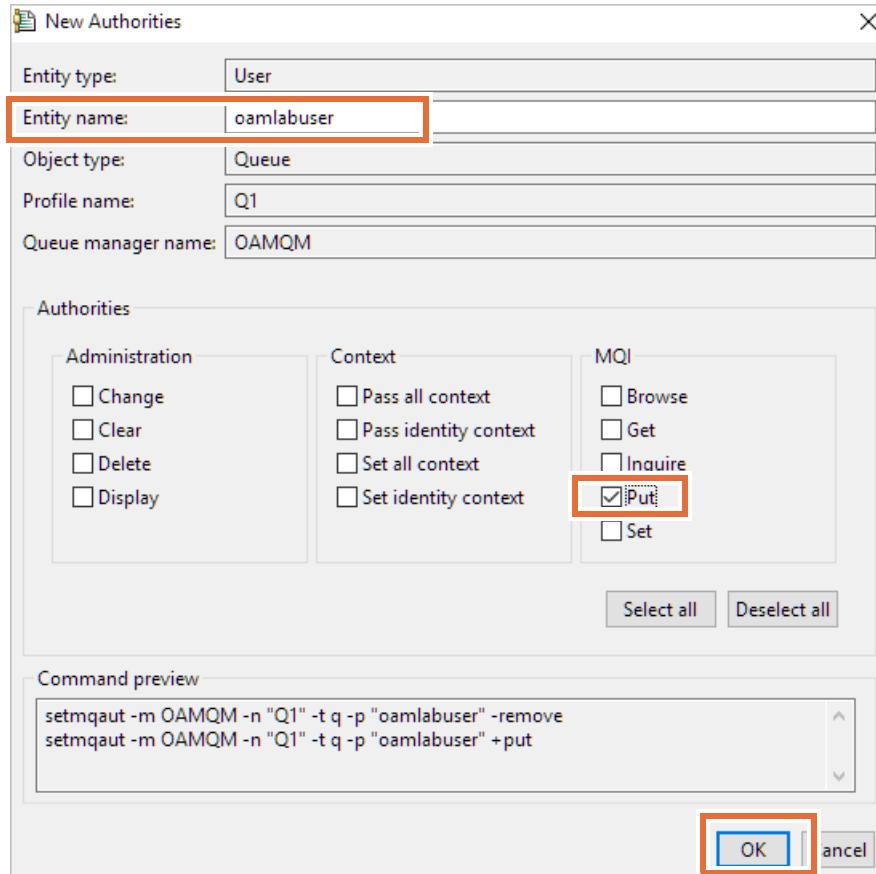


- ___ 3. Expand **Specific Profiles** and click **Q1**.

4. Select the **Users** tab and click **New...**.

The screenshot shows the 'OAMQM - Q1 - Manage Authority Records' application window. On the left, there's a navigation pane with 'Specific Profiles' expanded, showing 'Q1' selected. The main area has tabs for 'Groups' and 'Users'. The 'Users' tab is active, displaying a grid of authorities for the user 'Administrator@WS2016X64'. The grid columns are: Name, Browse, Change, Clear, Delete, Display, Get, Inquire, and Put. All columns have a green checkmark. Below the grid, a message says 'Last updated: 19:19:10 (1 item)'. At the bottom, there are buttons for 'Accumulated authorities...', 'Compare...', 'New...', 'Edit...', and 'Delete'. The 'New...' button is highlighted with a red box. The bottom right corner has 'Refresh' and 'Close' buttons.

- ___ 5. In the **Entity name** field, type `oamlabuser`, select **Put**, and click **OK**.



- ___ 6. Click **OK** to close the confirmation window.
- ___ 7. Verify that the `oamlabuser` record is correctly added with Put authority.

Name	Browse	Change	Clear	Delete	Display	Get	Inquire	Put
Administrator@WS2016X64	✓	✓	✓	✓	✓	✓	✓	✓
oamlabuser@WS2016X64								✓

- ___ 8. Add authorization for the `mqusers` group.
- Click the **Groups** tab and click **New**.
 - In the **Entity name** field, type `mqusers`.
 - In the **MQI** section, select **Put**.
 - Click **OK** and then click **OK** to close the confirmation window.
 - Verify that the `mqusers` group is correctly added to the list.
- ___ 9. Click **Close**.

1.5. Validate oamlabuser permissions

- ___ 1. In a command prompt, test the permissions for the unauthorized user, `oamlabuser`.
- Return to the command prompt running as `oamlabuser`.

- ___ b. If the command prompt that was running as oamlabuser is closed, open a new command prompt and enter the following command:

```
runas /user:oamlabuser cmd
```

- ___ c. When prompted for a password, enter: passw0rd
- ___ 2. In the unauthorized user (oamlabuser) command window, run the **amqspput** sample program to attempt to put a message to queue QL.A on the queue manager QM01.

```
amqspput Q1 OAMQM
```

This time, you are successfully prompted to put a message on the queue.

- ___ 3. Type a message, such as **My authorized message** and press Enter twice.
- ___ 4. In the same command prompt for oamlabuser, enter the following command to retrieve the messages from the queue:

```
amqsget Q1 OAMQM
```

The command fails because the user is authorized to *put* messages, but not to read or *browse* messages.

```
cmd (running as WS2016X64\oamlabuser)
Microsoft Windows [Version 10.0.14393]
(c) 2016 Microsoft Corporation. All rights reserved.

C:\Windows\system32>amqspput Q1 OAMQM
Sample AMQSPUT0 start
target queue is Q1
My authorized message

Sample AMQSPUT0 end

C:\Windows\system32>amqsget Q1 OAMQM
MQOPEN ended with reason code 2035
unable to open queue for input
Sample AMQSGET0 end

C:\Windows\system32>
```

Section 2. Publish and subscribe security

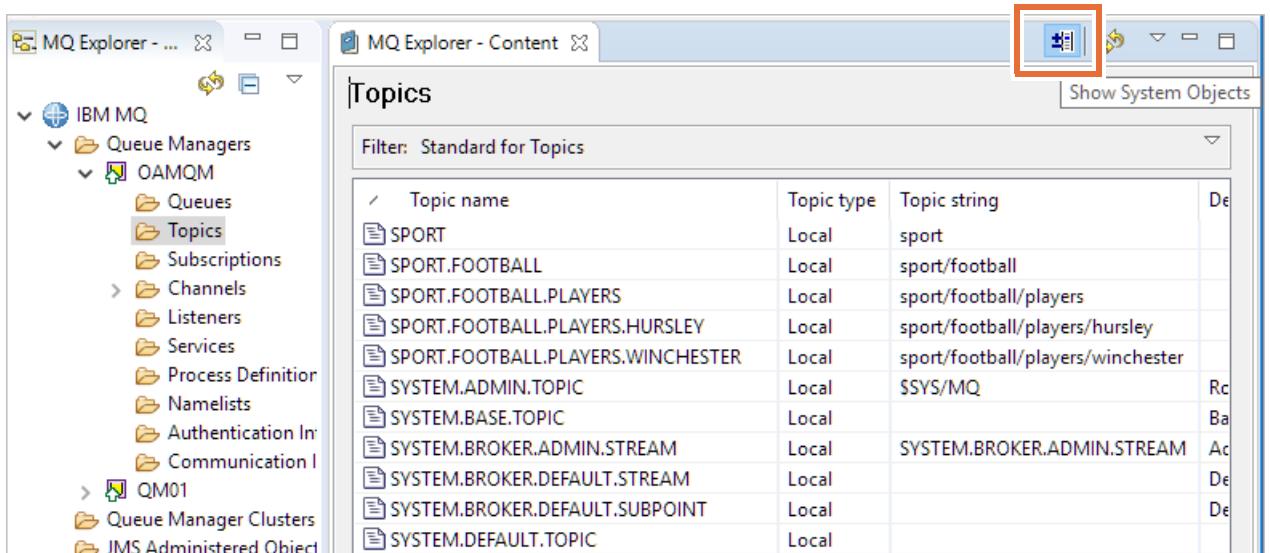
IBM MQ supports a topic-based publish and subscribe messaging system. One or more applications can act as publishers of messages on various topics. Likewise, one or more applications can act as subscribers of messages that are published to a topic. IBM MQ can support any number of publishers and subscribers on a topic.

IBM MQ uses OAM to authorize specific users and groups to access a topic. Security is enabled through topic objects.

In this section, you will explore how to use topic objects to manage authorization for publishing or subscribing to MQ topics.

2.1. Review of publish/subscribe security

- 1. In the **Navigator** pane of MQ Explorer, click the **Topics** folder of the **OAMQM** queue manager.
- 2. Click the **Show System Objects** icon.



The screenshot shows the IBM MQ Explorer interface. On the left, the Navigator pane displays a tree structure under 'IBM MQ' with 'OAMQM' expanded, showing 'Topics' as a child node. On the right, the Content pane is titled 'Topics' and lists a table of topics. The table has columns for 'Topic name', 'Topic type', 'Topic string', and 'Description'. The 'Topic name' column lists topics like 'SPORT', 'SPORT.FOOTBALL', 'SPORT.FOOTBALL.PLAYERS', etc. The 'Topic type' column shows most are 'Local'. The 'Topic string' column shows the full topic names. A red box highlights the 'Show System Objects' button in the toolbar above the table.

Topic name	Topic type	Topic string	Description
SPORT	Local	sport	
SPORT.FOOTBALL	Local	sport/football	
SPORT.FOOTBALL.PLAYERS	Local	sport/football/players	
SPORT.FOOTBALL.PLAYERS.HURSLEY	Local	sport/football/players/hursley	
SPORT.FOOTBALL.PLAYERS.WINCHESTER	Local	sport/football/players/winchester	
SYSTEM.ADMIN.TOPIC	Local	\$\$SYS/MQ	Rc
SYSTEM.BASE.TOPIC	Local		Ba
SYSTEM.BROKER.ADMIN.STREAM	Local	SYSTEM.BROKER.ADMIN.STREAM	Ac
SYSTEM.BROKER.DEFAULT.STREAM	Local		De
SYSTEM.BROKER.DEFAULT.SUBPOINT	Local		De
SYSTEM.DEFAULT.TOPIC	Local		

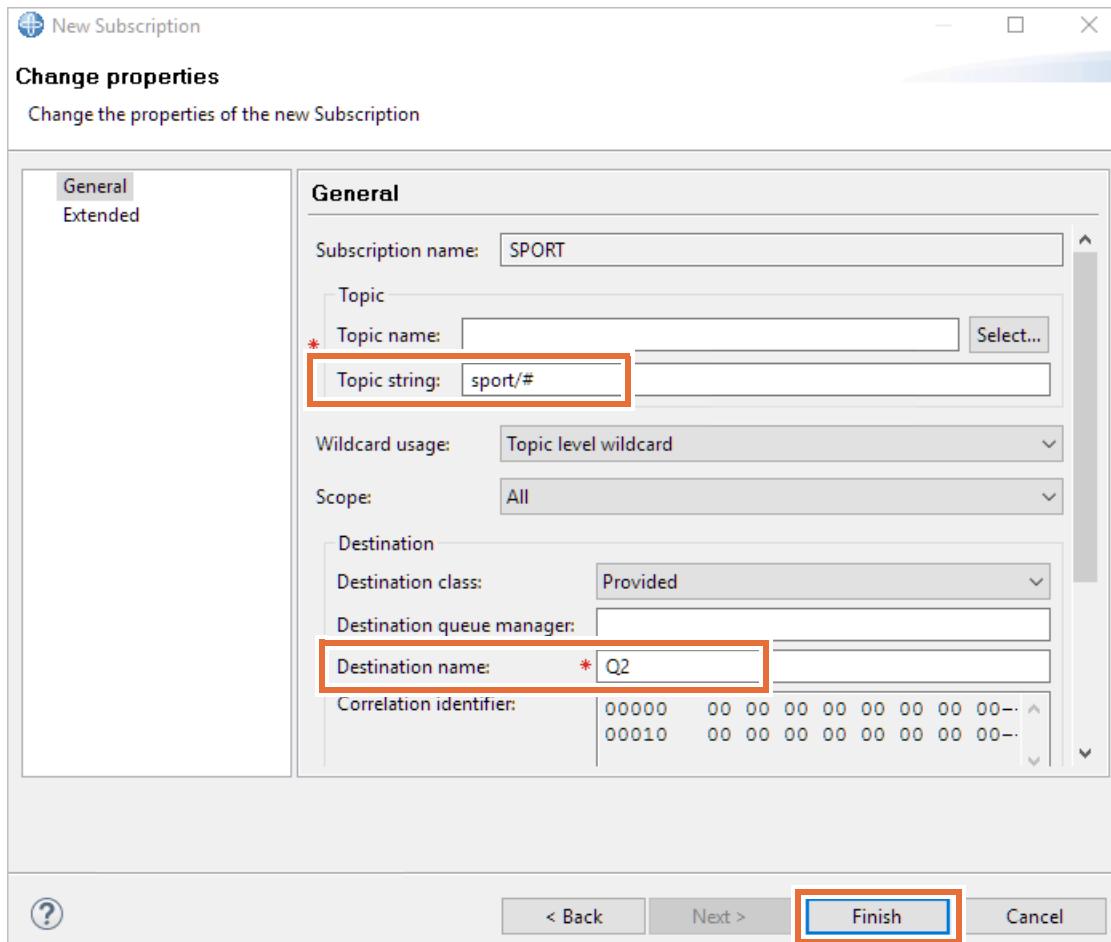
You see the list of topics that you created in [Section 1.1, "Set up the lab environment"](#), plus SYSTEM-provided topics.

The `SYSTEM.BASE.TOPIC` topic is the root level administrative topic for all user-defined topics in the queue manager. If an application publishes a message that is not matched by an administrative topic, then the default attributes, including authorization, are applied from this object.

- 3. Create an administered subscription to capture the messages.
 - a. In the **Navigator** pane, right-click the **OAMQM > Subscriptions** folder and click **New > Subscription**.
 - b. Set the **Name** to `SPORT` and click **Next**.

- __ c. On the General properties page, set the following properties, and then click **Finish**.

- **Topic string:** sport/#
- **Destination name:** Q2



Any messages that are successfully published to a topic that includes `sport` at the top of the topic hierarchy are sent to the Q2 queue.

2.2. Validate the permissions

- __ 1. Put a message on the SPORT topic as an administrator.

Note: Use a command prompt that is *not* running as oamlabuser and mquser.

```
amqspub sport OAMQM
```

- __ 2. Test publishing as **oamlabuser**.

- __ a. Return to the **oamlabuser** command prompt, or open a new one by typing the following command in a command prompt:

```
runas /user:oamlabuser cmd
```

- __ b. When prompted for a password, enter: `passw0rd`

- __ c. In the **oamlabuser** command window, try to put a message on the sport topic.

```
amqspub sport OAMQM
```

___ 3. Test publishing as **mquser**.

- ___ a. Open a new command prompt and enter the following command:

```
runas /user:mquser cmd
```

- ___ b. When prompted for a password, enter: passw0rd

- ___ c. In the **mquser** command window, try to put a message on the sport topic.

```
amqspub sport OAMQM
```

The **oamlabuser** and **mquser** users received an error with reason code 2035.



Questions

Is the error due to the lack of authorization to the OAMQM queue manager, or a lack of authorization to the sport topic?

Answer:

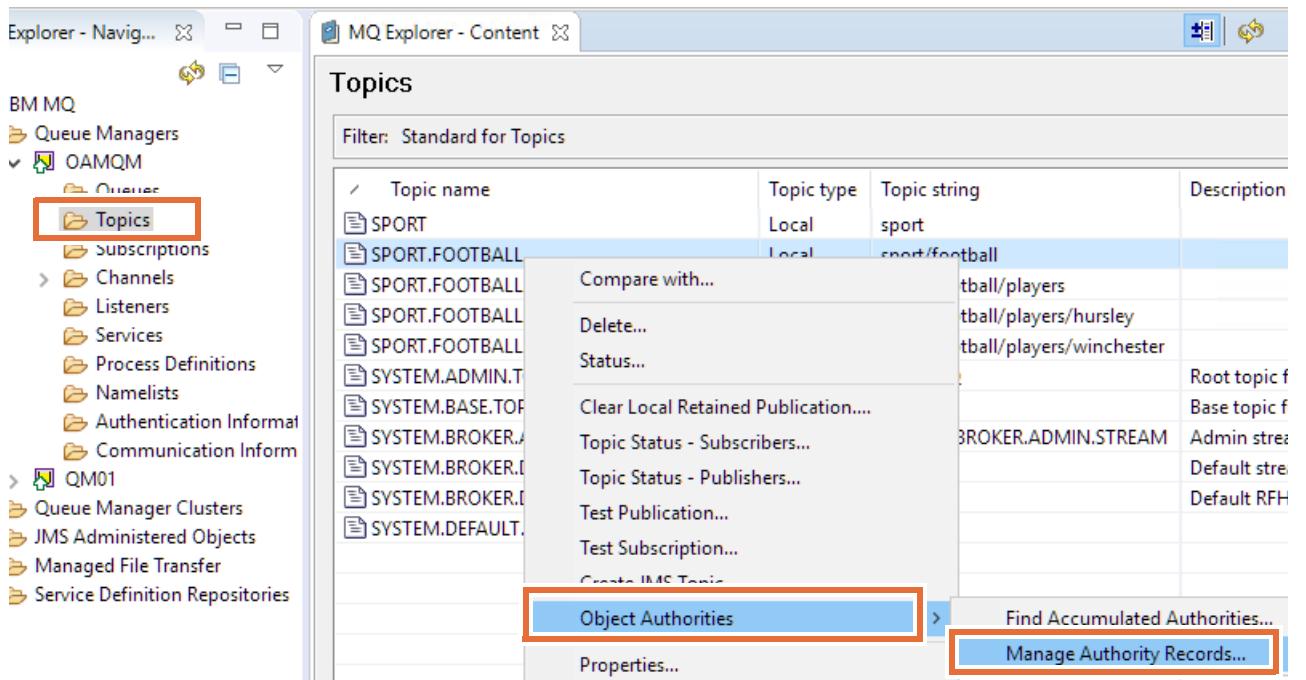
In [Section 1.3, "Grant authorization to the queue manager"](#), the users were granted access to the OAMQM queue manager.

When the amqspub application first attempted to open the topic to PUBLISH a message the OAM checked the topic hierarchy for authorization. Since no authorization was granted for the SPORT administrative topic, the queue manager checked the next level in the hierarchy, which was the SYSTEM.BASE.TOPIC. Since there are no authorizations granted at that level, only members of the mqm group are authorized to publish to the topic.

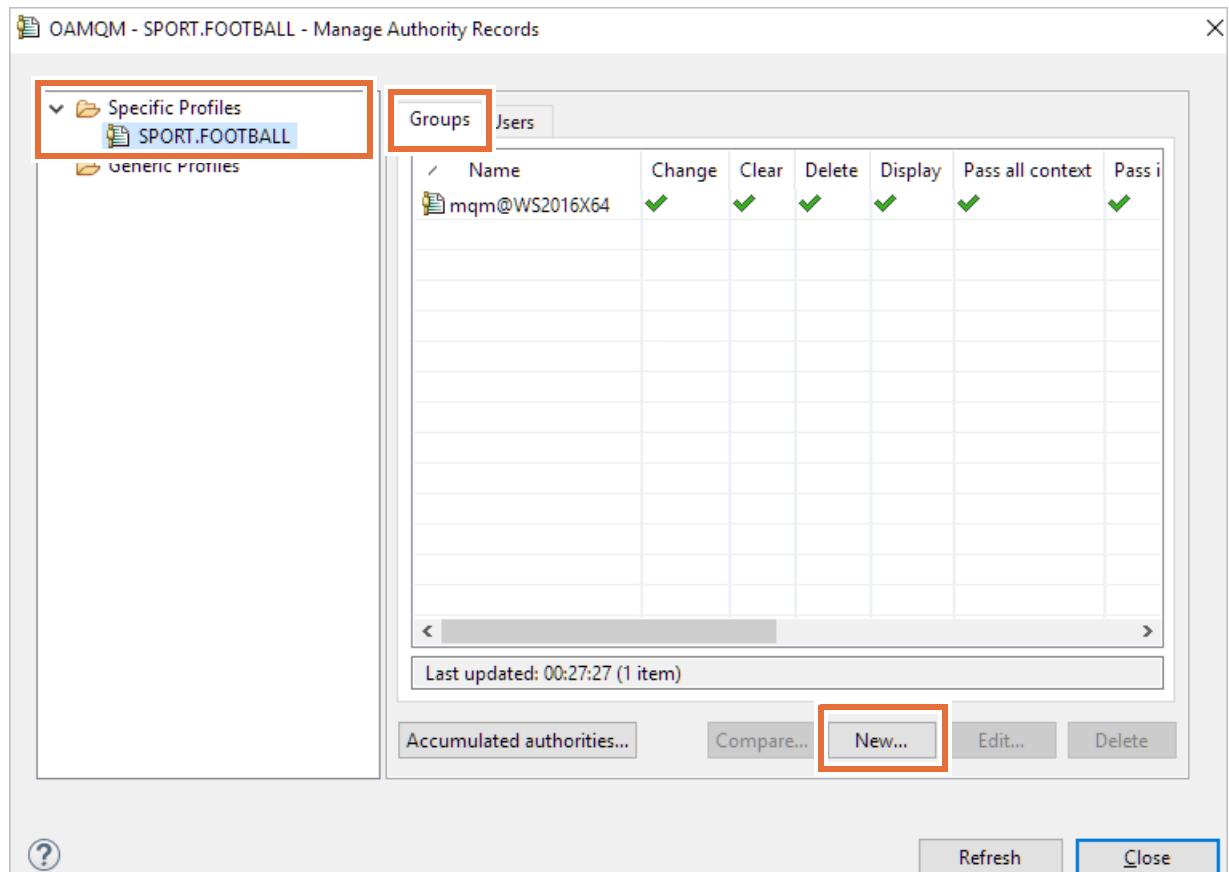
___ 4. In MQ Explorer, grant authorization to the SPORT.FOOTBALL administrative topic for the **mqusers** group.

- ___ a. In the **Navigator** pane, click the **Topics** folder for **OAMQM**.

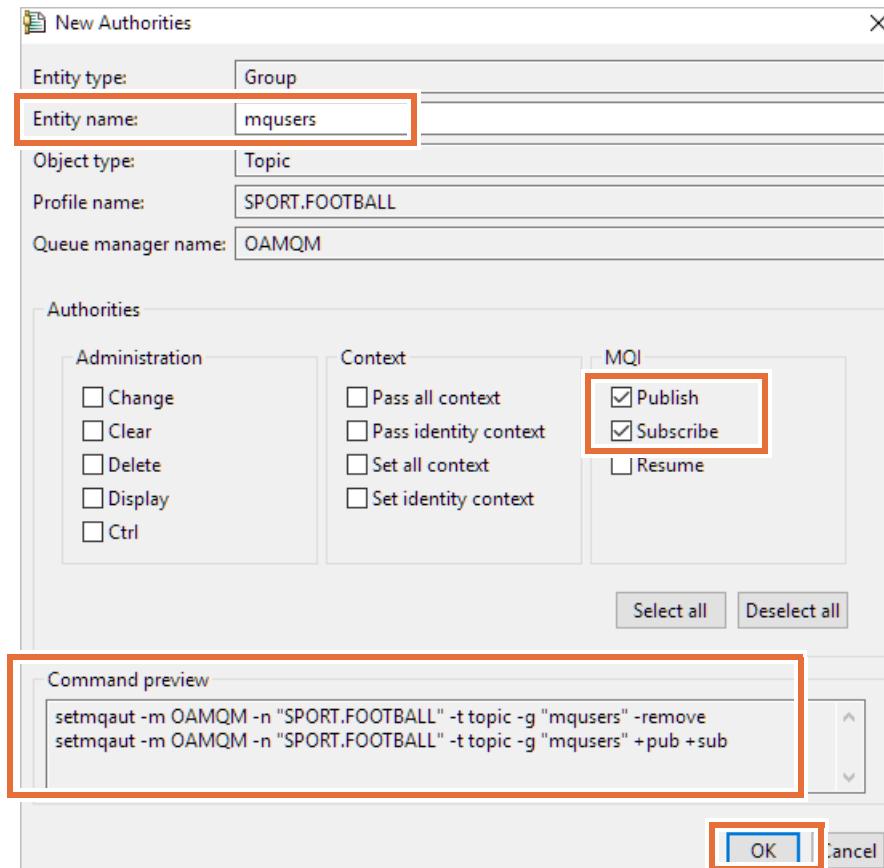
- __ b. Right-click **SPORT.FOOTBALL** and click **Object Authorities > Manage Authority Records**.



- __ c. Expand **Specific Profiles**, click **SPORT.FOOTBALL**, and on the **Groups** tab, click **New**.



- ___ d. In the **Entity name** field, enter: mqusers
- ___ e. In the **MQI** section, select **Publish** and **Subscribe**.
- ___ f. Notice the **Command preview** section to see the commands that are executed and click **OK**.



- ___ 5. Click **OK** to close the confirmation window.
- ___ 6. Click **Close** to close the Manage Authority Records window.
- ___ 7. Try publishing messages for each user ID at the level of the hierarchy where you granted authorization to publish to the members of the `mqusers` group. Use the following command:
`amqspub sport/football OAMQM`
This time, `mquser` could publish a message as a member of the `mqusers` group.
- ___ 8. Try publishing messages on the `sport/football/players/hursley` topic.
`amqspub sport/football/players/hursley OAMQM`



Information

Why can `mqusers` publish messages on the `sport/football/players/hursley` topic?

The queue manager checks for a matching topic at the current topic level to see whether authorization is granted. It then moves up the tree to each successive parent topic object until

an authorization is found. If no authorization is found, then authorization is based on the SYSTEM.BASE.TOPIC topic object.

In this case, since the `mquser` user ID is a member of the `mqusers` group, and the `mqusers` group has publish enabled on the `SPORT.FOOTBALL` topic, authorization to publish was granted.

2.3. Exercise cleanup

- __ 1. Stop the OAMQM queue manager.
- __ 2. Close all command prompts.

End of exercise

Exercise review and wrap-up

The exercise demonstrated how to use the Object Authority Manager (OAM) component of IBM MQ. You learned how to manage access to MQ objects, including queue managers, queues, and publish/subscribe topics.

Exercise 8. Securing channels with TLS

Estimated time

01:30

Overview

In this exercise, you define and start TLS channels between IBM MQ queue managers, and between an IBM MQ client and an IBM MQ server.

Objectives

After completing this exercise, you should be able to:

- Use IBM Key Management to create a certificate request
- Secure channels by using TLS on the channel

Introduction

In this exercise, you set up an environment to demonstrate a hacker eavesdropping on messages that are sent across TCP/IP channels. To resolve this issue, you work with IBM Key Management to create a certificate request.

You also create self-signed certificates and a key database file. After you exchange certificates between queue managers, you test the secure channels. You also define the channel to use distinguished name matching.

This exercise includes these sections:

- [Section 1, "Set up the environment"](#)
- [Section 2, "Eavesdropping on the connection"](#)
- [Section 3, "Setting up the key repository"](#)
- [Section 4, "TLS with IBM MQ clients"](#)

Requirements

This exercise uses a Java proxy program that requires the Java runtime directory (`C:\Program Files\IBM\MQ\java\jre\bin`) to be included in the Path environment variable for your VMware image. This setup is already complete on the image that is provided for this course.

Section 1. Set up the environment

In this first part of this exercise, you create the MQ objects.

1.1. Create the queue managers

- ___ 1. Open MQ Explorer and make sure that no queue managers are running.
- ___ 2. Create two queue managers with these attributes:

Name	Dead-letter queue name	Port
SECQM1	SECQM1.DLQ	9011
SECQM2	SECQM2.DLQ	9012



Hint

To create a queue manager, see [Section 2.1, "Create a queue manager" in Exercise 1, "Getting started with IBM MQ".](#)

- ___ 3. Use MQSC commands to disable channel authentication, connection authentication, and refresh security on each queue manager.



Note

By default, channel authentication and connection authentication are enabled on all new queue managers. For this exercise, you disable channel authentication and connection authentication to avoid connection issues.

- ___ a. Open a command prompt and run the MQSC on SECQM1:

```
runmqsc SECQM1
ALTER QMGR CHLAUTH(DISABLED)
ALTER QMGR CONNAUTH(' ')
REFRESH SECURITY(*)
DEF QL(SECQM1.DLQ) REPLACE
end
```

- ___ b. In the command prompt, and run the MQSC on SECQM2:

```
runmqsc SECQM2
ALTER QMGR CHLAUTH(DISABLED)
ALTER QMGR CONNAUTH(' ')
REFRESH SECURITY(*)
DEF QL(SECQM2.DLQ) REPLACE
end
```

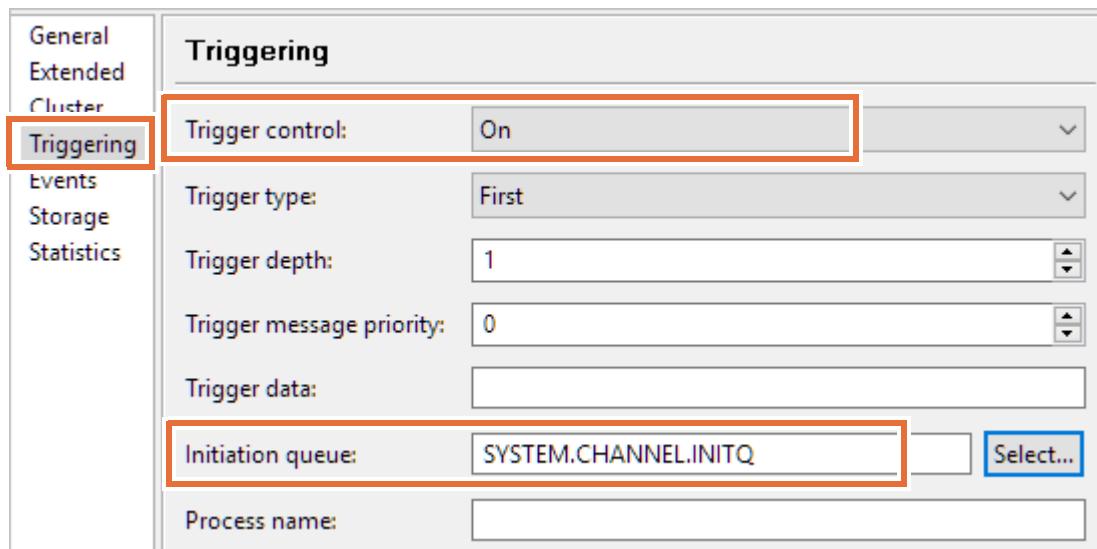
1.2. Create the SECQM1 objects

In this section, you create the following objects on SECQM1:

Local queue manager	SECQM1 (9011)
Transmission queue	SECQM2
Remote queue definition	QRMT
Sender channel	SECQM1 . SECQM2

- ___ 1. In MQ Explorer, create a *transmission queue* on SECQM1 that is named: SECQM2
 - ___ a. In the **Navigator** pane, right-click **SECQM1 > Queues** and click **New > Local Queue**.
 - ___ b. Set **Name** to **SECQM2** and click **Next**.
 - ___ c. Set **Usage** to **Transmission**
 - ___ d. Click the **Triggering** menu and set these properties:
 - **Trigger control:** On
 - **Initiation queue:** SYSTEM.CHANNEL.INITQ

The initiation queue starts the channel automatically.

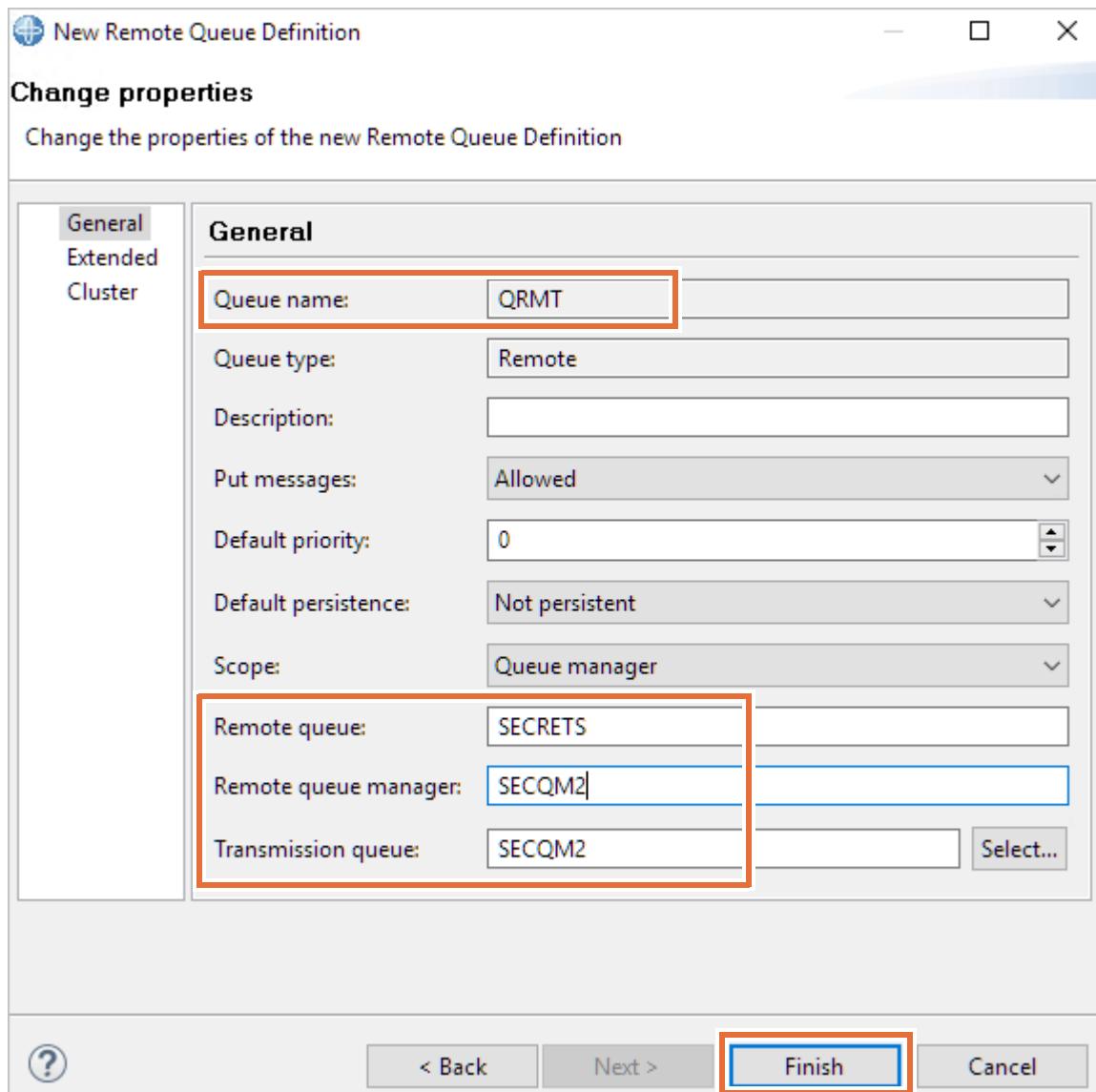


- ___ e. Click **Finish**.
 - ___ f. Click **OK** to close the success message.
- ___ 2. Create a *remote queue definition* with these attributes:

Name	QRMT
Remote queue	SECRETS
Remote queue manager	SECQM2
Transmission queue	SECQM2

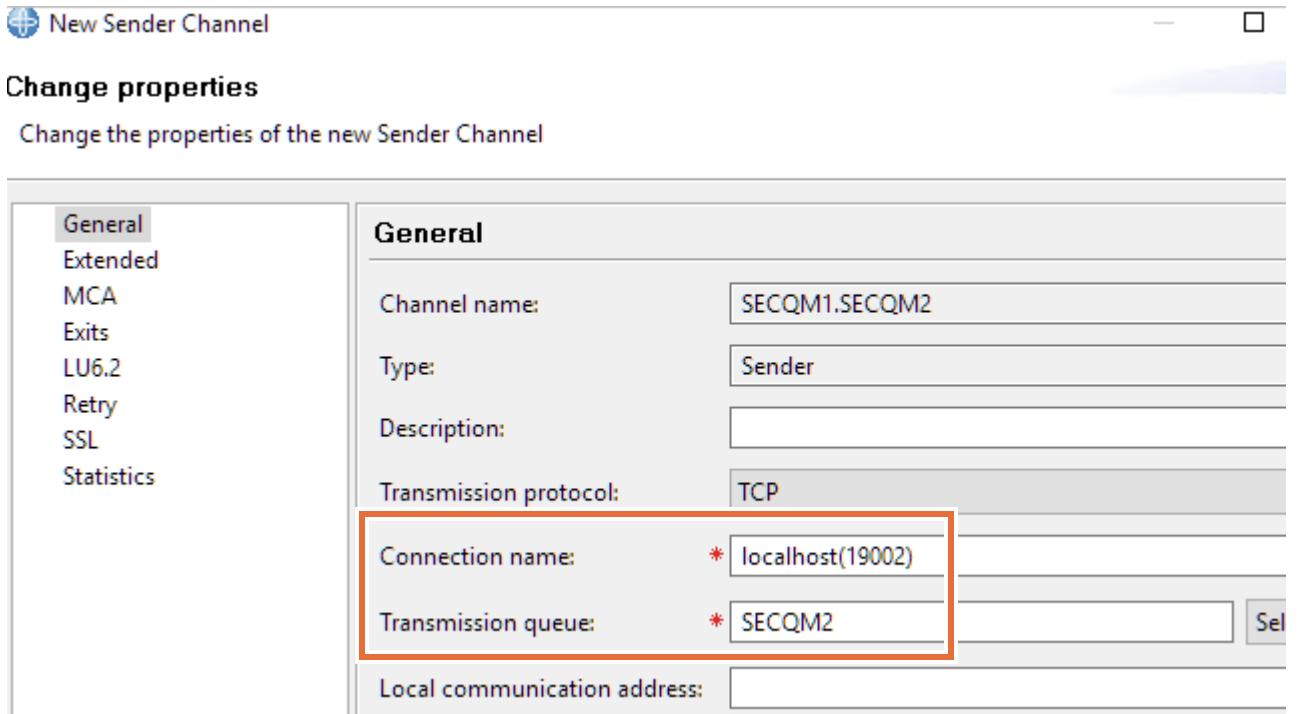
- ___ a. Right-click **SECQM1 > Queues** and click **New > Remote Queue Definition**.
- ___ b. Set **Name** to **QRMT** and click **Next**.

- __ c. Set **Remote queue** to: SECRETS
- __ d. Set **Remote queue manager** to: SECQM2
- __ e. Set **Transmission queue** to SECQM2 and click **Finish**.



- __ f. Click **OK** to close the success message.
3. Create a *sender channel* with these properties:
- | | |
|---------------------------|-------------------|
| Name | SECQM1.SECQM2 |
| Connection Name | localhost (19002) |
| Transmission queue | SECQM2 |
- __ a. Right-click SECQM1 > Channels and click New > Sender Channel.
 - __ b. Set **Name** to SECQM1.SECQM2 and click **Next**.
 - __ c. Set **Connection Name** to: localhost (19002)

- ___ d. Set **Transmission queue** to SECQM2 and click **Finish**.



- ___ e. Click **OK** to close the success message.

1.3. Create the SECQM2 objects

In this section, you create the following objects on SECQM2:

Remote queue manager	SECQM1 (9012)
Local queue	SECRETS
Receiver channel	SECQM1 . SECQM2

- ___ 1. On the SECQM2 queue manager, create a local queue that is called: SECRETS
 - ___ a. Right-click **SECQM2 > Queues** and click **New > Local Queue**.
 - ___ b. Set **Name** to **SECRETS** and click **Finish**.
 - ___ c. Click **OK** to close the success message.
- ___ 2. Create a receiver channel that is called: SECQM1 . SECQM2
 - ___ a. Right-click **SECQM2 > Channels** and click **New > Receiver Channel**.
 - ___ b. Set **Name** to **SECQM1 . SECQM2** and click **Finish**.
 - ___ c. Click **OK** to close the success message.

Section 2. Eavesdropping on the connection

In this part of the exercise, you simulate someone who spies on the communication between two queue managers over an unencrypted channel. You use a proxy program to inspect the communication flows as a message is received on the destination queue manager.

The supplied Java proxy program that is started with the `sps.bat` command receives TCP port requests on the receiving end of the channel. It then forwards requests to the channel listener port and also writes the data when it passes through the proxy to the screen.

The SECQM2 channel listener is listening on port 9012. The SECQM1 . SECQM2 sender channel connects to port 19002. The proxy listens on port 19002 and forwards the messages to port 9012. The port number is changed on the **Connection Name** property of the sending channel. The channel listener remains the same and notices no difference.

- ___ 1. Start the sender channel on SECQM1.
 - ___ a. In the **SECQM1 > Channels** view, right-click `SECQM1 . SECQM2` and click **Start**.
 - ___ b. Click **OK** to close the success message.



Troubleshooting

Continue with the next steps even if the channel status does not show as Running. The status is affected by the proxy port.

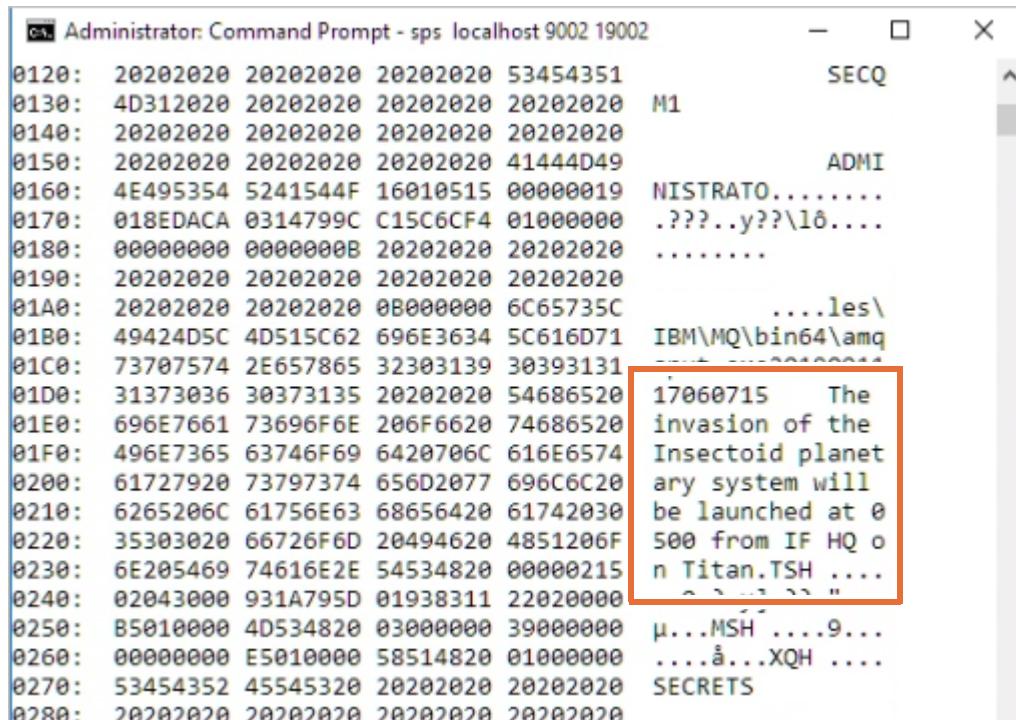
-
- ___ 2. Run the `sps` Java proxy program.
 - ___ a. In the command prompt, switch to the `C:\labfiles\Lab08-tls` directory.
`cd C:\labfiles\Lab08-tls`
 - ___ b. Run the batch file that is named `cp.bat` for the sample Java program.
`cp.bat`
 - ___ c. Start the proxy program against port 9012:
`sps localhost 9012 19002`
The following output is displayed:
`Starting proxy for localhost:9002 on port 19002`
 - ___ 3. Open another command prompt beside the proxy program command prompt, and use `amqsput` to send data from a sample file to the remote queue `SECRETS` on SECQM1.
`amqsput QRMT SECQM1 < C:\labfiles\Lab08-tls\plans.txt`
The sample file contains a secret message.
 - ___ 4. In the command window that is running the proxy program, inspect the data and find the transmitted message.



Troubleshooting

If the message does not appear in the proxy command window, restart the SECQM1.SECQM2 sender channel on SECQM1 and check the proxy command window again.

In the proxy command window, you should be able to find the transmitted message that matches the message in the `plans.txt` file.



```

Administrator: Command Prompt - sps localhost 9002 19002
0120: 20202020 20202020 20202020 53454351 SECQ
0130: 4D312020 20202020 20202020 20202020 M1
0140: 20202020 20202020 20202020 20202020
0150: 20202020 20202020 20202020 41444D49 ADMI
0160: 4E495354 5241544F 16010515 00000019 NISTRATO.....
0170: 018EDACA 0314799C C15C6CF4 01000000 .???.y??\16....
0180: 00000000 0000000B 20202020 20202020 .....
0190: 20202020 20202020 20202020 20202020
01A0: 20202020 20202020 0B000000 6C65735C ....les\
01B0: 49424D5C 4D515C62 696E3634 5C616D71 IBM\MQ\bin64\amq
01C0: 73707574 2E657865 32303139 30393131
01D0: 31373036 30373135 20202020 54686520
01E0: 696E7661 73696F6E 206F6620 74686520
01F0: 496E7365 63746F69 6420706C 616E6574
0200: 61727920 73797374 656D2077 696C6C20
0210: 6265206C 61756E63 68656420 61742030
0220: 35303020 66726F6D 20494620 4851206F
0230: 6E205469 74616E2E 54534820 00000215
0240: 02043000 931A795D 01938311 22020000
0250: B5010000 4D534820 03000000 39000000 μ....MSH ....9...
0260: 00000000 E5010000 58514820 01000000 ....å...XQH ....
0270: 53454352 45545320 20202020 20202020 SECRETS
0280: 20202020 20202020 20202020 20202020

```



Note

You can also browse the messages on the SECRETS queue of the SECQM2 queue manager.

- ___ 5. Leave the `sps` proxy program running in the command window.
- ___ 6. Stop the sender channel.

Section 3. Setting up the key repository

In this part of the exercise, you begin security configuration for the queue managers by creating a key repository for each queue manager.

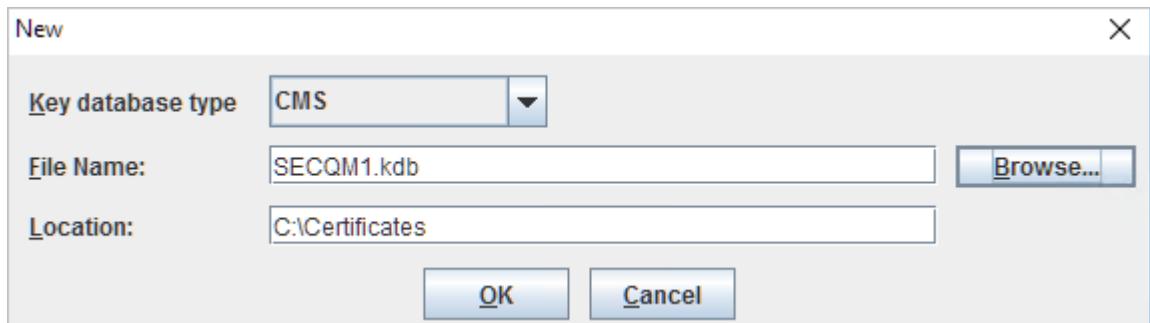


Important

To avoid a TLS mismatch later in this exercise, make sure that you type the label and file names exactly as they are shown in the exercise instructions.

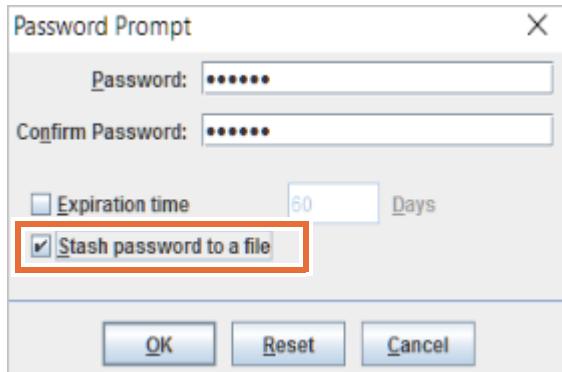
3.1. Creating certificates

- 1. In Windows Explorer, create a directory that is named: **C:\Certificates**
You use this directory to store the new key and certificates that you create in this exercise.
- 2. Go to MQ Explorer, and in the **Navigator** pane, right-click **IBM MQ** and click **Manage SSL Certificates** to open IBM Key Management.
- 3. In IBM Key Management, generate a new key database file for the SECQM1 queue manager.
 - a. Click **Key Database File > New**.
 - b. Keep **CMS** as the **Key database type**.
 - c. Set **File Name** to: **SECQM1.kdb**
 - d. Set **Location** to: **C:\Certificates**
 - e. Click **OK**.



- f. In the Password Prompt window, set **Password** and **Confirm Password** to: **mqpass**

- ___ g. Select **Stash password to a file**, which stores the password in the C:\Certificates directory.



- ___ h. Click **OK**.



Troubleshooting

If you get read/write errors when you try to save the file:

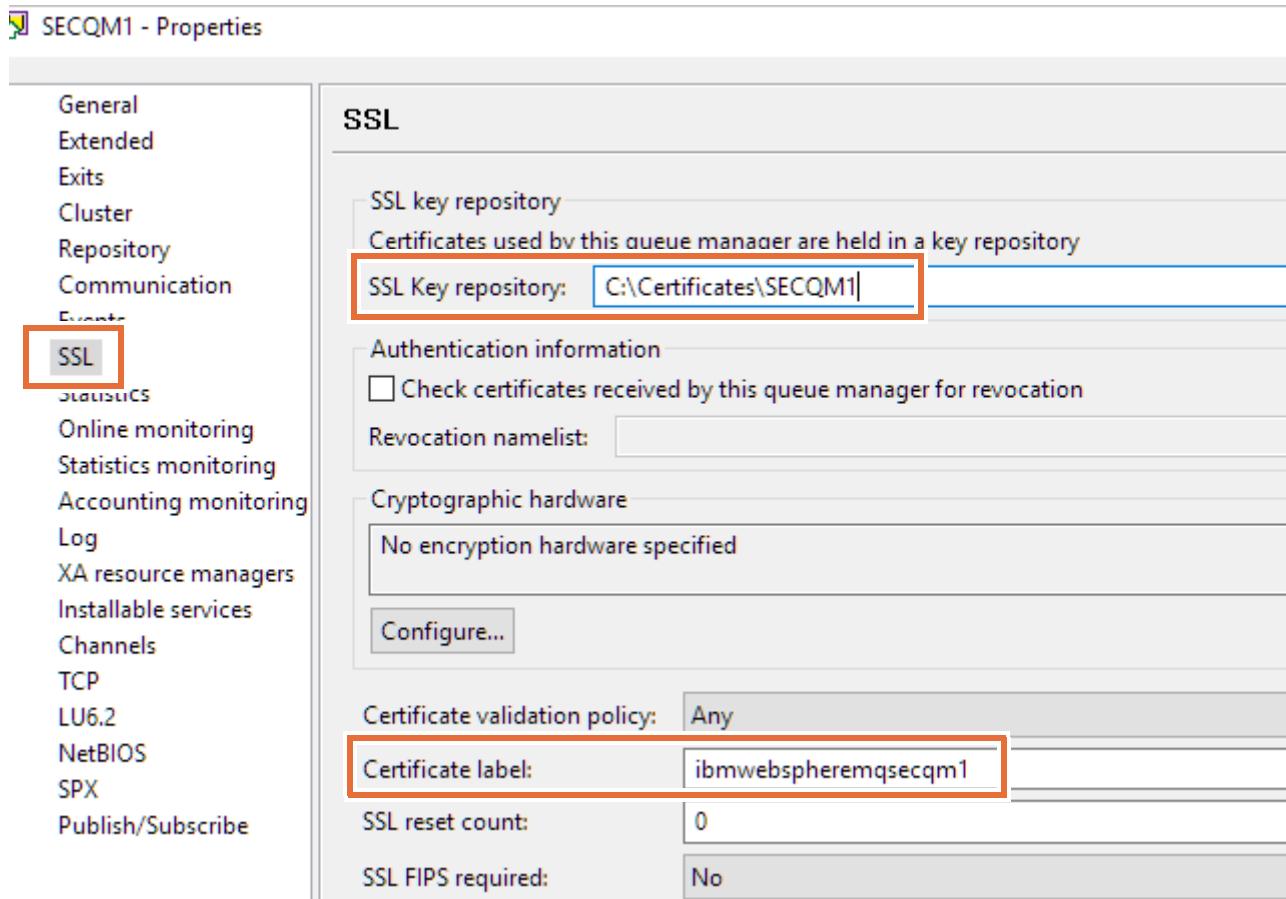
- Repeat [Step 3](#), and for [Step d](#), click **Browse** beside the **File Name** field.
- Browse to the **C:\Certificates** folder to ensure that you save to the correct folder.

- ___ 4. Generate a new key database file for the SECQM2 queue manager.
- Click **Key Database File > New**.
 - Keep **CMS** as the **Key database type**.
 - Set **File Name** to: **SECQM2.kdb**
 - Make sure that **Location** is set to **C:\Certificates** and click **OK**.
 - Set the password to **mqpass**, select **Stash password to a file**, and click **OK**.
- ___ 5. Minimize IBM Key Management, but leave it open.

3.2. Update the queue managers with the key database files

1. In MQ Explorer, specify the location of your queue manager key database file on SECQM1.
 - In the **Navigator** pane, right-click **SECQM1** and click **Properties**.
 - In left menu pane, click **SSL**.
 - Change the **SSL Key repository** field to the location and name of your new key database file: **C:\Certificates\SECQM1**

You do not need to specify the `.kdb` file extension. Notice that the certificate label that is created automatically in the **Certificate label** field.



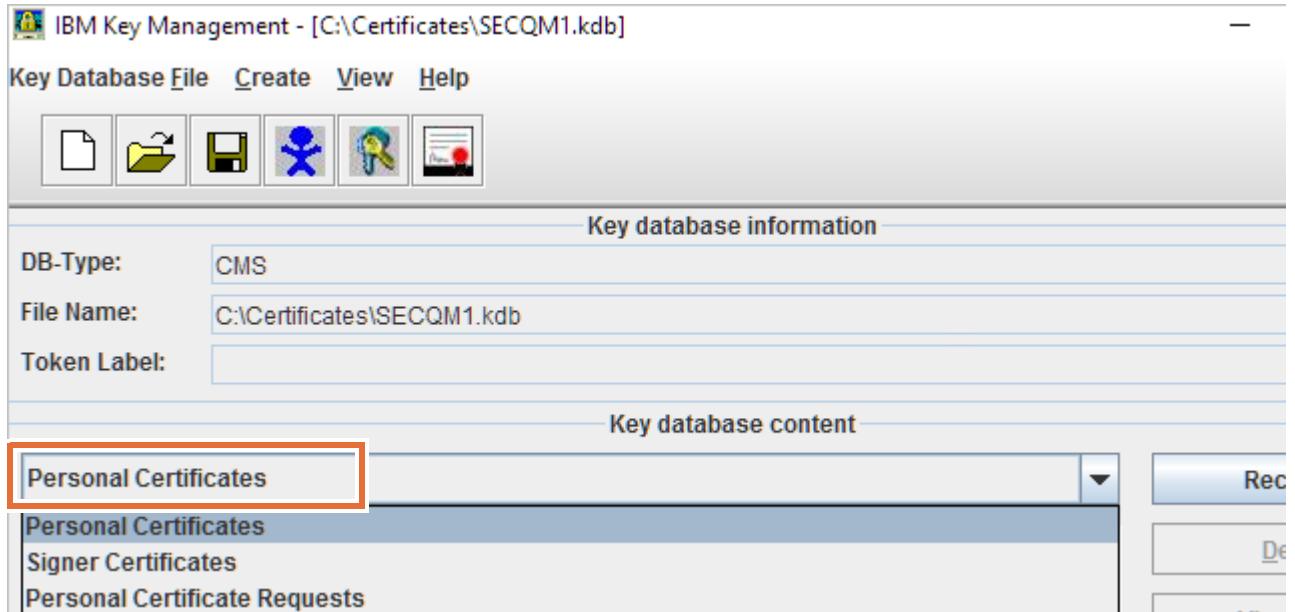
- ___ d. Click **OK** and click **Yes** when prompted to confirm.
- ___ 2. Set the SSL properties for SECQM2.
 - ___ a. In the **Navigator** pane, right-click **SECQM2** and click **Properties**.
 - ___ b. In left menu pane, click **SSL**.
 - ___ c. Change the **SSL Key repository** field to the location and name of your new key database file: `C:\Certificates\SECQM2`
 - ___ d. Click **OK** and then click **Yes**.

3.3. Create self-signed certificates for the queue managers

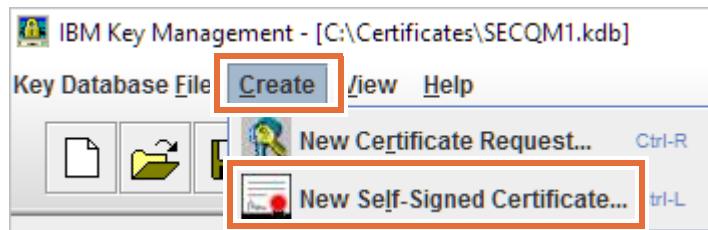
In this part of the exercise, you create self-signed certificates for testing purposes.

- ___ 1. Return to IBM Key Management and open the `SECQM1.kdb` file.
 - ___ a. Click **Key Database File > Open** and click **Browse**.
 - ___ b. Select `SECQM1.kdb`, click **Open** and click **OK**.
 - ___ c. Enter `mqpass` as the password and click **OK**.

2. In the **Key database content** menu, make sure that **Personal Certificates** is selected.

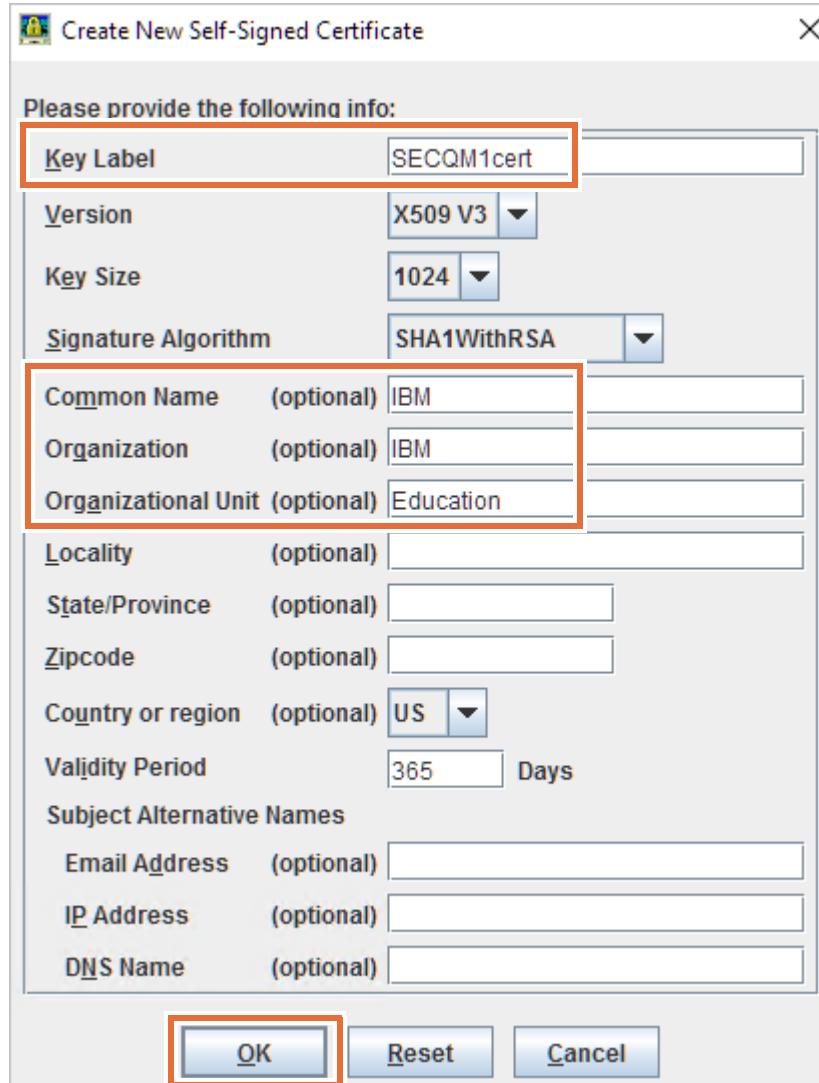


3. Click **Create > New Self-Signed Certificate**.



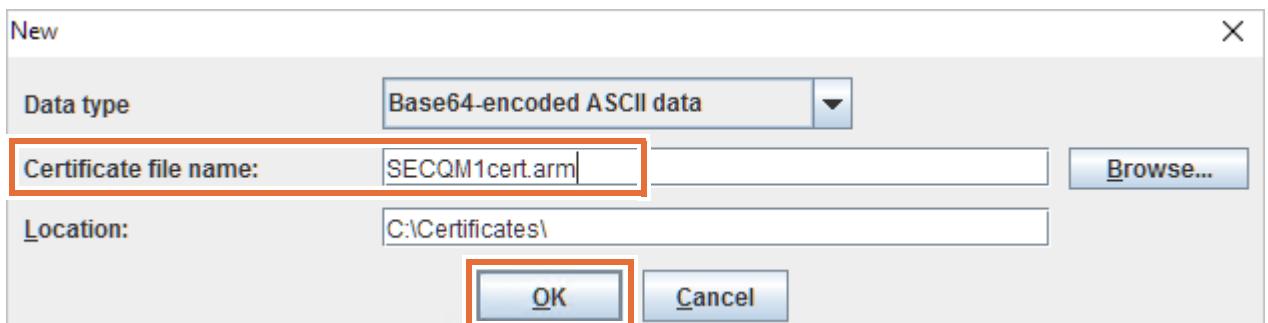
4. Enter the following values and click **OK**.

- **Key Label:** SECQM1cert
- **Common Name:** IBM
- **Organization:** IBM
- **Organization Unit:** Education



5. In the **Personal Certificates** list, make sure that **SECQM1cert** is selected and click **Extract Certificate**.

6. Set **Certificate file name** to **SECQM1cert.arm** and click **OK**.

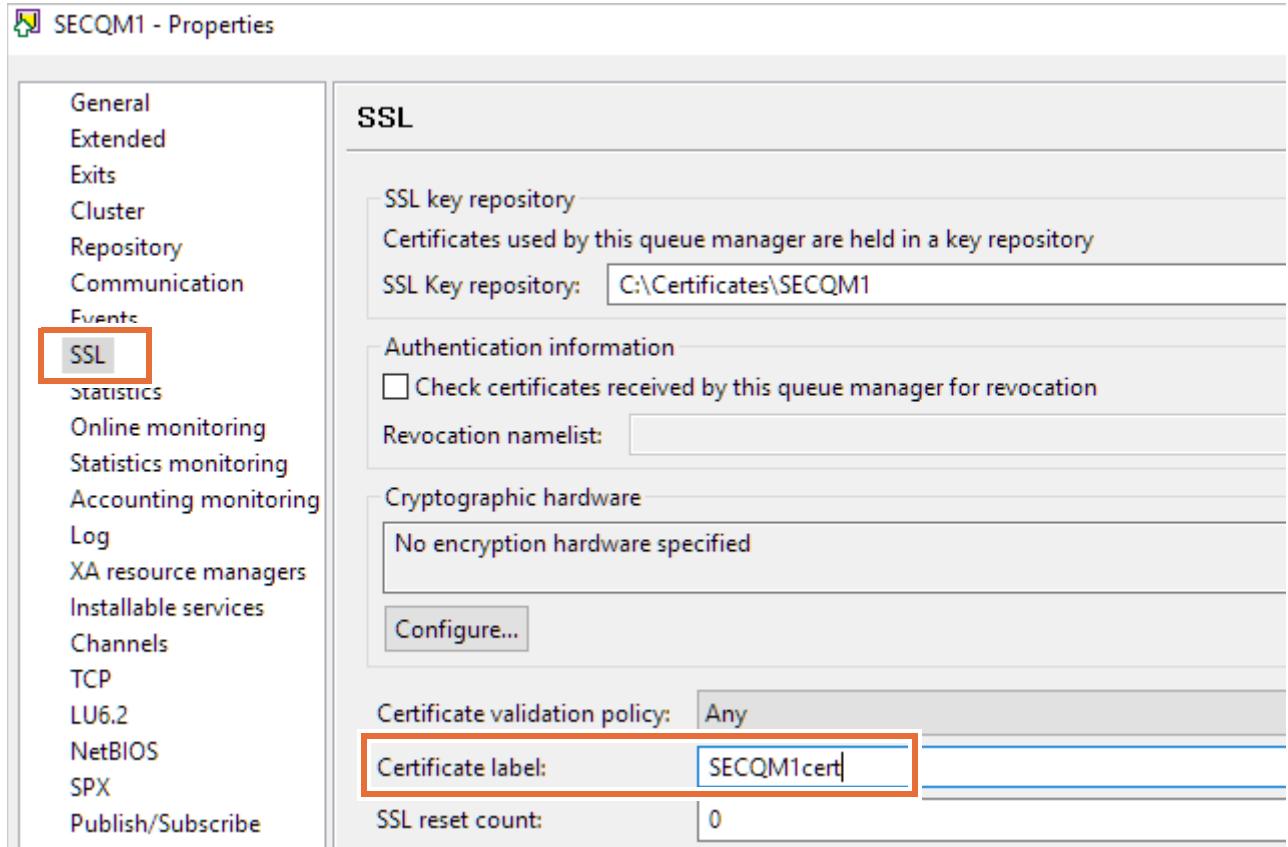


- ___ 7. Create a self-signed certificate for SECQM2.
 - ___ a. Click **Key Database File > Open**.
 - ___ b. Browse to the `SECQM2.kdb` file, click **Open** and click **OK**.
 - ___ c. Enter `mqpass` as the password and click **OK**.
 - ___ d. Click **Create > New Self-Signed Certificate**, enter the following values:
 - **Key Label:** SECQM2cert
 - **Common Name:** IBM
 - **Organization:** IBM
 - **Organization Unit:** Education
 - ___ e. Click **OK**.
 - ___ f. In the **Personal Certificates** list, make sure that **SECQM2cert** and click **Extract Certificate**.
 - ___ g. Set the **Certificate file name** field to `SECQM2cert.arm` and click **OK**.
- ___ 8. Minimize IBM Key Management.

3.4. Update the SSL certificate label properties on the queue managers

- ___ 1. Go to MQ Explorer and change the SSL Certificate Label property for the SECQM1 queue manager.
 - ___ a. In the **Navigator** pane, right-click **SECQM1** and click **Properties**.

- __ b. Click **SSL** and change **Certificate Label** to: SECQM1cert



- __ c. Click **OK**.
- __ 2. Change the SSL Certificate Label property for the SECQM2 queue manager.
- Right-click **SECQM2** and click **Properties**.
 - Click **SSL** and change **Certificate Label** to **SECQM2cert** and click **OK**.

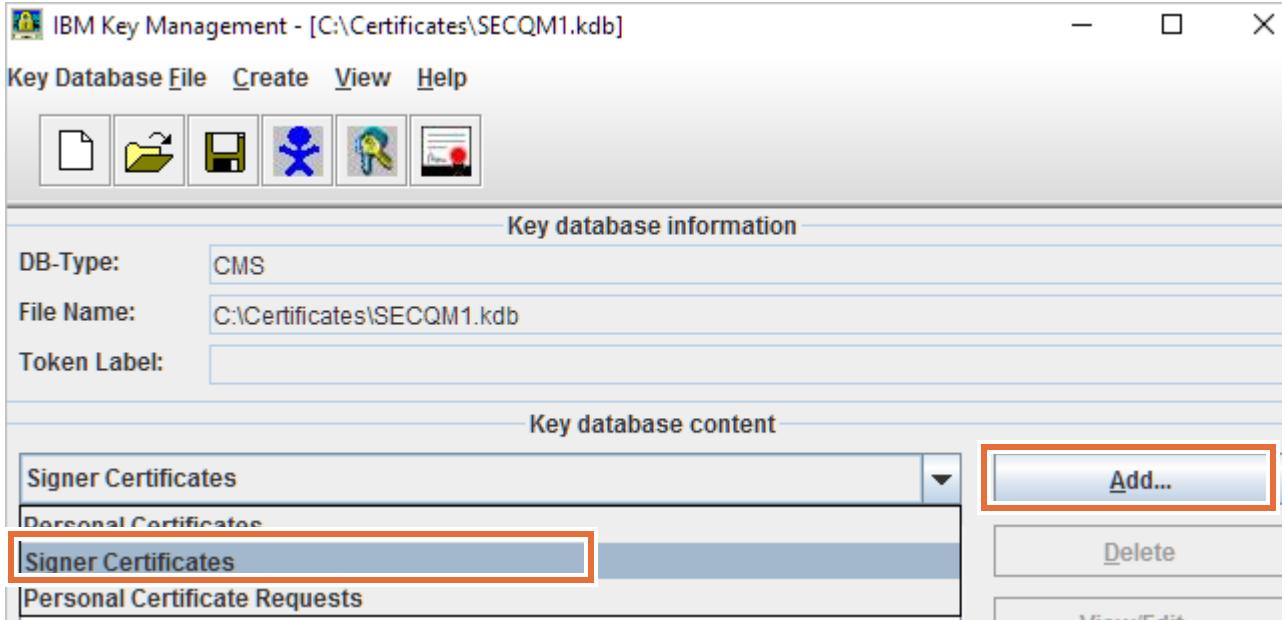
3.5. Exchange certificates

Self-signed certificates cannot be used as a proof of identity because they cannot be verified. You must manually import the self-signed certificate from the other queue manager and accept it as a CA certificate so that your queue manager accepts the self-signed certificate.

- Reopen **SECQM1.kdb** in IBM Key Management.

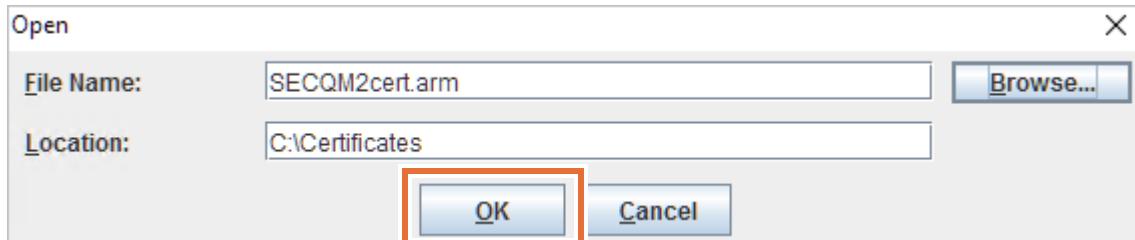
__ 2. Add a signer certificate.

___ a. In the **Key database content** list, select **Signer Certificates** and click **Add**.



___ b. Click **Browse** and select the SECQM2cert.arm file in the C:\Certificates folder and click **Open**.

___ c. Click **OK**.



___ d. Set the label to SECQM2cert and click **OK**.



The certificate is now in the list of Signer Certificates.

__ 3. Reopen SECQM2.kdb with IBM Key Management and add the signer certificate for SECQM2 (as described in [Step 2](#)).

___ a. Make sure that **Signer Certificates** is selected from the **Key database content** list, and click **Add**.

___ b. Browse to the SECQM1cert.arm file to open it and set the label to: SECQM1cert

__ 4. Minimize IBM Key Management.

3.6. Defining your TLS channels

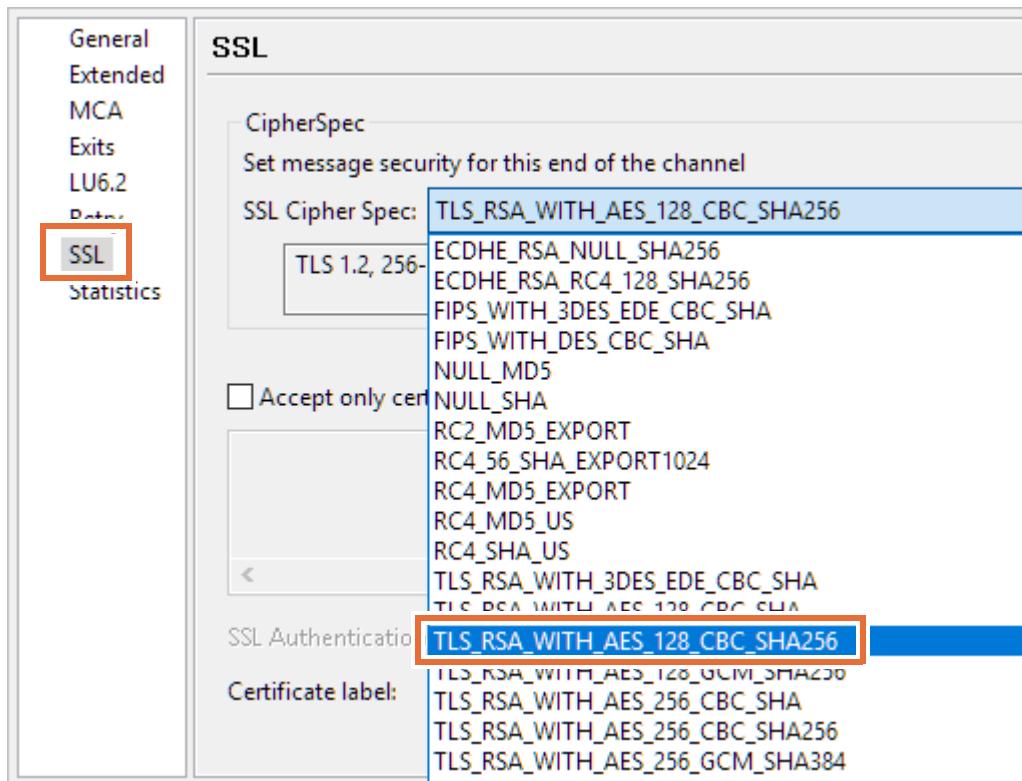
In this part of the exercise, you update your channels to use TLS.



Information

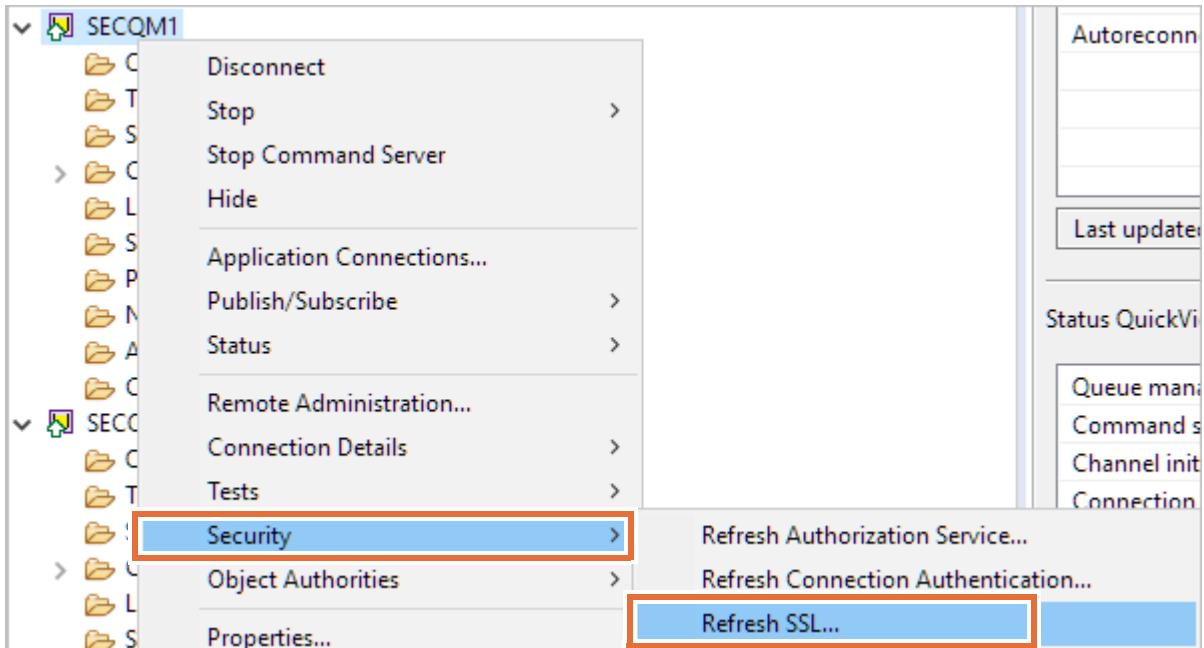
In practice, you should always define and test the new channels without TLS first as you did in this exercise. After you successfully send and receive messages on the new channels, then you enable TLS on the channels.

- 1. In MQ Explorer, update the sender channel on the SECQM1 queue manager to use TLS.
 - a. In the **Navigator** pane, click **SECQM1 > Channels**, double-click **SECQM1.SECQM2** to open **Properties**.
 - b. Click the **SSL** menu and set **SSL Cipher Spec** to: `TLS_RSA_WITH_AES_128_CBC_SHA256`



- c. Click **OK**.
- 2. On SECQM2, update the receiver channel to use TLS.
 - a. Click **SECQM2 > Channels**, double-click **SECQM1.SECQM2** to open **Properties**.
 - b. Click the **SSL** menu and set **SSL Cipher Spec** to: `TLS_RSA_WITH_AES_128_CBC_SHA256`
 - c. Click **OK**.

- ___ 3. Refresh the security cache on the queue managers.
 - ___ a. In the Navigator pane, right-click **SECQM1** and click **Security > Refresh SSL**, and then click **Yes** when prompted to confirm.



- ___ b. Right-click **SECQM2** and click **Security > Refresh SSL**, and then click **Yes** when prompted to confirm.



Information

Alternative ways to refresh the security:

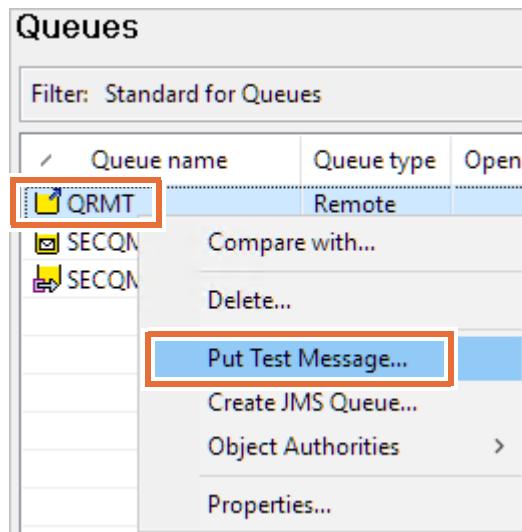
- In a command prompt, run the following MQSC command on each queue manager:
REFRESH SECURITY TYPE (SSL)
- In IBM MQ Console, select the queue manager in the **Local Queue Managers** widget, click the three dots (...) to open the menu and click **Refresh Security**. Then, click **SSL**.

3.7. Test the TLS channels

- ___ 1. Start the sender channel on SECQM1.
 - ___ a. Click **SECQM1 > Queues** and right-click the **SECQM1.SECQM2** sender channel
 - ___ b. Click **Start**.

__ 2. Put a test message on the queue.

__ a. In the **SECQM1 > Queues** view, right-click **QRMT** and click **Put message**.



__ b. Type a message such as: Testing TLS

__ c. Click **Put message** and click **Close**.

__ 3. Verify that the message was sent to SECRETS queue on SECQM2 by right-clicking the queue and clicking **Browse Messages**.

You should see your message, along with the previous messages sent during the first part of the exercise.



Troubleshooting

If the message is stuck on the transmission queue (SECM2):

- Make sure that the SECQM1.SECQM2 *sender* channel is running. If the channel is not running, start it manually.
- Check the dead-letter queues on both sides of the channel.

- 4. Look at the window that is running the proxy program and search for IBM and Education in the data. These distinguished names are passed as part of the TLS handshake.

```

Administrator: Command Prompt - sps localhost 9002 19002

Client-->Server (830 bytes)
0000: 16030303 390B0002 27000224 00022130 ....9...'$..!0
0010: 82021D30 820186A0 03020102 02045DD9 ?..0?.? .....]??
0020: C37B300D 06092A86 4886F70D 01010505 ?{0...*?H?+.....?
0030: 00303D31 0B300906 03550406 13025553 0-1 0 " " "c
0040: 310C300A 06035504 0A130349 424D3112 1.0....U....IBM1.
0050: 30100603 55040B13 09456475 63617469 0....U....Educati
0060: 6F6E310C 300A0603 55040313 0349424D on1.0....U....IBM
0070: 301E170D 31393131 32333233 34303433 0....1222234043Z
0080: 5A170D32 30313132 32323334 3034335A Z..201122234043Z
0090: 303D310B 30090603 55040613 02555331 0-1 0 " " "c1
00A0: 0C300A06 0355040A 13034942 4D311230 .0....U....IBM1.0
00B0: 10060355 040B1309 45647563 6174696F ...U....Educatio
00C0: 6E310C30 0A060355 04031303 49424D30 n1.0....U....IBM0
00D0: 819F300D 06092A86 4886F70D 01010105 rrr...-rhr+.....
00E0: 0003818D 00308189 02818100 C7E29D7E ..???.0???.???.~
00F0: DD26DE94 0516553B B071348D 1D415F4E ?&??..U;^q4?.A_N
0100: E2F66DDE 12CEF1F5 8F6F8A60 D99B1EB0 äöm?.?ñ??o?`???.°
0110: F2125271 5B3CC40E E2837B87 6E64432E ö.Rq[<Ä.ä?{?ndC.
0120: 50540384 81167FDD 81A75933 CC6CDEA3 PT.??..@???Y3?1?£
0130: 80288C3E FDB31FBA BF59A290 DB10663C ?(>??..@?Y¢??..f<
0140: D83DDE33 4A28FF32 ADF7ECF2 CD0C4D96 ?=?3J(ÿ2?+ìò?.M?
0150: 1EFDDC28 A6D6BB12 66E6F43F 37B39102 .?Ü(?Ö».fæô?7??.
```

You see lots of data, however, the data is not readable.

- 5. Stop the SECQM1.SECQM2 *sender* channel on the SECQM1 queue manager.

Section 4. TLS with IBM MQ clients

In this part of the exercise, you create a keystore for use with an IBM MQ client to verify and communicate securely over a TLS client connection channel. You use both the IBM MQ client and server on the same computer. You also use the Java proxy program to observe communications between the client and the server over the TCP/IP localhost connection.

In this section, you test the following scenarios:

- No SSL settings to verify the client configuration.
- Cipher specification `TLS_RSA_WITH_AES_128_CBC_SHA256` (Secure Hash Algorithm, 128-bit AES encryption). This setting is a strong security scheme that can be used for communications across the internet.
- Distinguished name matching so that you can specify who can connect to a particular client connection.

4.1. Set up the client and server channels

- 1. Run the proxy against SECQM1 instead of SECQM2.
 - a. Stop the proxy program by opening the command window that is running the **sps** proxy program and pressing Ctrl+C.
Note: You might need to press Ctrl+C more than once.
 - b. When prompted to terminate the batch job, type: `y`
 - c. Restart the proxy program to run against SECQM1 by entering the following command:
`sps localhost 9011 19001`
- 2. In MQ Explorer, create a *client-connection* channel on **SECQM1**.
 - a. In the **Navigator** pane, expand **SECQM1 > Channels**, right-click **Client Connection** and click **New > Client-connection Channel**.
 - b. Set **Name** to `CLIENT.TLS` and click **Next**.
 - c. Set **Connection name** to: `localhost(19001)`
Port 19001 is the proxy port. If you use 9011, the channel functions, but you do not see any output in the proxy window.
 - d. Click **Finish** and click **OK** to close the success message.
- 3. Create a server-connection channel on **SECQM1**.
 - a. Right-click **SECQM1 > Channels** and click **New > Server-connection Channel**.
 - b. Set **Name** to `CLIENT.TLS` and click **Finish**.
 - c. Click **OK** to close the success message.
- 4. Create a local queue on SECQM1 that is named: `DEPOSIT1`

4.2. Test without TLS

- 1. Open a new command prompt beside the command prompt that is running the proxy program, and enter the following commands to set the MQCHLLIB and MQCHLTAB environment variables:

```
set MQCHLLIB=c:\ProgramData\IBM\MQ\qmgrs\SECQM1@\ipcc
set MQCHLTAB=AMQCLCHL.TAB
```



Information

You cannot use the MQSERVER environment variable for channels with TLS attributes. In this step, you use the client channel definition table that is generated automatically in the following directory:

```
c:\ProgramData\IBM\MQ\qmgrs\SECQM1@\ipcc\AMQCLCHL.TAB
```

Normally, you would first need to copy this file to the client computer. In this exercise, you are using the same computer for both the client and server so you do not need to copy the file.

- 2. In the same command window that you set the environment variables, run the `amqsputc` sample client program to put a test message.

- a. Run the client program to put a message.

```
amqsputc DEPOSIT1
```

- b. Type the following message.

```
Client message to DEPOSIT1
```

- c. Press Enter twice to end the program.

- 3. Look at the proxy window. You should see that the channel CLIENT.TLS was used and you should be able to find your message.

```
C:\labfiles\Lab08-tls>REM This batch file start the Java proxy program
Starting proxy for localhost:9001 on port 19001

Client-->Server (268 bytes)
0000: 54534820 0000010C 02010100 00000000 TSH .....
0010: 00000000 22020000 B5010000 49442020 .....".µ...ID
0020: 0D250000 00000000 EC7F0000 00004000 ~~~~~~G
0030: 00000000 434C4945 4E542E54 4C532020 .....
0040: 20202020 20202020 D400B501 20202020 .....
0050: 20202020 20202020 20202020 20202020 .....
```

....CLIENT.TLS

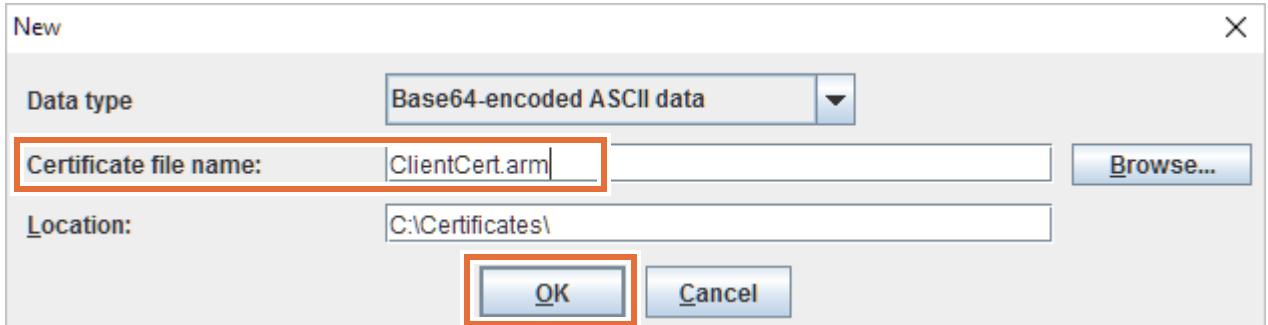
```
01D0: 00000000 00000000 00000000 00000000 .....
01E0: 00000000 00000000 00000000 00000000 .....
01F0: 00000000 00000000 1A000000 436C6965 .....Clie
0200: 6E74206D 65737361 67652074 6F204445 nt message to DE
0210: 504F5349 54310000 POSIT1..
```

.....Clie
nt message to DE
POSIT1..

```
Server-->Client (508 bytes)
```

4.3. Create a keystore for the client connection

- ___ 1. In IBM Key Management, create a keystore in C:\Certificates that is called client.kdb
 - ___ a. Click **Key Database File > New**.
 - ___ b. Set **File Name** to client.kdb, keep **Location** as: C:\Certificates and click **OK**.
 - ___ c. Set the password to `mopass`, select **Stash password to a file** and click **OK**.
- ___ 2. In IBM Key Management, create a self-signed certificate for your new client.kdb client.
 - ___ a. Click **Create > New Self-Signed Certificate**.
 - ___ b. Set these fields:
 - **Key Label:** MQClient
 - **Common Name:** IBM
 - **Organization:** IBM
 - **Organization Unit:** Education
 - ___ c. Click **OK**.
- ___ 3. Extract the certificate to a file.
 - ___ a. In the **Key database content** list, select **Personal Certificates**.
 - ___ b. Select the new **MQClient** certificate from the list of **Personal Certificates** and click **Extract Certificate**.
 - ___ c. Set **Certificate file name** to ClientCert.arm and click **OK**.



- ___ 4. Add the signer certificate.
 - ___ a. Switch the **Key database content** list to **Signer Certificates** and click **Add**.
 - ___ b. Click **Browse** to select SECQM1cert.arm and click **Open**.
 - ___ c. Click **OK**.
 - ___ d. Set the label to SECQM1cert and click **OK**.

This step is required so that the IBM MQ client can verify the identity that is passed on the certificate from SEQM1.
- ___ 5. Add the newly exported client certificate ClientCert.arm to the SECQM1 certificate store as a signer certificate.
 - ___ a. Click **Key Database File > Open**, browse to **SECQM1.kdb**, click **Open** and click **OK**.
 - ___ b. Enter `mopass` as the password and click **OK**.

- ___ c. In the **Key database content** list, select **Signer Certificates** and click **Add**.
- ___ d. Click **Browse** to select the ClientCert.arm file in the c:\Certificates folder, click **Open** and click **OK**.
- ___ e. Set the label to `MQClient` and click **OK**.

The certificate is now added to the list of Signer Certificates (which already includes SECQM2cert).

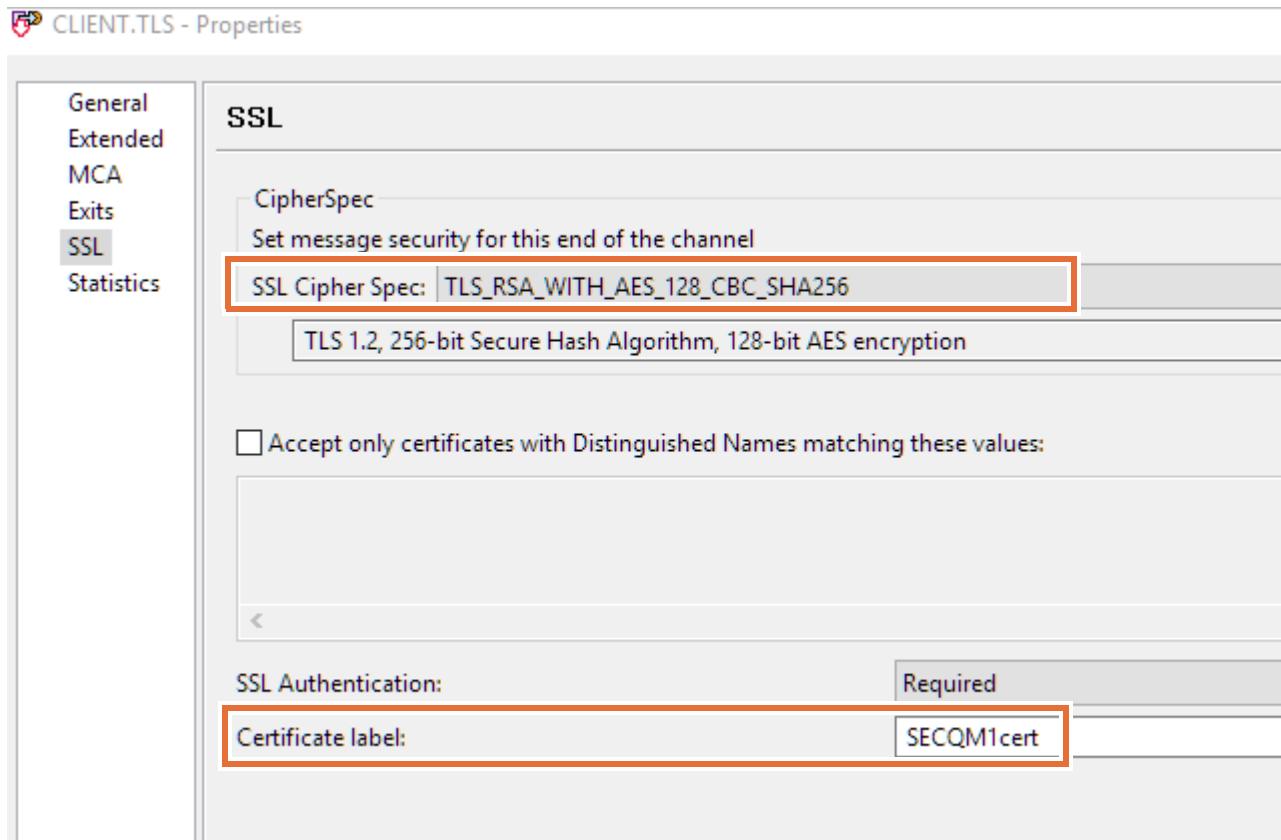
- ___ 6. Minimize IBM Key Management.
- ___ 7. In the command window where you set the MQCHLLIB and MQCHLTAB environment variables, enter the following command to set the keystore for the IBM MQ client:

```
set MQSSLKEYR=C:\Certificates\client
```

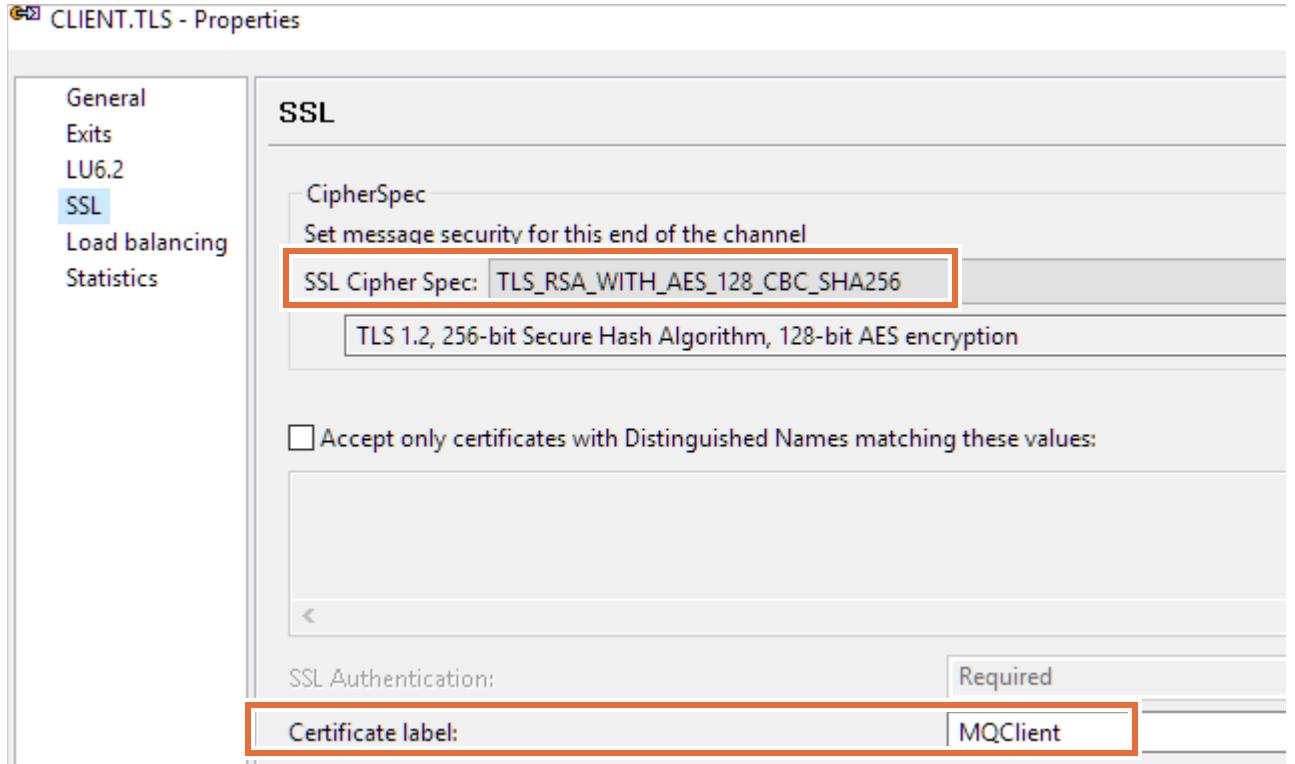
Note: The `.kdb` file extension is not required on the `MQSSLKEYR` variable.

4.4. Test with TLS

- ___ 1. In MQ Explorer, set the SSL (TLS) properties of the *server-connection* channel on SECQM1 to use a cipher specification.
 - ___ a. Click **SECQM1 > Channels** and in the Channels view, double-click **CLIENT.TLS** to open **Properties**.
 - ___ b. Click the **SSL** menu, and set **SSL Cipher Spec** to: **TLS_RSA_WITH_AES_128_CBC_SHA256**.
 - ___ c. Set the **Certificate label** to: `SECQM1cert`



- ___ d. Click **OK**.
- ___ 2. Set the SSL (TLS) properties of the *client-connection* channel to use a cipher specification.
 - ___ a. Click **SECQM1 > Channels > Client Connections** and in the Client Connections view, double-click **CLIENT.TLS** to open **Properties**.
 - ___ b. Click the **SSL** menu, and set **SSL Cipher Spec** to: **TLS_RSA_WITH_AES_128_CBC_SHA256**.
 - ___ c. Set **Certificate label** to: **MQClient**



- ___ d. Click **OK**.
- ___ 3. Return to the command window where you set the environment variables, and repeat the client *put* test. Make sure that the proxy command window is also visible.

```
amqsputc DEPOSIT1
```

- ___ 4. Look at the proxy window.

Can you find your message? More information is passed back and forth between the client and server, including certificates for mutual authentication.

It is important to note that message hashes are generated for the data that is passed. Each side can use these message hashes to verify that the passed data was not tampered with.



Troubleshooting

Many things can go wrong with TLS client channels. Sometimes, the problems are not easy to figure out. These troubleshooting suggestions might help you find the problem.

- Define the channels without TLS first and ensure that they work correctly. Complete an end-to-end test without TLS, by using a sample program such as `amqsputc`.
- Make sure that the client certificate is labeled correctly.
- If you are using self-signed certificates, ensure that your client certificate is imported into your queue manager certificate store as a CA signer certificate.
- If you receive MQRC 2538 errors, try setting your queue manager listener to use IP address `localhost`, then restart the listener and rerun the test.
- Refresh SSL (TLS) on the queue manager.

4.5. Test distinguished name matching

- 1. In IBM MQ Explorer, change the SSL properties of the CLIENT.TLS *server-connection* channel on SECQM1 to accept only certificates with matching Distinguished Names.
 - a. Click the **SECQM1 > Channels** folder and double-click **CLIENT.TLS** (which is the *server-connection* channel) to open the properties.
 - b. On the **SSL** page, select **Accept only certificates with Distinguished Names matching these values** and enter these values:
CN=IBM, OU=Education

This value is the SSLPEER value and should match the DN in your existing client certificate.

Accept only certificates with Distinguished Names matching these values:
CN=IBM,OU=Education

SSL Authentication: Required
Certificate label: SECQM1cert

- ___ c. Click **OK**.
- ___ 2. Repeat the `amqsputc` test. The results should not be different from previous tests.
- ___ 3. Now test with a DN that does not match.
 - ___ a. In the SSL properties for the channel CLIENT.TLS, modify the DN string to use the following string and click **OK**.
CN=IBM, OU=Research
 - ___ b. Repeat the `amqsputc` test from the command window.

After a delay, the program returns an error.

```
C:\Users\Administrator>amqsputc DEPOSIT1
Sample AMOSPUT0 start
MQCONN ended with reason code 2393
```

- ___ 4. Look at the error messages for SECQM1.
 - ___ a. In Notepad, open the log file:
`C:\ProgramData\IBM\mq\qmgrs\SECQM1\errors\AMQERR01.LOG`
 - ___ b. Go to the end of the file and find the **AMQ9636E** error message, and read the EXPLANATION and ACTION sections.
- `AMQ9636E: SSL distinguished name does not match peer name, channel 'CLIENT.TLS'.`

The client application sees that the connection to the server was unsuccessful.

The reason for the failure is not passed back to the client. The AMQ9636 message is in the server queue manager's error log only.

4.6. Exercise cleanup

- 1. Close all open command windows.
- 2. Close the IBM Key Management application.
- 3. Close Notepad.
- 4. Stop the SECQM1 and SECQM2 queue managers.

End of exercise

Exercise review and wrap-up

The first part of the exercise showed you how to use IBM Key Management to create a certificate request. You also learned how to secure channels by using TLS.

Exercise 9. Implementing connection authentication

Estimated time

01:30

Overview

In this exercise, you modify an IBM MQ network to add connection authentication security.

Objectives

After completing this exercise, you should be able to:

- Check locally bound connections
- Check client connections
- Configure the authentication failure delay

Introduction

In this exercise, you enable connection authentication security on a queue manager to check the user ID and password for local connections and client connections.

You create an authentication information object that is named AUTHUSER and configure the queue manager to use this authentication information object. You then modify the **Check locally bound connection** property to see how this property affects connection authentication. You also use the client connection option to authenticate client connections.

This exercise includes these sections:

- [Section 1, "Setting up permissions"](#)
- [Section 2, "Checking locally bound connections"](#)

Requirements

Make sure that you complete [Exercise 7, "Controlling access to IBM MQ"](#) before starting this exercise. This exercises uses the same Windows user definitions.

Section 1. Setting up permissions

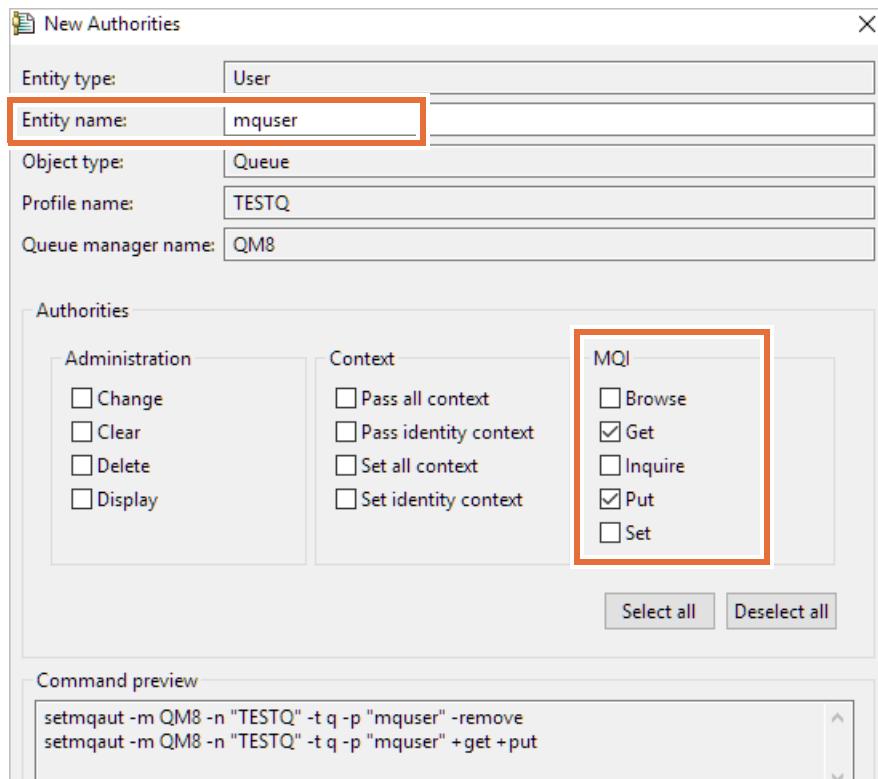
In this part of the exercise, you set up the queue manager and create some local queues for testing purposes. This exercise requires a non-administrative user that is named `mquser`.

1.1. Set up the environment

- ___ 1. In IBM MQ Explorer, create the following MQ objects.

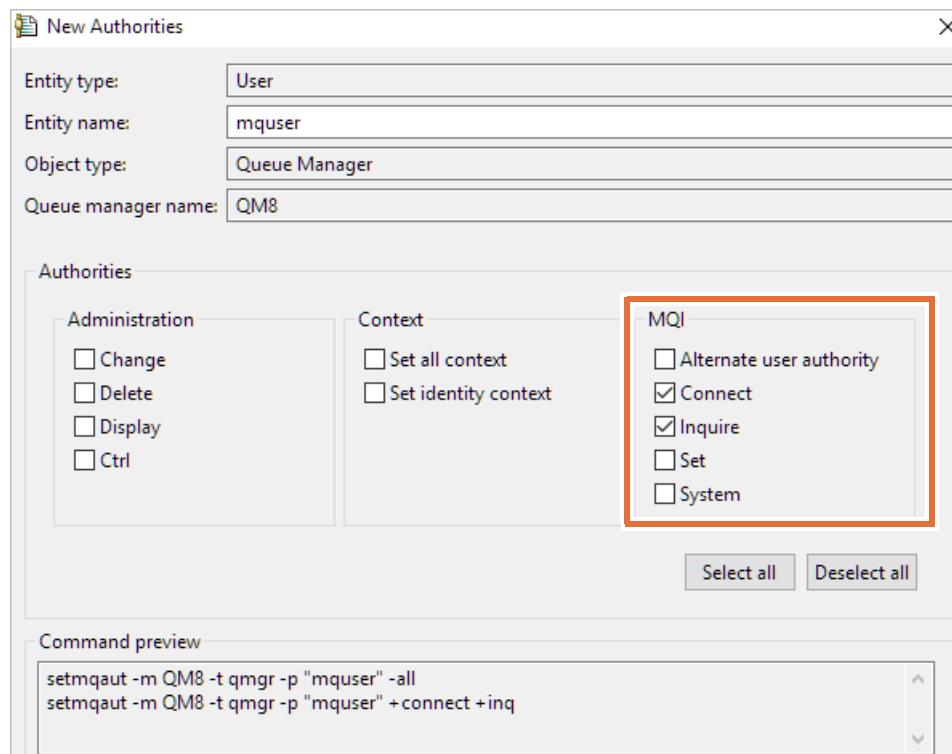
Queue manager	<code>QM8</code> on port 5555
Local queue	<code>TESTQ</code>

- ___ 2. Authorize `mquser` to have PUT and GET permissions on TESTQ.
 - ___ a. In the **QM8 > Queues** content view, right-click **TESTQ** and click **Object Authorities > Manage Authority Records**.
 - ___ b. Under **Specific Profiles**, click **TESTQ**.
 - ___ c. Click the **Users** tab and click **New**.
 - ___ d. For the **Entity name**, type: `mquser`
 - ___ e. In the **MQI** section, select the **Get** and **Put** options and click **OK**.



- ___ f. Click **OK** to close the success message.
- ___ g. Click **Refresh** to ensure that changes are applied
- ___ h. Click **close**.

- ___ 3. Authorize **mquser** to have connection and inquire permissions on the QM8 queue manager.
 - ___ a. In the **Navigator** pane, right-click **QM8** and click **Object Authorities > Manage Queue Manager Authority Records**.
 - ___ b. On the **Users** tab, click **New**.
 - ___ c. For the **Entity name**, type: **mquser**
 - ___ d. In the **MCI** section, select the **Connect** and **Inquire** options, and click **OK**.

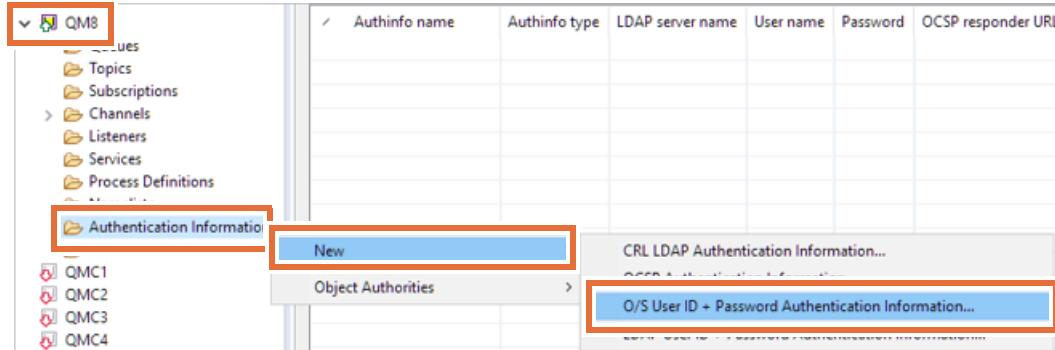


- ___ e. Click **OK** to close the success message.
 - ___ f. Click **Refresh** to ensure that changes are applied to the queue manager, and click **Close**.
- ___ 4. Create a server-connection channel that is named **MYSVRCONN**.
 - ___ a. Right-click the **QM8 > Channels** folder and click **New > Server-connection Channel**.
 - ___ b. Set **Name** to **MYSVRCONN** and click **Finish**.
 - ___ c. Click **OK** to close the success message.

1.2. Defining user ID and password connection authentication

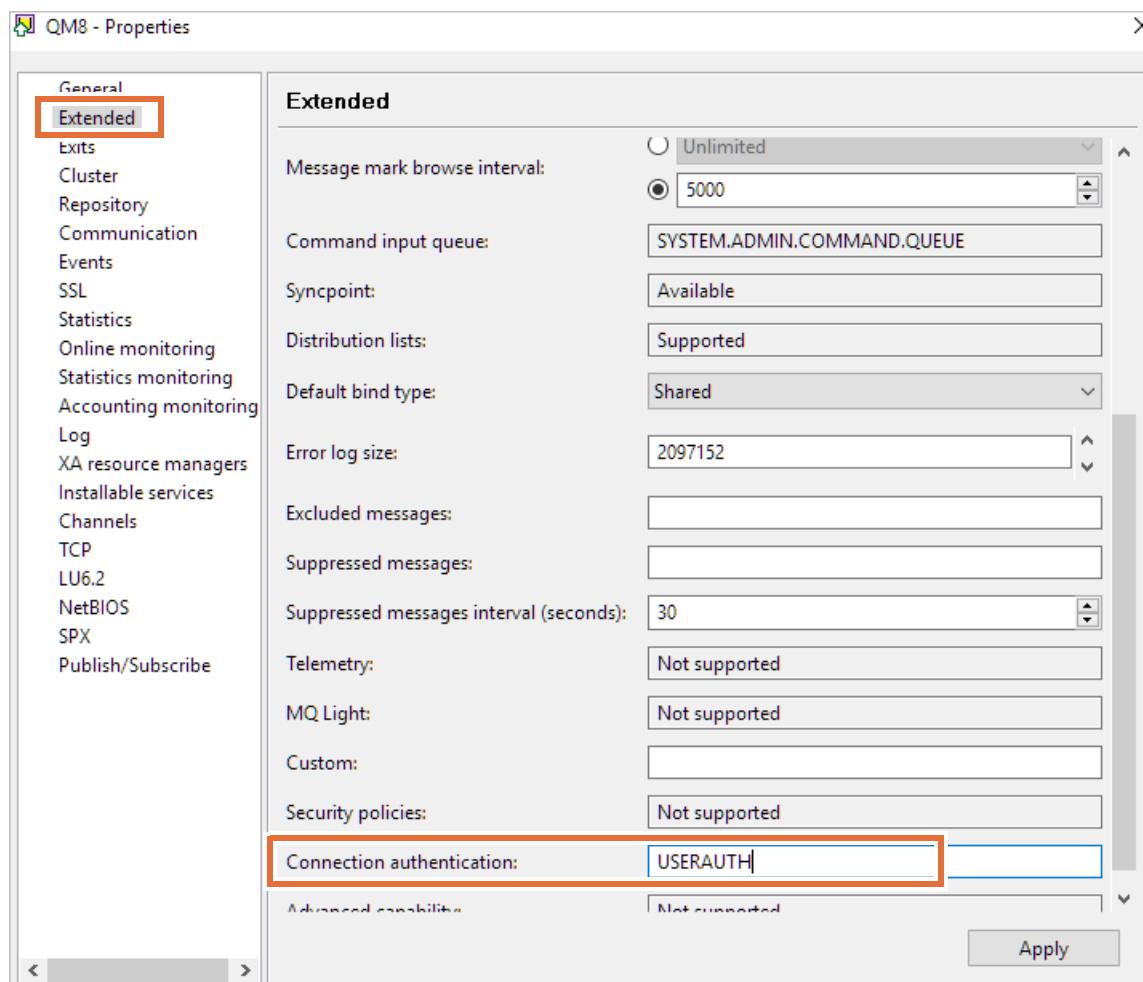
The **Authentication information** (AUTHINFO) object and the **Connection authentication** (CONNAUTH) property of the queue manager work together to provide connection authentication. In this part of the exercise, you create the AUTHINFO object.

- 1. In MQ Explorer, create an AUTHINFO object that is named **USERAUTH** to store the security configuration data.
 - a. In the **Navigator** pane, right-click the **QM8 > Authentication Information** folder and click **New > O/S User ID + Password Authentication Information**.



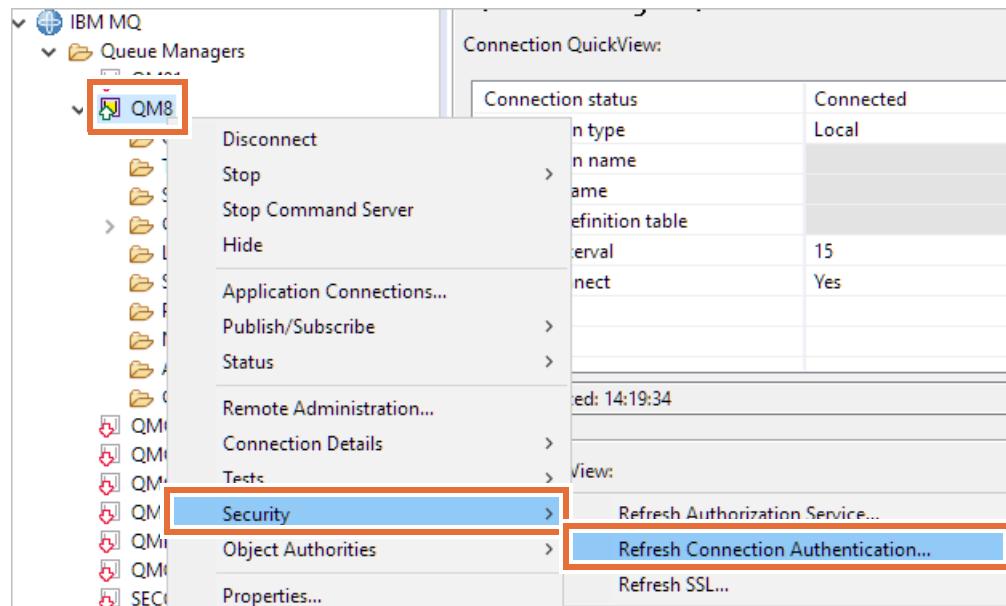
- b. Set **Name** to **USERAUTH** and click **Finish**.
The AUTHINFO object name is case-sensitive.
- c. Click **OK** to close the success message.
- 2. Modify the queue manager **Connection Authentication** property to use the **USERAUTH** object.
 - a. In the **Navigator** pane, right-click **QM8** and click **Properties**.

- ___ b. In the left pane, click **Extended**, and (near the bottom of the page) change the value of the **Connection authentication** property to: USERAUTH



- ___ c. Click **OK**.

- 3. To refresh security on the queue manager, right-click **QM8** and click **Security > Refresh Connection Authentication**.



- 4. When prompted to refresh, click **Yes**.

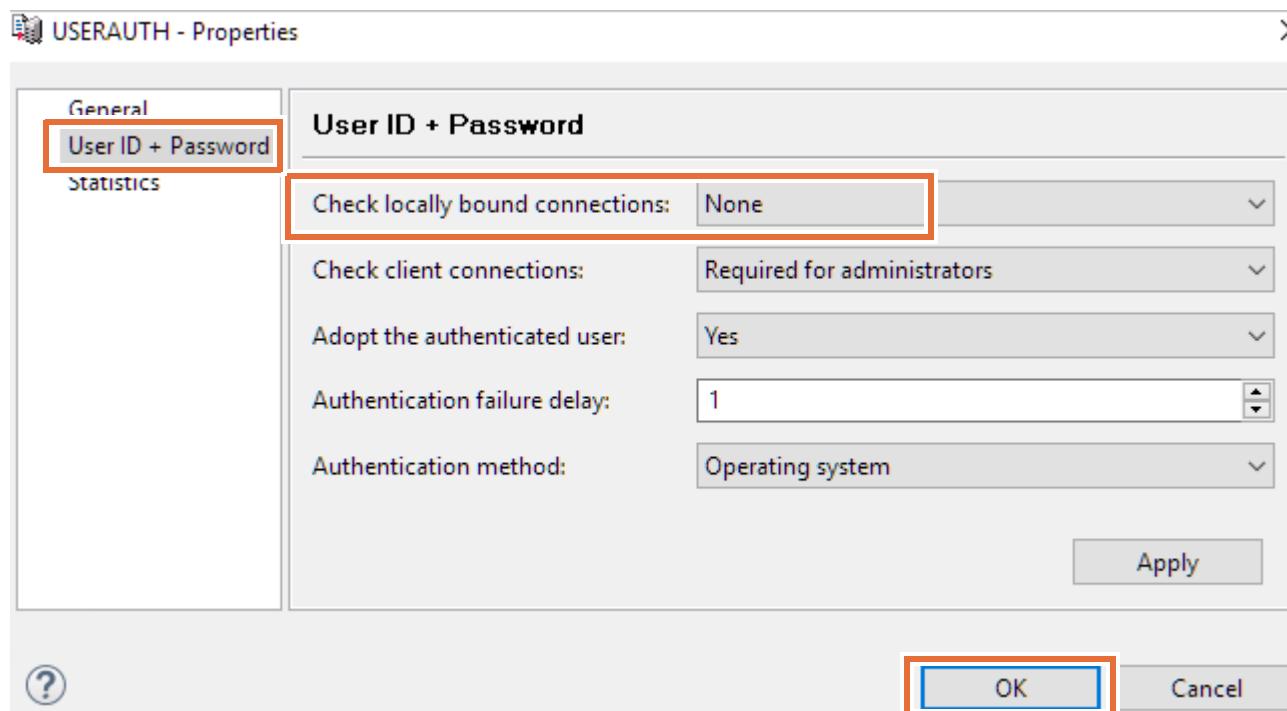
Section 2. Checking locally bound connections

In this part of the exercise, you modify the **Check locally bound connection** property to see how this property affects connection authentication. This part of the exercise focuses on IBM MQ local bindings.

You use the IBM MQ sample programs `amqspput` and `amqsget` to test the connection authentication. The `MQSAMP_USER_ID` environment variable provides the user identification and password for the sample programs.

2.1. Testing with the Check locally bound connections option set to None

- 1. In MQ Explorer, set the **Check locally bound connections** option to **None** on the **USERAUTH** object.
 - a. In the **Navigator** pane, click the **QM8 > Authentication Information** folder.
 - b. Double-click **USERAUTH**.
 - c. Click **User ID + Password** and set **Check locally bound connections** to **None**.
 - d. Click **OK**.



- 2. To refresh security on the queue manager, right-click **QM8**, click **Security > Refresh Connection Authentication** and when prompted to confirm, click **Yes**.
- 3. Use the IBM MQ `amqspput` and `amqsget` sample programs to test the authorization for queue TESTQ on QM8.
 - a. Open two command prompt windows side by side.
 - b. In the first command window, run the `amqspput` sample program:
`amqspput TESTQ QM8`

- ___ c. Type some messages, and then press Enter twice to close the sample.
- ___ d. In a second command window, type the following command to get the messages from the queue TESTQ on QM8:

```
amqsget TESTQ QM8
```

This program stops when no more messages are left on the queue.

2.2. Testing with the Check locally bound connections option set to Optional

- ___ 1. Return to MQ Explorer to set the **Check locally bound connection** property to **Optional** on the USERAUTH object.
 - ___ a. Click the **QM8 > Authentication Information** folder and double-click **USERAUTH**.
 - ___ b. On the **User ID + Password** page, set **Check locally bound connections** to **Optional** and click **OK**.
 - ___ c. Refresh security on the queue manager (right-click **QM8** and click **Security > Refresh Connection Authentication**).
- ___ 2. In the first command prompt (where you ran `amqsput`), set the `MQSAMP_USER_ID` environment variable.


```
set MQSAMP_USER_ID=Administrator
```
- ___ 3. Put some messages on TESTQ.
 - ___ a. In the first command window, run the `amqsput` sample program.


```
amqsput TESTQ QM8
```
 - ___ b. When prompted for the password, type: `passw0rd`
 - ___ c. Enter some messages and press Enter twice to end the program.
- ___ 4. Rerun the `amqsput` sample program, but this time, use an *invalid* password.

You get an authorization failure: `MQCONN` ended with reason code 2035
- ___ 5. In the second command window, run the `amqsget` sample program.


```
amqsget TESTQ QM8
```

You should see that no password is required in this command window. **Check locally bound connection** is set to **Optional** and you did not set the `MQSAMP_USER_ID` environment variable in this command window.

2.3. Testing with the Check locally bound connections option set to Required

- ___ 1. Return to MQ Explorer to set the **Check locally bound connection** property to **Required for all** on the USERAUTH object.
 - ___ a. Click the **QM8 > Authentication Information** folder and double-click **USERAUTH**.

- ___ b. On the **User ID + Password** page, set **Check locally bound connections to Required for all** and click **OK**.
- ___ c. Refresh security on the queue manager (right-click **QM8** and click **Security > Refresh Connection Authentication**).
- ___ 2. Put some messages on TESTQ to test required authorization.
 - ___ a. Return to the first command prompt where you set `MQSAMP_USER_ID=Administrator` and run the `amqspput` sample.


```
amqspput TESTQ QM8
```
 - ___ b. When prompted for the password, type: `passw0rd`
 - ___ c. Enter some messages and press Enter twice to end the program.

The sample program should successfully connect to the queue manager and write messages.

 - ___ d. In the second command window, run the `amqsget` sample.

```
amqsget TESTQ QM8
```

This time, you get a connection error because authorization is required to access the messages.

 - ___ e. In the same command prompt, set the `MQSAMP_USER_ID` environment variable to **Administrator**.

```
set MQSAMP_USER_ID=Administrator
```

 - ___ f. Rerun the `amqsget` sample and when prompted for the password, type: `passw0rd`

The sample program should successfully connect to the queue manager and get the messages.
- ___ 3. Test authorization for **mquser**.
 - ___ a. In the first command prompt, set the `MQSAMP_USER_ID` environment variable to **mquser**:

```
set MQSAMP_USER_ID= mquser
```

 - ___ b. Run the `amqspput` sample and when prompted for the password, type: `passw0rd`.

```
amqspput TESTQ QM8
```

You should see that the applications can connect to the queue manager by using a different user ID and password. The user ID must have the authority read and write on the queue.



Information

Setting CHCKLOCL to REQUIRED or REQDADM means that you cannot locally administer the queue manager with MQSC unless the user specifies the `-u UserId` parameter on the `runmqsc` command line. With the user ID set, `runmqsc` prompts for the user's password at the console.

Similarly, a user running MQ Explorer on the local system receives error AMQ4036 when attempting to connect to the queue manager. To specify a username and password in IBM MQ

Explorer, right-click the local queue manager object and then click **Connection Details > Properties**. In the **Userid** section, enter the username and password.

- ___ 4. Test connection authorization by using MQSC.

When the **Check locally bound connection** property is set to **Required for all**, the queue manager prohibits any access without authentication. Run MQSC as the Administrator to change the **Check locally bound connection** property back to **Optional**.

- ___ a. In one of the command prompts, start MQSC as Administrator (which is a valid user ID).
`runmqsc -u Administrator QM8`
- ___ b. When prompted for the password, enter: `passw0rd`
- ___ c. Reset the **Check locally bound connection** property back to **Optional** and refresh the security.
`ALTER AUTHINFO ('USERAUTH') AUTHTYPE (IDPWOS) CHCKLOCL (OPTIONAL)
REFRESH SECURITY
end`

- ___ 5. Open a new command prompt and run the `amqsput` program again to check that the security is refreshed.

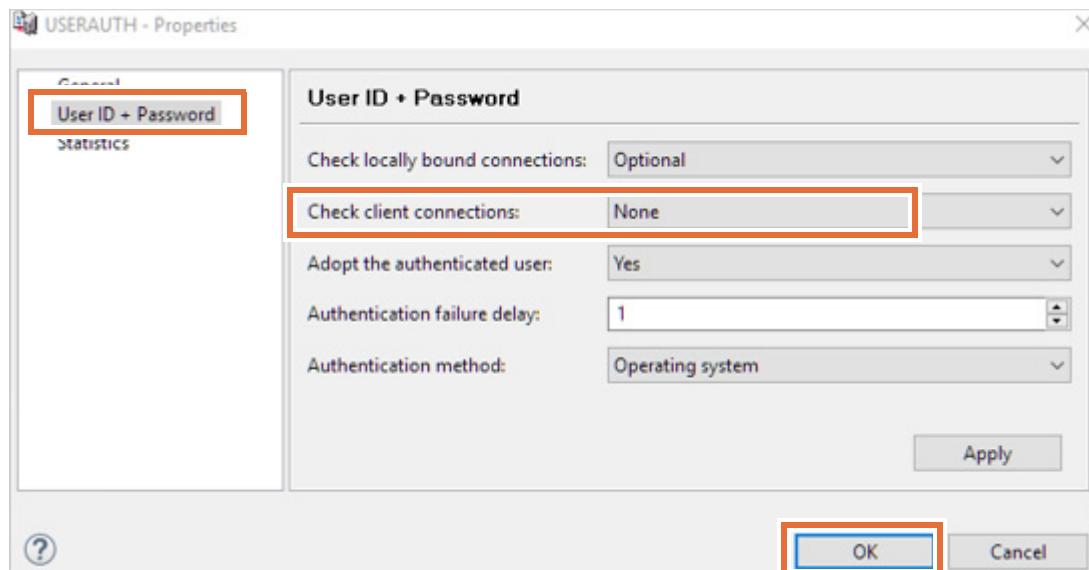
```
amqsput TESTQ QM8
```

- ___ 6. The program connects without requiring a password, so end the program and close all the command prompts.

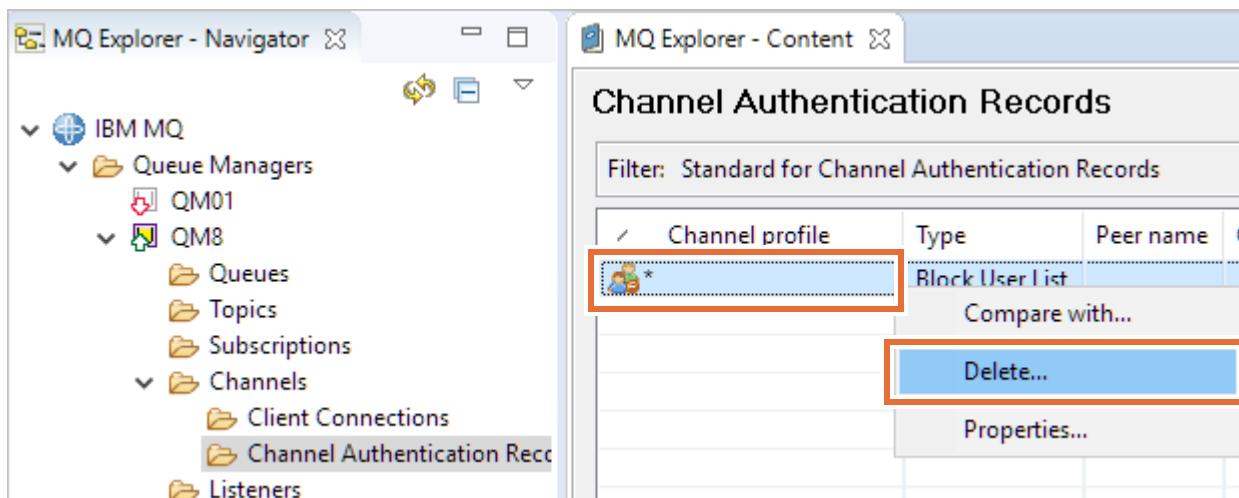
Section 3. Checking client connections (MQ client bindings)

In this part of the exercise, you use the client connection option to authenticate client connections. You use the `amqsputc` and `amqsgetc` sample programs to provide user identification and password.

- 1. In MQ Explorer, set the **Check client connections** property for USERAUTH to **None**.
 - a. Click the **QM8 > Authentication Information** folder and double-click **USERAUTH**.
 - b. On the **User ID + Password** page, set **Check client connections** to **None** and click **OK**.



- c. Refresh security on the queue manager (right-click **QM8** and click **Security > Refresh Connection Authentication**) and when prompted to confirm, click **Yes**.
- 2. For this exercise, disable channel authentication so that it does not block the channel during the next steps.
 - a. Expand **QM8 > Channels** and click the **Channel Authentication Records** folder.
 - b. Right-click the profile field and click **Delete**.



- c. When prompted, click **Delete** again and click **OK** to close the success message.

- ___ 3. Open a new command window and set the MQSERVER environment variable.

```
set MQSERVER=MYSVRCONN/TCP/localhost(5555)
```



Note

You must set the MQSERVER environment variable in any command window that you use to run the client sample programs during this part of the exercise.

- ___ 4. Use the client sample programs `amqsputc` and `amqsgetc` to test the client authentication settings. You should see that connection authentication is not required to write and read the messages from the queue.

- ___ a. In the command prompt, run the `amqsputc` sample program.

```
amqsputc TESTQ
```

- ___ b. Enter some messages and press Enter twice to end the program.

- ___ c. In the same command window, run the `amqsgetc` sample to get the messages:

```
amqsgetc TESTQ
```

- ___ 5. Set the **Check client connections** property for USERAUTH to **Optional**.

- ___ a. In MQ Explorer, click the **QM8 > Authentication Information** folder and double-click **USERAUTH**.

- ___ b. On the **User ID + Password** page, set **Check client connections** to **Optional** and click **OK**.

- ___ c. Refresh security on the queue manager (right-click **QM8** and click **Security > Refresh Connection Authentication**) and when prompted to confirm, click **Yes**.

- ___ 6. Retest the client connection with `amqsputc` and `amqsgetc`.

- ___ a. In a command prompt, set the **MQSAMPU_USER_ID** environment variable to **Administrator**.

```
SET MQSAMPU_USER_ID=Administrator
```

- ___ b. Run `amqsputc` and when prompted for a password, type: `passw0rd`

```
amqsputc TESTQ
```

- ___ c. Enter some messages and press Enter twice to end the program.

- ___ d. In the same command prompt, run `amqsgetc` to get the messages, and when prompted for a password, type: `passw0rd`

```
amqsgetc TESTQ
```

- ___ 7. Set the **Check client connections** property for USERAUTH to **Required for all**.

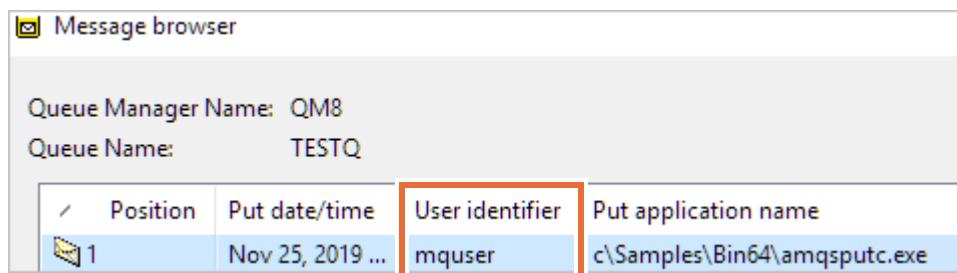
- ___ a. In MQ Explorer, click the **QM8 > Authentication Information** folder and double-click **USERAUTH**.

- ___ b. On the **User ID + Password** page, set **Check client connections** to **Required for all** and click **OK**.

- ___ c. Refresh security on the queue manager (right-click **QM8** and click **Security > Refresh Connection Authentication**) and when prompted to confirm, click **Yes**.
- ___ 8. Change the **MQSAMP_USER_ID** environment variable in the command window to use the "mquser" user ID.

```
SET MQSAMP_USER_ID= mquser
```
- ___ 9. Put messages on TESTQ to test authentication on the client.
 - ___ a. In the command prompt, run the **amqsputc** sample program.
`amqsputc TESTQ`
 - ___ b. Enter some messages and press Enter twice to end the program.

The application uses the Administrator user ID because **Adopt the authentication user** is set to **No** for USERAUTH.
- ___ 10. Return to MQ Explorer to browse the message TESTQ.
 - ___ a. Click the **QM8 > Queues** folder, right-click **TESTQ** and click **Browse Messages**.
 - ___ b. In the Message Browser window, note the user ID that is contained in the message header.



- ___ c. Close the Message browser.

3.1. Exercise cleanup

- ___ 1. Close all command windows.
- ___ 2. Stop the queue manager QM8.

End of exercise

Exercise review and wrap-up

In this exercise, you modified an IBM MQ network to add connection authentication security.

Exercise 10. Running an IBM MQ trace

Estimated time

00:30

Overview

In this exercise, you start a trace on the IBM MQ **amqspput** and **amqsget** sample programs and examine the trace output. You also configure IBM MQ to automatically handle messages that arrive on the dead-letter queue by using the dead-letter queue handler.

Objectives

After completing this exercise, you should be able to:

- Start and stop an IBM MQ trace
- Analyze the output from the IBM MQ trace
- Handle dead-letter messages

Introduction

The trace function in IBM MQ is used for problem determination or as a program development aid.

In this exercise, you start a trace on the sample programs. You use a simple scenario of using the sample programs to put and get messages, and getting the process ID (PID) of the sample programs. You use the PID to locate the trace files.

The exercise includes these sections:

- [Section 1, "Starting a trace"](#)
- [Section 2, "Viewing the trace files"](#)
- [Section 3, "Using the dspmqte application"](#)
- [Section 4, "Working with the dead-letter queue"](#)

Requirements

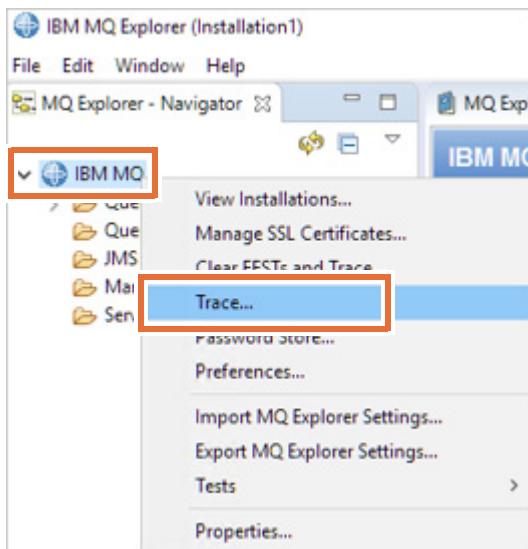
This exercise uses the queue manager and queues that you defined in [Exercise 1, "Getting started with IBM MQ"](#) and [Exercise 2, "Working with IBM MQ administration tools"](#).

Section 1. Starting a trace

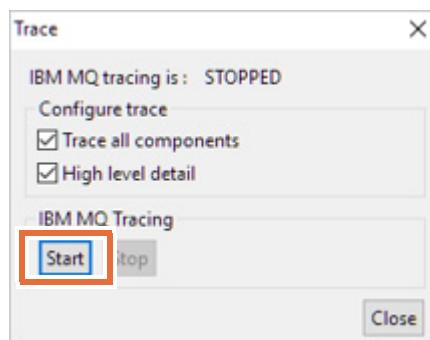
In this exercise, you start a trace on the IBM MQ **amqspput** and **amqsget** sample programs. You then use the sample programs to send and receive messages and examine the trace output.

1.1. Start the trace

- ___ 1. In IBM MQ Explorer, in the **Queue Managers** folder, start the **QM01** queue manager.
 - ___ a. Right-click QM01 and click **Start**.
 - ___ b. Click **OK**.
- ___ 2. Start the trace.
 - ___ a. In the **Navigator** pane, right-click the **IBM MQ** and click **Trace**.



- ___ b. In the Trace dialog box, click **Start**.





Information

When you use MQ Explorer to start a trace, you do not have the option of selecting specific components to trace.

When you use the `strmqtrc` command, you can identify specific components. For example, to start the trace on the `amqspput` and `amqsget` sample programs, you run this command:

```
strmqtrc -m QM01 -p amqspput.exe,amqsget.exe
```

1.2. Use the sample programs to put and get messages from a queue

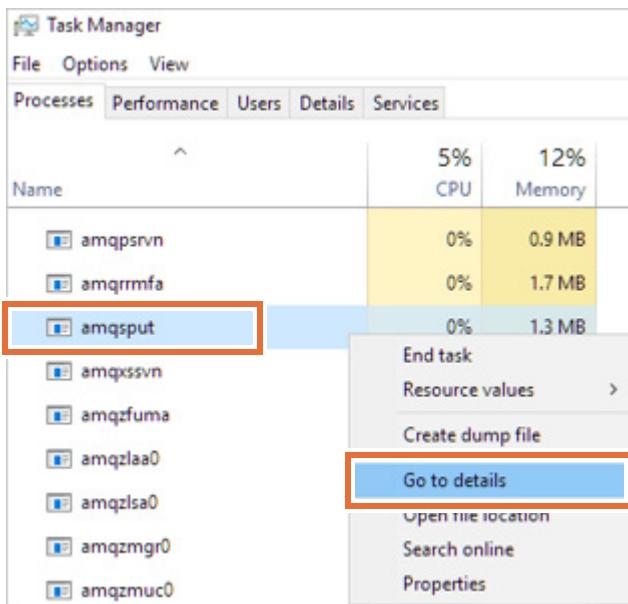
In this part of the exercise, you run the `amqspput` and `amqsget` sample programs to generate trace files. You get the process ID (PID) of the sample programs so that you can identify the trace files for each program.

- ___ 1. Open a command prompt to start the `amqspput` sample program.

```
amqspput QL.A QM01
```

Do not put any messages on the queue yet.

- ___ 2. Get the PID of the `amqspput` sample program by using Windows Task Manager.
 - ___ a. Right-click the Windows taskbar, and select **Task Manager**.
 - ___ b. Click **More details**, and on the **Processes** tab, click **Name** to sort the processes alphabetically by program name.
 - ___ c. Right-click the `amqspput` program and click **Go to details**.



- ___ d. In the **PID** column, note the number and write it here: _____

Name	PID	Status
amqfcxba.exe	4136	Running
amqfqpublisher.exe	3524	Running
amqmtnbrn.exe	364	Running
amqpcsea.exe	4300	Running
amqpsrvn.exe	3336	Running
amqrrmfa.exe	4920	Running
amqspput.exe	4516	Running

- ___ e. Leave the Task Manager running. You repeat these steps to get the PID of the `amqspput` sample program.
- ___ 3. In the command prompt that is running `amqspput`, enter some test messages and then press Enter again to exit the sample `amqspput` program.
- ___ 4. Get the `amqspget` PID.
- NOTE:** The `amqspget` displays messages in the queue, and then waits 15 seconds to see whether more messages arrive on the queue. You need to identify the `amqspget` PID during this 15-second interval.
- ___ a. Make sure that Windows Task Manager is open to the **Details** tab.
- ___ b. Start the `amqspget` sample program.
- ```
amqspget QL.A QM01
```
- \_\_\_ c. Find the PID for `amqspget` in the Windows Task Manager and write it here:
- \_\_\_\_\_
- \_\_\_ 5. Stop the MQ trace.
- \_\_\_ a. In the **MQ Explorer**, right-click the **IBM MQ**, in the **Navigator** pane, and click **Trace**.
- \_\_\_ b. Click **Stop**.



## Information

You can stop the trace with the following MQ command:

```
endmqtrc -m QM01
```

## Section 2. Viewing the trace files

- \_\_\_ 1. Go to the C:\ProgramData\IBM\MQ\trace directory.

A trace file is created for each process. The trace files are readable without formatting and are called **AMQxxxx.0.TRC** where **xxxx** is the process identifier (PID) that created the file.

- \_\_\_ 2. Referencing the PIDs that you noted in the previous section for the `amqspput` and `amqsget` sample programs, locate the trace files and open them in Notepad.

Sample excerpt of a Windows trace file for `amqspput`:

```

Process : C:\Program Files\IBM\MQ\bin64\amqspput.exe (64-bit)
Arguments :
Host : WS2016X64
Operating System : Windows Server 2016 Server Standard Edition, Build 14393
Product Long Name : IBM MQ for Windows (x64 platform)
Version : 9.1.0.1 Level : p910-001-181108
O/S Registered : 1
Data Path : C:\ProgramData\IBM\MQ
Installation Path : C:\Program Files\IBM\MQ
Installation Name : Installation1 (1)
License Type : Production
UTC Date : 2019/11/27: Time : 22:42:12.359
LCL Date : 2019/11/27: Time : 17:42:12.359 Eastern Standard Time
LCL Date : 2019/11/27: Time : 17:42:12.359 Eastern Standard Time

Counter TimeStamp PID.TID Ident Data
=====

086CD21F 17:42:12.358583 6000.1 : !! - Thread stack (from mqe.dll)
086CD220 17:42:12.358610 6000.1 : !! - -> InitProcessInitialisation
086CD221 17:42:12.358619 6000.1 : --{ InitProcessInitialisation

...

```

- \_\_\_ 3. Close the files.

## Section 3. Using the `dspmqrte` application

In this part of the exercise, you create the queue managers, channels, and queues so that you can exchange messages between the queue managers.

### 3.1. Creating the queue manager and connections

- 1. In MQ Explorer, create a queue manager with these properties:
  - **Name:** QM02
  - **Dead-letter queue:** DLQ
  - **Port:** 1415
- 2. In a command prompt, run MQSC on QM02 to create the queues and channels so that QM02 can send and receive messages from QM17.

```
runmqsc QM02
```

```
ALTER QMGR CHLAUTH(DISABLED)
ALTER QMGR CONNAUTH(' ')
REFRESH SECURITY(*)
DEF QL(QL.A) REPLACE
DEF QL(DLQ) REPLACE

DEF QL(QM17) USAGE(XMITQ)
DEF CHL(QM02.QM17) CHLTYPE(SDR) REPLACE +
TRPTYPE(TCP) CONNAME('localhost(1415)') +
XMITQ(QM17)

DEF CHL(QM17.QM02) CHLTYPE(RCVR) REPLACE +
TRPTYPE(TCP)
end
```

You receive a message that 8 MQSC commands were read with no syntax error.

The result is the creation of the following objects on the QM02 queue manager:

- A local queue: QL.A
- A dead-letter queue: DLQ
- A transmission queue: QM17
- A sender channel: QM02.QM17 that points to QM17 and uses the transmission queue QM17
- A receiver channel: QM17.QM02

- 1. In MQ Explorer, create a queue manager with these properties:
  - **Name:** QM17
  - **Dead-letter queue:** DLQ
  - **Port:** 1417

- \_\_ 2. Run MQSC on QM17 to create the IBM MQ objects that this exercise requires.

```
runmqsc QM17
```

```
ALTER QMGR CHLAUTH(DISABLED)
ALTER QMGR CONNAUTH(' ')
REFRESH SECURITY(*)
DEF QL(QM02) REPLACE USAGE(XMITQ)
DEF QL(DLQ) REPLACE
DEF CHL(QM17.QM02) CHLTYPE(SDR) REPLACE +
TRPTYPE(TCP) CONNAME('localhost(1415)') +
XMITQ(QM02)
DEF CHL(QM02.QM17) CHLTYPE(RCVR) REPLACE +
TRPTYPE(TCP)
DEF QR(QRMT1) REPLACE +
RNAME(QL.A) RQMNAME(QM02) +
XMITQ(QM02)
end
```

You receive a message that 8 MQSC commands were read with no syntax error.

The result is the creation of the following objects on the QM17 queue manager:

- A dead-letter queue: DLQ
- A transmission queue: QM02
- A sender channel: QM17.QM02
- A receiver channel: QM02.QM17
- A remote queue definition: QRMT1 that points to QL.A on QM02 and uses the transmission queue that is named QM02

### 3.2. Test the channel definitions

Ensure that the sender (SDR) channels are started on QM17 and QM02, and that the listeners are running.

- \_\_ 1. To test the channel definitions, use the following MQSC commands on QM17 to ping the message channel from the QM17 (the sender):

```
runmqsc QM17
PING CHL(QM17.QM02)
```

You should receive a message: Ping IBM MQ channel complete

- \_\_ 2. After completion of the ping command, start the sender channel.

```
START CHL(QM17.QM02)
end
```

You should receive a message: Start IBM MQ channel accepted

- \_\_\_ 3. Use the sample program `amqspput` to test the connection by sending a message from the queue manager QM17 to the queue QL.A on the queue manager QM02.
  - \_\_\_ a. Run the `amqspput` program.  
`amqspput QRMT1 QM17`  
**Note:** QRMT is the remote queue definition that points to QL.A on QM02.
  - \_\_\_ b. Send a message, such as:  
`Testing transmission to QL.A`
  - \_\_\_ c. Press Enter twice to end the program.
- \_\_\_ 4. Use IBM MQ Explorer to verify that the message was put on QL.A on QM02.

### 3.3. Running `dspmqrte`

In this part of the exercise, QM17 is the local queue manager and QM02 is the remote queue manager. The display route application sends a message to and from a queue manager.

- \_\_\_ 1. In MQ Explorer, in the **Navigator** pane, click **Channels** folder under the **QM02** and **QM17** queue manager folders to check that the sender channels are started on QM17 and QM02.
- \_\_\_ 2. Click the **Listeners** folder under the **QM02** and **QM17** queue manager folders to verify that **Listener status** is Running.



#### Troubleshooting

If the channel status is **Retrying**, try stopping the channel and restarting it. If the QM17.QM02 sender channel on QM17 is running, you can continue with the next steps.

- \_\_\_ 3. Observe the route of the normal delivery of a message by sending a message from QM17 to QL.A on QM02.
  - \_\_\_ a. In a command prompt run `amqspput` to put messages through QRMT1 to the remote queue QL.A.  
`amqspput QRMT1 QM17`
  - \_\_\_ b. Enter a message such as `Testing dspmqrte` and press Enter twice to end the program.
- \_\_\_ 4. In the command prompt, run the display route command and examine the response.  
`dspmqrte -m QM17 -q QRMT1 -v outline -w 10`

**Note:** Use the `-w` parameter to specify how long the program should wait before finishing. In this case, `-w 10` means a 10-second timeout wait.

Your results should be similar to this example.

```
C:\Users\Administrator>dspmqrte -m QM17 -q QRMT1 -v outline -w 10
AMQ8653I: DSPMQRTE command started with options '-m QM17 -q QRMT1 -v outline -w 10'.
AMQ8659I: DSPMQRTE command successfully put a message on queue 'QM02', queue manager 'QM17'.
AMQ8674I: DSPMQRTE command is now waiting for information to display.
```

---

```
Activity:
ApplName: 'es\IBM\MQ\bin64\dspmqrte.exe'
```

```
Operation:
OperationType: Put
QMgrName: 'QM17'
QName: 'QRMT1'
ResolvedQName: 'QM02'
RemoteQName: 'QL.A
RemoteQMgrName: 'QM02'
```

---

```
Activity:
ApplName: 'es\IBM\MQ\bin64\runmqch1.exe'
```

```
Operation:
OperationType: Get
QMgrName: 'QM17'
QName: 'QM02'
ResolvedQName: 'QM02'
```

```
Operation:
OperationType: Send
QMgrName: 'QM17'
RemoteQMgrName: 'QM02'
ChannelName: 'QM17.QM02'
ChannelType: Sender
XmitQName: 'QM02'
```

---

```
AMQ8652I: DSPMQRTE command has finished.
```

- \_\_\_ 5. Examine each activity of the trace route output and the process names that are associated with each activity.
- \_\_\_ 6. Observe the trace route output when a channel is stopped.
  - \_\_\_ a. In IBM MQ Explorer, stop the sender channel **QM17.QM02** on QM17.
  - \_\_\_ b. Run the display route command again.

```
dspmqrte -m QM17 -q QRMT1 -v outline -w 10
```



## Questions

What do you observe as the final activity?

```
C:\Users\Administrator>dspmqrte -m QM17 -q QRMT1 -v outline
AMQ8653I: DSPMQRTE command started with options '-m QM17 -q QRMT1 -v outline'.
AMQ8659I: DSPMQRTE command successfully put a message on queue 'QM02', queue manager 'QM17'.
AMQ8674I: DSPMQRTE command is now waiting for information to display.
```

---

```
Activity:
ApplName: 'es\IBM\MQ\bin64\dspmqrte.exe'
```

```
Operation:
OperationType: Put
QMgrName: 'QM17'
QName: 'QRMT1'
ResolvedQName: 'QM02'
RemoteQName: 'QL.A'
RemoteQMGRName: 'QM02'
```

---

This result is incomplete. You conclude that the message is being held up.

The message is on the transmission queue QM02. You should conclude that the channel is not processing messages for some reason.

- 
- 7. Restart the sender channel **QM17.QM02** on QM17.

## Section 4. Working with the dead-letter queue

### 4.1. Viewing messages on the dead-letter queue

- \_\_\_ 1. On QM02, modify the remote queue by inhibiting PUT on QL.A.
  - \_\_\_ a. In MQ Explorer, in the **QM02 > Queues** view, double-click **QL.A** and set **Put messages to Inhibited**.
  - \_\_\_ b. Use `amqspput` to try putting a message on the QL.A queue of QM02 from the queue manager QM17.

```
amqspput QRMT1 QM17
```

- \_\_\_ c. After you type a message, press Enter twice to end the program.

You should see that message is put on **DLQ** (the dead-letter queue) of QM02.

- \_\_\_ 2. Modify QL.A on QM02 to allow PUT.
- \_\_\_ 3. Examine the dead-letter queue (DLQ) header by using the browse queue sample program `amqsbcg` and determine the reason code.
  - \_\_\_ a. In the command prompt, run `amqsbcg`.

```
amqsbcg DLQ QM02
```

- \_\_\_ b. Locate the 4-byte reason code in the dead-letter header (bytes 9-12) in the **Message** section.

```
**** Message ****
```

```
length - 183 of 183 bytes
```

```
00000000: 444C 4820 0100 0000 0308 0000 514C 2E41 'DLHQL.A'
00000010: 2020 2020 2020 2020 2020 2020 2020 '
00000020: 2020 2020 2020 2020 2020 2020 2020 '
00000030: 2020 2020 2020 2020 2020 514D 3032 ' QM02'
00000040: 2020 2020 2020 2020 2020 2020 2020 '
00000050: 2020 2020 2020 2020 2020 2020 2020 '
00000060: 2020 2020 2020 2020 2202 0000 ' ...
00000070: B501 0000 4D51 5354 5220 2020 0B00 0000 '....MQSTR'
00000080: 6573 5C49 424D 5C4D 515C 6269 6E36 345C 'es\IBM\MQ\bin64\
00000090: 616D 7172 6D70 7061 2E65 7865 3230 3139 'amqrmpaa.exe2019'
000000A0: 3131 3238 3031 3337 3035 3339 7465 7374 '112801370539test'
000000B0: 696E 6720 646C 71 'ing dlq
```

- \_\_\_ c. Run the `mqrc` command to find the meaning of this reason code.

```
mqrc 0x00000803
```

This command should return the reason of `MQRC_PUT_INHIBITED`.

In the results of the browse queue sample program `amqsbcg`, the 4-byte reason code is represented as `0308 0000`, which is interpreted as: `0000 0803`

- \_\_\_ 4. Use IBM MQ Explorer to find the reason code.
  - \_\_\_ a. In the **QM02 > Queues** view, right-click **DLQ** (the dead-letter queue) and click **Browse messages**.

- \_\_\_ b. Double-click the message to open **Properties**.
- \_\_\_ c. In the left pane, click **Dead-letter header**.

You should see that the **Reason** field is set to **MQRC\_PUT\_INHIBITED**.

| Dead-letter header         |                              |
|----------------------------|------------------------------|
| Reason:                    | MQRC_PUT_INHIBITED           |
| Destination queue:         | QL.A                         |
| Destination queue manager: | QM1                          |
| Original encoding:         | 546                          |
| Original CCSID:            | 437                          |
| Original format:           | MQSTR                        |
| Put application type:      | Windows                      |
| Put application name:      | es\IBM\MQ\bin64\amqrmppa.exe |
| Put date:                  | 2019/09/19                   |
| Put time:                  | 01:13:24                     |

- \_\_\_ d. Close the properties window and the message browser.

## 4.2. Using the dead-letter queue handler

In this part of the exercise, you use the IBM MQ dead-letter queue handler program to process messages on the dead-letter queue.

- \_\_\_ 1. In MQ Explorer, create a local queue on QM02 that is named QL.B.  
This queue is used to store messages from the dead-letter queue DLQ.
- \_\_\_ 2. Create a dead-letter queue handler rules table file that forwards the dead-letter message that is destined for QL.A on QM02 to QL.B on QM02 but strips the DLQ header.

- \_\_\_ a. In Notepad, type the following commands, and after the last line, press Enter twice.

```
inputqm(QM02) inputq(DLQ)
reason(MQRC_PUT_INHIBITED) action(fwd) fwdq(QL.B) +
fwdqm(QM02) header(no)
```



### Important

You must add a blank line after the last line in the rules table file.

- 
- \_\_\_ b. Save the file as **DLQrules.txt** in the **C:\labfiles** directory.
  - \_\_\_ 3. In a command prompt, start the dead-letter queue handler.

```
runmqdlq DLQ QM02 < C:\labfiles\DLQrules.txt
```

You should see the following message:

```
Dead-letter queue handler started to process INPUTQ(DLQ)
```

- \_\_\_ 4. In MQ Explorer, confirm that the dead-letter queue handler moved the message from DLQ to QL.B.
- \_\_\_ 5. In MQ Explorer, browse the message on QL.B (right-click the queue and click **Browse messages**).

The dead-letter header is removed from the message, and you see the message that you sent in [Section 4.1, "Viewing messages on the dead-letter queue"](#).

### 4.3. Exercise cleanup

- \_\_\_ 1. Stop queue managers QM01, QM02, and QM17.
- \_\_\_ 2. Close the trace files.
- \_\_\_ 3. Close Windows Task Manager.
- \_\_\_ 4. Close the command prompts.

### End of exercise

## Exercise review and wrap-up

This exercise demonstrated how to start a trace on the IBM MQ sample programs and examine the trace output.

---

# Exercise 11. 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
- Recreate 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.

This exercise includes these sections:

- [Section 1, "Restarting the queue manager"](#)
- [Section 2, "Recovering media"](#)

## Requirements

Make sure that you complete [Exercise 10, "Running an IBM MQ trace"](#) before starting this exercise.

## Section 1. Restarting the queue manager

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.

### 1.1. Create a queue manager, local queues, and alias queues

- \_\_\_ 1. In MQ Explorer, make sure that no queue managers are running to avoid any conflicts or problems during this exercise. If any queue managers are running, stop them.
- \_\_\_ 2. In a command prompt, 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.  

```
crtmqm -l1 -p 1417 -u SYSTEM.DEAD.LETTER.QUEUE QML01
```
- \_\_\_ 3. Start the queue manager.  

```
strmqm QML01
```
- \_\_\_ 4. Create an MQSC command file, named `recover.mqsc`, to define two local queues and two alias queues.
  - \_\_\_ a. Open a text editor, such as Notepad.
  - \_\_\_ b. Type the following lines to define two local queues, named QL.A and QL.B, with the default persistence set to YES.
 

```
DEF QL(QL.A) DEFPSIST(YES) REPLACE
DEF QL(QL.B) DEFPSIST(YES) REPLACE
```
  - \_\_\_ c. Append the following line to define an alias queue, named QA.A, that resolves to the local queue QL.A and with default persistence set to NO.
 

```
DEF QA(QA.A) TARGET(QL.A) DEFPSIST(NO) REPLACE
```
  - \_\_\_ d. Append the following line to define an alias queue, named QA.B, that resolves to the local queue QL.B and with default persistence set to NO.
 

```
DEF QA(QA.B) TARGET(QL.B) DEFPSIST(NO) REPLACE
```
  - \_\_\_ e. Save the file as `recover.mqsc` in the `C:\labfiles\Lab11-recover` directory, and close it.
- \_\_\_ 5. Run the MQSC file to process the command file and redirect the output to a report file.
  - \_\_\_ a. In the command prompt, switch to the `C:\labfiles\Lab11-recover` directory.  
`cd C:\labfiles\Lab11-recover`
  - \_\_\_ b. Run the MQSC file.  
`runmqsc QML01 < recover.mqsc > recoverreport.txt`
- \_\_\_ 6. In Windows Explorer, go to the `Lab11-recover` directory to review the `recoverreport.txt` file and verify that all commands ran successfully, and close the file when you are done.

## 1.2. Test the queues

- 1. Run the `amqsput` sample program on the local and alias queues and enter some messages on each.

- For QL.A, type: `amqsput QL.A QML01` and enter some messages
  - For QA.A, type: `amqsput QA.A QML01` and enter some messages
  - For QL.B, type: `amqsput QL.B QML01` and enter some messages
  - For QA.B, type: `amqsput QA.B QML01` and enter some messages

The value of the **DefPersistence** attribute of the queue determines whether the message that is put on the queue is persistent or non-persistent. The `amqsput` sample program puts persistent messages on the local queues and non-persistent messages on the alias queues.

- \_\_ 2. Use the MQ Explorer or the sample program `amqsbcl` 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.

- For QL.A, type: amqsbqcg QL.A QML01
  - For QA.A, type: amqsbqcg QA.A QML01
  - For QL.B, type: amqsbqcg QL.B QML01
  - For QA.B, type: amqsbqcg QA.B 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.

**Note:** You can also use MQ Explorer to browse the messages.

- ### 3. Stop the queue manager QML01.

```
endmam -i OML01
```

- #### 4. Restart the queue manager QML01.

strmam OML01

5. Use `amqscfg` to browse the messages on QL.A and QL.B again, and verify that only the persistent messages on the local queue survived the restart.

- For QL.A, type: amqsbqcg QL.A QML01
  - For QL.B, type: amqsbqcg QL.B QML01

## Section 2. Recovering media

### 2.1. Damage your queue

- 1. Write an image of the queue to the log for use in media recovery by running the `rcdmqimg` command against the queue QL.B.

```
rcdmqimg -m QML01 -t ql QL.B
```

- 2. Damage the queue by running the following command to delete the file that implements local queue QL.B:

```
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.

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

This command still works because it does not yet recognize that the queue is damaged.

- 4. Run `amqsput QL.B QML01` and put 10 - 15 messages on the queue until the program ends with an error code.

Because of buffering, you might need to put 10 or more messages before the queue manager detects that the queue is damaged and reports the damage by returning MQRC 2101, which means MQRC\_OBJECT\_DAMAGED.

- 5. Repeat [Step 3](#) to display the attributes of QL.B in MQSC mode.

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

### 2.2. Re-create your queue

- 1. Use the re-create object (`rcrmqobj`) command to recover local queue QL.B from its media image.

```
rcrmqobj -m QML01 -t ql QL.B
```

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

- 2. Repeat [Step 3](#) to confirm that you can display the attributes of the queue.

- 3. Run `amqsbcg QL.B QML01` to browse the local queue QL.B to check whether the messages are recovered successfully.



#### Questions

Were the non-persistent messages recovered?

## 2.3. Exercise cleanup

- \_\_\_ 1. Clear the messages from QL.A.  
amqsget QL.A QML01
- \_\_\_ 2. Clear the messages from QL.B.  
amqsget QL.B QML01
- \_\_\_ 3. Stop the queue manager QML01.  
endmqm -i QML01
- \_\_\_ 4. Close all command prompts.

**End of exercise**

## Exercise review and wrap-up

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

# Exercise 12. 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:

- Back up object definitions of a queue manager
- 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

Make sure that you completed [Exercise 11, "Using a media image to restore a queue"](#) before starting this exercise.

## Section 1. Backing up and restoring queue manager configuration

### 1.1. Create a backup

- \_\_\_ 1. Open a command prompt and start QM01.  

```
strmqm QM01
```
- \_\_\_ 2. Run the IBM MQ `dmpmqcfg` command to back up QM01.
  - \_\_\_ a. In the command prompt, switch to the `C:\labfiles\Lab12-restore` directory.  
`cd C:\labfiles\Lab12-restore`
  - \_\_\_ b. Use the default formatting option of MQSC and all attributes, and use standard output redirection to store the definitions into a file.  
`dmpmqcfg -m QM01 -a > QM01.mqsc`
- \_\_\_ 3. In Windows Explorer, go to the `C:\labfiles\Lab12-restore` directory to open the `QM01.mqsc` file in Notepad and answer the following questions.



#### Questions

Verify that `SYSTEM.*` objects appear in the backup file.

- Can you suppress generation of the `SYSTEM.*` objects?
- Is the `REPLACE` option present in the generated definitions?
- What does `REPLACE` do?

- 
- \_\_\_ 4. Close Notepad.

### 1.2. 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 IBM MQ objects. You do not replace any definitions because some of the `SYSTEM` queues are not empty. Redefining the `SYSTEM` queue objects 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, and take note of the number of messages that are reported on the queues.  

```
runmqsc QM01
DIS QL(SYSTEM.*) WHERE (CURDEPTH GT 0)
end
```
- \_\_\_ 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. In a text editor, examine the output in `QM01.out` file by confirming that the last line includes the message: No commands have a syntax error

If you see errors, find and correct the error, and then rerun the command with the `-v` switch.

- 
- \_\_\_ 4. Run MQSC again to verify that messages on the SYSTEM queues on QM01 are intact and that the actual queue depths match the queue depths that were reported in [Step 1](#).

```
runmqsc QM01
DIS QL(SYSTEM.*)
end
```

## Section 2. Dumping security definitions

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

### 2.1. Write security definitions to a file

- \_\_\_ 1. Use the `dmpmqcfg` command 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. Open MQ Explorer, and in the **Navigator** pane, right-click **QM01 > Object Authorities > Save All**.
  - \_\_\_ b. Save the file as `qm01-oam.txt` in the `C:\labfiles\Lab12-restore` directory.
- \_\_\_ 3. Examine the output from the security definitions file to find the `setmqaut` entries for QL.A.
- \_\_\_ 4. Run the `dmpmqaut` command to write a formatted view of the authorities for the queue QL.A.  
`dmpmqaut -m QM01 -n QL.A -t q`
- \_\_\_ 5. Compare the output from the `dmpmqaut` command 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.

The `dmpmqaut` command 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.

### 2.2. Exercise cleanup

- \_\_\_ 1. Close all command prompts.
- \_\_\_ 2. Close Notepad if it is open.

## End of exercise

## Exercise review and wrap-up

This exercise showed you how to use the **dmpmqcfg** command to unload a queue manager's object definitions.

# Exercise 13. Monitoring IBM MQ for performance

## Estimated time

01:30

## Overview

In this exercise, you enable and configure the online monitoring, statistics, and accounting features of IBM MQ.

## Objectives

After completing this exercise, you should be able to:

- Enable accounting and statistics collection in IBM MQ
- View accounting and statistics data
- Configure a queue manager for online monitoring
- Monitor system resource usage

## Introduction

With the IBM MQ monitoring facilities, you can post-process monitoring data and monitor your queue manager online.

This exercise focuses on four types of monitoring:

- Statistics monitoring
- Accounting monitoring
- Online real-time monitoring
- System resource monitoring

In this exercise, you configure and use the monitoring functions of IBM MQ.

The exercise includes the following sections:

- [Section 1, "Collecting statistics data"](#)
- [Section 2, "Collecting accounting data"](#)
- [Section 3, "Online monitoring"](#)
- [Section 4, "Resource monitoring"](#)

## Requirements

This exercise uses queue managers that you created during [Exercise 10, "Running an IBM MQ trace"](#). You also use the `statsdata.txt` text file that is provided in the `C:\labfiles\Lab13-monitor\data` directory on Windows.

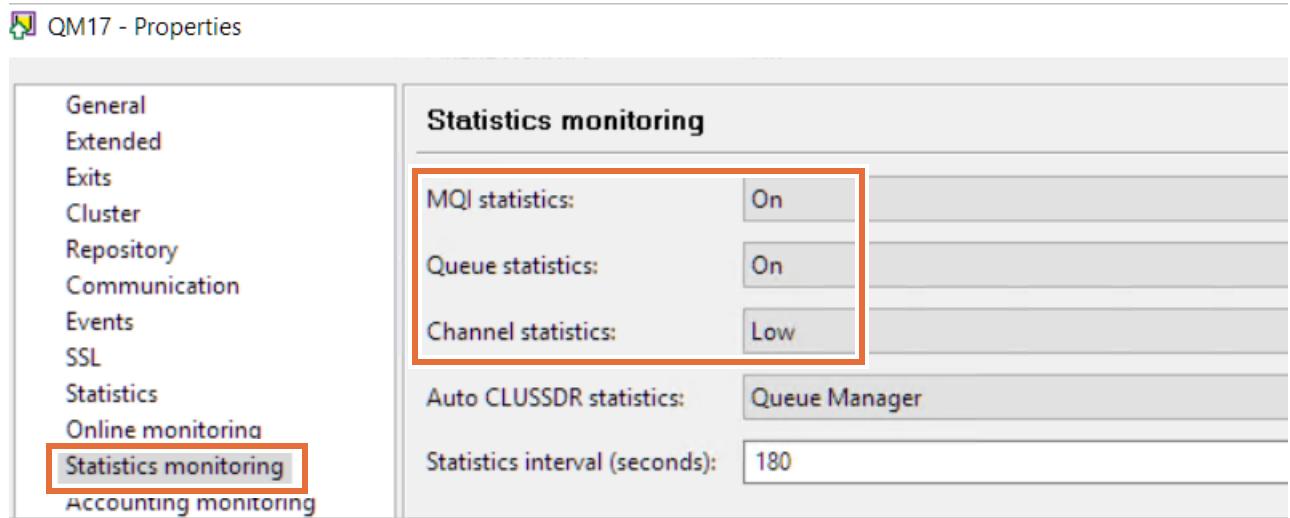
## Section 1. Collecting statistics data

In this part of the exercise, you configure a queue manager to collect statistics data. The configuration is controlled by using a set of queue manager, queue and channel attributes.

The attributes can be set by using the MQ Explorer or by using MQSC commands; instructions in this exercise use MQ Explorer.

### 1.1. Enable statistics monitoring

- \_\_\_ 1. In MQ Explorer, make sure that queue manager QM17 is running
- \_\_\_ 2. Edit the properties for queue manager QM17 to configure statistics monitoring properties.
  - \_\_\_ a. In the **Navigator** view, right-click **QM17** and click **Properties**.
  - \_\_\_ b. Click **Statistics monitoring** in the left pane, and set the following properties:
    - MQI statistics: **On**
    - Queue statistics: **On**
    - Channel statistics: **Low**
    - Statistics interval: **180**



For this exercise, you set **Statistics interval** to 180 so that statistics messages are generated every 3 minutes to demonstrate this function.

- \_\_\_ c. Click **OK**.

**Note**

You can disable statistics generation on each queue or channel by setting the **Queue statistics** (STATQ) or **Channel statistics** (STATCHL) attribute of the object to **Off**.

## 1.2. Define the queues for statistics

**Note**

This part of the exercises assumes that you successfully completed [Exercise 3, "Implementing distributed queuing"](#).

- \_\_\_ 1. Start the queue manager QM02 and verify that the listener is running.
- \_\_\_ 2. On **QM02**, create a local queue that is named: STATS.Q1
- \_\_\_ 3. On **QM17**, create these local queues:
  - STATS.Q1
  - STATS.Q2
- \_\_\_ 4. On **QM17**, create a remote queue definition with these properties:
  - **Name:** STATS.Q1.REMOTE
  - **Remote queue:** STATS.Q1
  - **Remote queue manager:** QM02
  - **Transmission queue:** QM02

The **Queues** view for **QM17** should now include: **STATS.Q1**, **STATS.Q2**, **STATS.Q1.REMOTE**
- \_\_\_ 5. On QM17, start the sender channel **QM17.QM02** and wait until the channel status is **Running**, and then check that the Listener is running.

## 1.3. Put some messages to the queue to generate statistics

- \_\_\_ 1. Run the **amqspput** sample program to put messages to the remote queue over the QM17 to QM02 channel by using the **statsdata.txt** file, which contains 536 messages.
  - \_\_\_ a. Open a command prompt and switch to the **C:\labfiles\Lab13-monitor\data** directory.  
`cd C:\labfiles\Lab13-monitor\data`
  - \_\_\_ b. Run the **amqspput** program.  
`amqspput STATS.Q1.REMOTE QM17 < C:\labfiles\Lab13-monitor\data\statsdata.txt`

This command writes messages to the transmission queue, which then is sent across the channel to queue manager QM02.

- \_\_\_ 2. In MQ Explorer, verify that you have 536 messages on STATS.Q1 on **QM02**.
- \_\_\_ 3. Use the `amqspput` program to write messages to STATS.Q2 on **QM17**.  
`amqspput STATS.Q2 QM17 < C:\labfiles\Lab13-monitor\data\statsdata.txt`
- \_\_\_ 4. Verify that you have 536 messages on STATS.Q2 on **QM17**.
- \_\_\_ 5. Use the `amqsget` program to read the messages off the STATS.Q2 queue.  
`amqsget STATS.Q2 QM17`
- \_\_\_ 6. After the `amqsget` sample program ends, use the MQ sample program `amqsmon` to read the statistics messages from **SYSTEM.ADMIN.STATISTICS.QUEUE** on **QM17**.  
`amqsmon -m QM17 -t statistics > c:\labfiles\Lab13-monitor\statistics.txt`  
MQ provides a sample program that can be used to read and format statistics and accounting data. You use `amqsmon` to read the accounting PCF messages from **SYSTEM.ADMIN.ACOUNTING.QUEUE** and write the results to text file.

## 1.4. Review the statistics data

- \_\_\_ 1. Go to the `c:\labfiles\Lab13-monitor` directory and open `statistics.txt` in a text editor.  
The file contains several statistics messages that your tests generate. At least 3 statistics messages are generated in each statistics interval (as specified on the **Statistics Interval Queue Manager** property).  
One message shows queue manager statistics. One message is generated for each queue to show queue statistics. One message is generated for the channel to show channel statistics.
- \_\_\_ 2. Examine the file and locate the following messages:
  - The queue manager statistics message for the test period
  - The queue statistics message for STATS.Q2 for the test period
  - The queue statistics message for QM02 (that is, the transmission queue that is used to send messages to your other queue manager)
  - The channel statistics message for QM17.QM02 for the test period
- \_\_\_ 3. Close the file when you are done.

## 1.5. Disable statistics monitoring

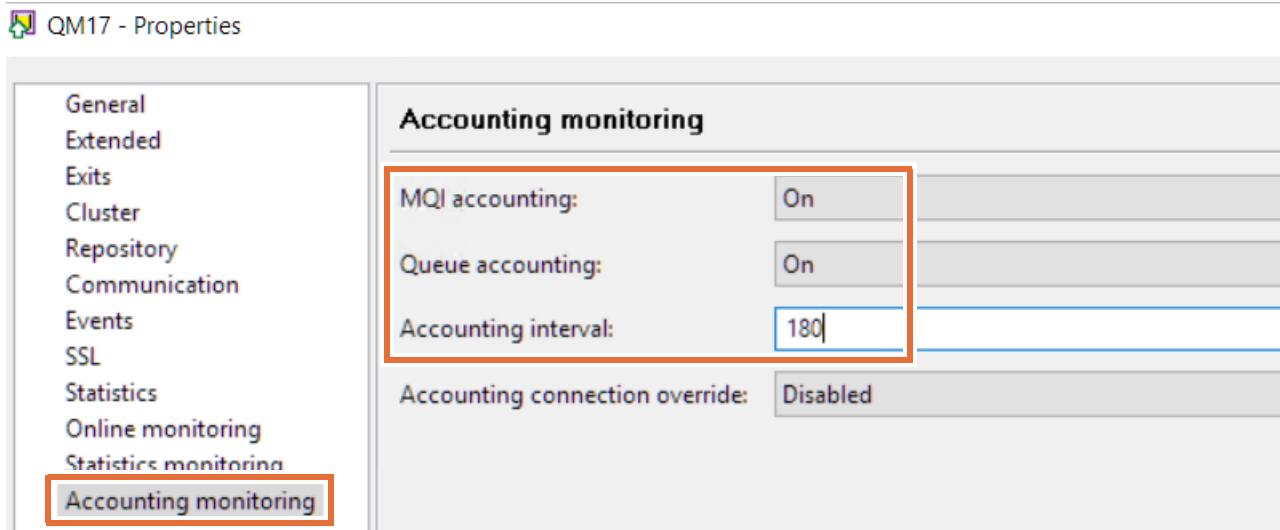
- \_\_\_ 1. In **MQ Explorer**, in the **Navigator** view, right-click **QM17** and click **Properties**.
- \_\_\_ 2. Click **Statistics monitoring** and set the following properties in **Statistics monitoring** window:
  - MQI statistics: **Off**
  - Queue statistics: **Off**
  - Channel statistics: **Off**
- \_\_\_ 3. Click **Apply**. You can leave the window open to set the next properties.

## Section 2. Collecting accounting data

In this part of the exercise, you configure a queue manager to collect accounting data. The configuration is controlled by using a set of queue manager, queue, and channel properties.

### 2.1. Enable accounting monitoring

- 1. In the QM17 properties dialog box, click **Accounting monitoring** in the left pane and set these attributes:
  - MQI accounting: **On**
  - Queue accounting: **On**
  - Accounting interval: **180**



- 2. Click **OK**.

### 2.2. Generate and view accounting data

In this part of the exercise, you run IBM MQ sample programs to simulate MQ activity and use the sample program to read and format the generated PCF messages. The test programs write messages to a local queue, and read them back off the queue.

- 1. On QM17, create a local queue that is named: ACCTG.STATS.Q1
- 2. From a command window, run the `amqspput` program with the `statsdata.txt` file to put messages to the local queue, ACCTG.STATS.Q1.  
`amqspput ACCTG.STATS.Q1 QM17 < C:\labfiles\Lab13-monitor\data\statsdata.txt`  
 The `statsdata.txt` file is a text file of 536 lines, which generates a message for each line.
- 3. Use MQ Explorer to check the queue depth of queue ACCTG.STATS.Q1 and verify that the queue contains 536 messages.
- 4. Use the `amqsget` program to read and get the messages off the queue.  
`amqsget ACCTG.STATS.Q1 QM17`

- \_\_\_ 5. Display the accounting data that was generated.

```
amqsmon -m QM17 -t accounting > c:\labfiles\Lab13-monitor\accounting.txt
```

- \_\_\_ 6. Go the `c:\labfiles\Lab13-monitor` directory and open the `accounting.txt` file in a text editor.

The file contains several messages that the `amqsput` and `amqsget` sample programs generate when you use them.

Look at the file and locate the messages that are associated with `amqsput` and `amqsget`. Each program (`amqsput` and `amqsget`) generates two messages. One message is generated to show MQI accounting, that is, the MQI operations that get each application runs. The other message is generated to show queue accounting, that is, the details of the queue that the application uses.



### Note

For applications that use multiple queues, a separate message exists for each queue.

Look at the file and locate the MQI accounting and queue accounting messages for each application.

## 2.3. Disable accounting monitoring

- \_\_\_ 1. In MQ Explorer, reopen the properties for QM17.
- \_\_\_ 2. Click **Accounting monitoring** in left pane and set the following attributes:
  - MQI accounting: **Off**
  - Queue accounting: **Off**
- \_\_\_ 3. Click **Apply**.

## Section 3. Online monitoring

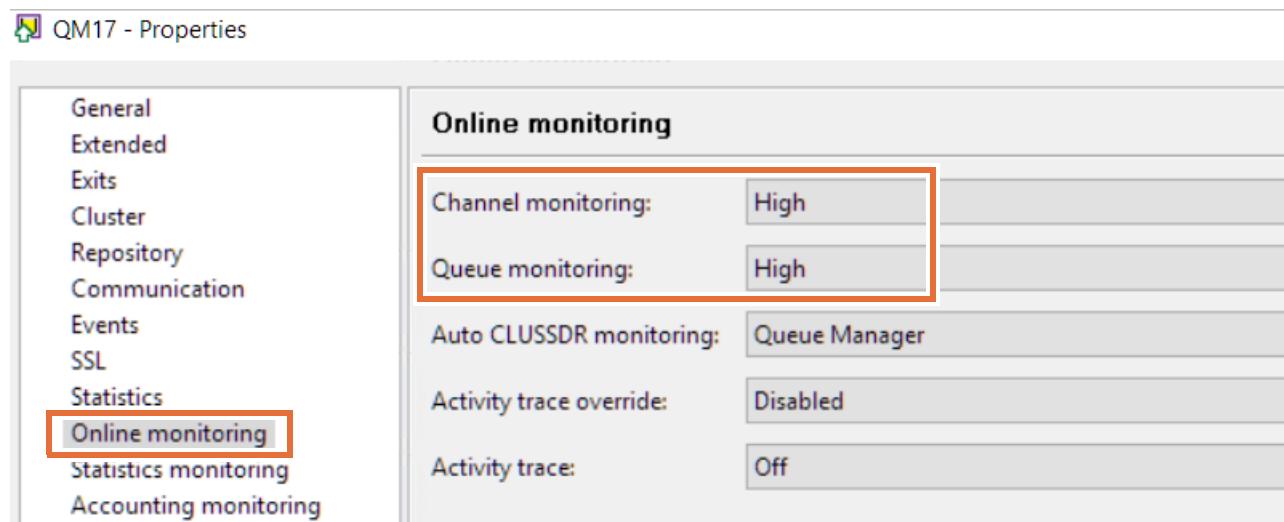
In this section, you configure a queue manager for online monitoring. By default, online monitoring data is not collected. It must be enabled by using a set of queue manager attributes.

The configuration of monitoring data collection has three values to enable the collection of data (LOW, MEDIUM, and HIGH). Collecting monitoring data might require that you run some instructions that affect performance (for example, obtaining system time). To reduce the impact of online monitoring, the resource-intensive measurements are sampled instead of measuring every value.

The online monitoring values can be set on each queue manager. The configuration is controlled by using a set of queue manager, queue, and channel attributes. These values can be set by using the MQ Explorer or MQSC commands; the instructions in this exercise are for MQ Explorer.

### 3.1. Enable online monitoring and generate data

- 1. In the QM17 properties dialog box, select **Online monitoring** in the left pane, and set the following values:
  - Channel monitoring: **High**
  - Queue monitoring: **High**



- 2. Run the `amqspput` sample program with the `statsdata.txt` file to put 536 messages on STATS.Q2.

```
amqspput STATS.Q2 QM17 < C:\labfiles\Lab13-monitor\data\statsdata.txt
```

- 3. Run the `amqsget` sample program against STATS.Q2 to get the messages.

```
amqsget STATS.Q2 QM17
```

- 4. Review the generated online monitoring data for QM17 by using MQSC commands.

```
runmqsc QM17
DIS QSTATUS(STATS.Q2) MONITOR
end
```

This command displays the monitoring data for the queue. This table summarizes the monitoring fields.

|          |                                                                                                                                                                                                                       |
|----------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| QTIME    | The interval between when messages are put on the queue and when they are destructively read. The value is shown in microseconds, formatted into a string of up to 9 digits.                                          |
| LPUTDATE | Date on which the last message was put to the queue since the queue manager started. If no PUT date is available, then the value is blank.                                                                            |
| LPUTTIME | Time at which the last message was put to the queue since the queue manager start. If no put time is available, the value shows as blank.                                                                             |
| LGETDATE | Date on which the last message was retrieved from the queue since the queue manager start. A message that is browsed does not count as a retrieved message. If no get date is available, the value is shown as blank. |
| LGETTIME | Time at which the last message was retrieved from the queue since the queue manager start. A browsed message does not count as a retrieved message. If no get time is available, the value is shown as blank.         |
| MONQ     | Current level of monitoring data collection for the queue. Possible values: OFF, LOW, MEDIUM, HIGH.                                                                                                                   |
| MSGAGE   | Age, in seconds, of the oldest message on the queue. The maximum displayable value is 999 999 999. If the age exceeds this value, 999 999 999 is displayed.                                                           |

### 3.2. Disable online monitoring

- \_\_\_ 1. Disable online monitoring on the QM17 queue manager.
  - \_\_\_ a. In MQ Explorer, open the properties for queue manager QM17.
  - \_\_\_ b. Click **Online monitoring**.
  - \_\_\_ c. Deactivate recording of statistics for channels and queues. In the **Online Monitoring** properties for QM17, set the following values:
    - Channel monitoring: **Off**
    - Queue monitoring: **Off**
  - \_\_\_ d. Click **OK**.

## Section 4. Resource monitoring

In this part of the exercise, you use the `amqsrua` sample program to enable and display MQ resource monitoring data. The `amqsrua` command reports metadata that is published by queue managers. This data can include information about the CPU, memory, and disk usage.

In this part of the exercise, you need two command windows (sessions). In one window, you run the `amqsrua` sample program. In the second window, you run the commands to put and get messages to a queue on a QM17

### 4.1. Get a queue status summary

- \_\_\_ 1. In a command prompt, start the `amqsrua` program:

```
amqsrua -m QM17
```



#### Note

The `amqsrua` program is case-sensitive.

- \_\_\_ 2. When prompted with the set of resource classes, type `STATQ` and press Enter.
- \_\_\_ 3. When prompted to specify the type of action to monitor, type `PUT` and press Enter.
- \_\_\_ 4. When prompted for the name of the queue to monitor, type `STATS.Q1` and press Enter.

The `amqsrua` sample program is now subscribing to the specified class of resource monitoring events. Using the command in this way reports on the values of those resource elements every 10 seconds. The queue is empty so all values are zero.

- \_\_\_ 5. Open a second command window (beside the one running `amqsrua`) and run the `amqsput` program to put messages to the `STATS.Q1` queue on QM17.

```
amqsput STATS.Q1 QM17 < C:\labfiles\Lab13-monitor\data\statsdata.txt
```

- \_\_\_ 6. Observe the counts in queue status window. You should see that the MQPUT non-persistent message count is now 536.

```
STATS.Q1 queue avoided bytes 0.00%
Publication received PutDate:20191203 PutTime:06542904 Interval:10.000 seconds
STATS.Q1 MQPUT/MQPUT1 count 536 54/sec
STATS.Q1 MQPUT byte count 35376 3537/sec
STATS.Q1 MQPUT non-persistent message count 536
STATS.Q1 MQPUT1 persistent message count 0
STATS.Q1 MQPUT1 non-persistent message count 0
STATS.Q1 MQPUT1 persistent message count 0
STATS.Q1 non-persistent byte count 35376 3537/sec
STATS.Q1 persistent byte count 0
STATS.Q1 lock contention 0.00%
STATS.Q1 queue avoided puts 0.00%
STATS.Q1 queue avoided bytes 0.00%
Publication received PutDate:20191203 PutTime:06543904 Interval:10.000 seconds
STATS.Q1 MQPUT/MQPUT1 count 0
STATS.Q1 MQPUT byte count 0
STATS.Q1 MQPUT non-persistent message count 0
STATS.Q1 MQPUT persistent message count 0
STATS.Q1 MQPUT1 non-persistent message count 0
STATS.Q1 MQPUT1 persistent message count 0
STATS.Q1 non-persistent byte count 0
STATS.Q1 persistent byte count 0
STATS.Q1 lock contention 0.00%
STATS.Q1 queue avoided puts 0.00%
STATS.Q1 queue avoided bytes 0.00%
```

- \_\_\_ 7. End the `amqsrua` program by pressing **Ctrl+C**.
- \_\_\_ 8. Run the `amqsrua` program in interactive mode to get the log statistics.

- \_\_\_ a. In the command windows, type:

```
amqsrua -m QM17
```

- \_\_\_ b. For the class selection, type `DISK` and press Enter.
- \_\_\_ c. For the type selection, type `Log` and press Enter.

**Note:** Make sure that you do not use upper-case when you type: `Log`

The Log statistics include the bytes in use, the maximum bytes available, physical bytes written, logical bytes written, and write latency.

The command should return output that is similar to the following example.

```
Publication received PutDate:20191203 PutTime:07015046 Interval:10.000 seconds
Log - bytes in use 50331648
Log - bytes max 83886080
Log - physical bytes written 0
Log - logical bytes written 0
Log - write latency 1420 uSec
Log - write size 5400
Log - current primary space in use 2.53%
Log - workload primary space utilization 2.53%
```

- \_\_\_ d. End the `amqsrua` program by pressing **Ctrl+C**.
- \_\_\_ 9. Run the `amqsrua` program in command-line mode to get message data on directories for errors, trace, and the number of MQ FDC records.

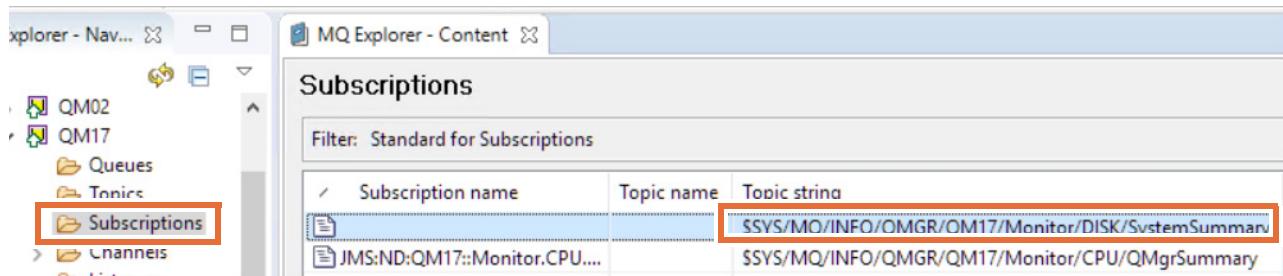
```
amqsrua -m QM17 -c DISK -t SystemSummary
```

The command should return output similar to the following example:

```
Publication received PutDate:20191203 PutTime:07041017 Interval:10.003
seconds
MQ errors file system - bytes in use 38781MB
MQ errors file system - free space 68.43%
MQ FDC file count 2
MQ trace file system - bytes in use 38781MB
MQ trace file system - free space 68.43%
```

- \_\_\_ 10. In MQ Explorer, verify that the `amqsrua` sample program is subscribed to the MQ resource monitoring topics.
  - \_\_\_ a. In MQ Explorer, click the **QM17 > Subscriptions** folder.
  - \_\_\_ b. Verify that you see a subscription for the topic string:

```
$SYS/MQ/INFO/QMGR/QM17/Monitor/DISK/SystemSummary
```



## Information

You can run the `amqsrua` program with the argument `-d 1` to enable debug mode. The results show the system topics that the program subscribes to.

- 
- \_\_\_ 11. In the command prompt, end the `amqsrua` program by pressing **Ctrl+C**.

## 4.2. Exercise cleanup

- 1. Close all command prompts.
- 2. Close any open text files.

**End of exercise**

## Exercise review and wrap-up

The first part of the exercise ...

# Exercise 14. Monitoring resources with the IBM MQ Console

## Estimated time

01:00

## Overview

In this exercise, you use the IBM MQ Console to monitor system resources. You also learn how to configure and share dashboard layouts.

## Objectives

After completing this exercise, you should be able to:

- Monitor system resources
- Configure dashboard layouts
- Share an IBM MQ Console dashboard between user roles

## Introduction

This exercise explores some of the monitoring capabilities of IBM MQ Console.

The exercise includes these sections:

- [Section 1, "Configuring IBM MQ Console to monitor MQ objects"](#)
- [Section 2, "Testing security with the basic registry"](#)



### Information

The IBM MQ Console does not currently offer all of the configuration functions of IBM MQ Explorer.

## Requirements

This exercise uses the queue managers and queues that you created in [Exercise 1, "Getting started with IBM MQ"](#).

# Section 1. Configuring IBM MQ Console to monitor MQ objects

In this part of the exercise, you create custom dashboards for monitoring and problem determination purposes.

## 1.1. Sign in to IBM MQ Console

- \_\_\_ 1. Connect to the IBM MQ console from a web browser.
  - \_\_\_ a. In Mozilla Firefox, go to this URL:  
<https://localhost:9443/ibmmq/console>
  - \_\_\_ b. If prompted about security, accept the risk and continue.



### Troubleshooting

If you are unable to connect, the mqweb server might not be running. To start the server, open a command prompt and type:

```
strmqweb
```

- \_\_\_ 2. Sign in to the console with these credentials:
  - **User Name:** mqadmin
  - **Password:** mqadmin

## 1.2. Configure the dashboard

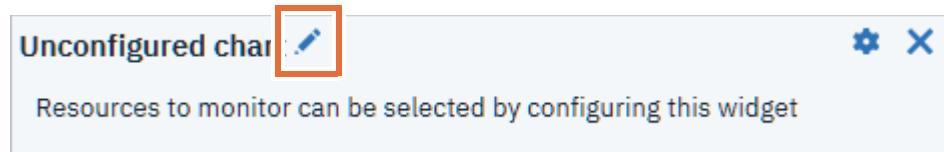
- \_\_\_ 1. Create a tab for monitoring some aspects of the environment.
    - \_\_\_ a. Click the **Add a new tab** icon.
- The screenshot shows the IBM MQ console interface. At the top, there's a header with the IBM MQ logo and a dropdown menu labeled 'Overview'. To the right of the dropdown is a red-bordered '+' button. Below the header, the text 'Lab environment' is displayed. The entire interface is contained within a light gray box.
- \_\_\_ b. Set **Tab name to:** Monitoring
  - \_\_\_ c. Set **Description to:** Monitor system resources
  - \_\_\_ d. Click **Add**.
- \_\_\_ 2. Add a Chart widget to the **Monitoring** tab.
    - \_\_\_ a. On the **Monitoring** tab, click **Add widget**.

- \_\_\_ b. Click the **Chart** link.



- \_\_\_ 3. Rename the widget.

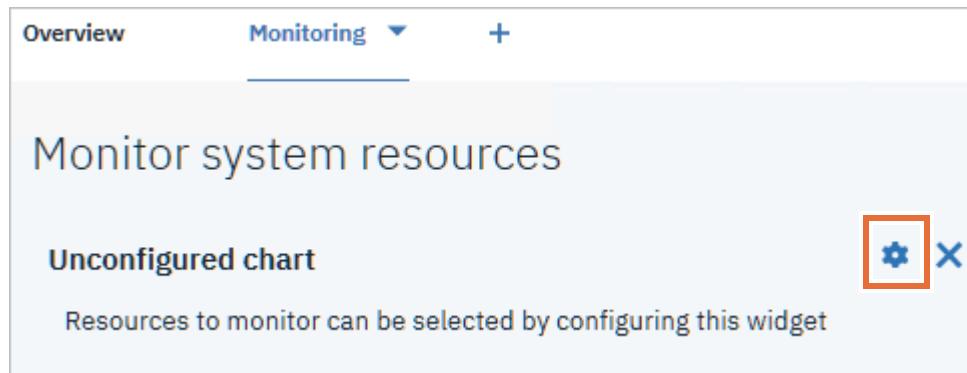
- \_\_\_ a. Hover your mouse over the widget title until you see the Edit (pencil) icon and click it.



- \_\_\_ b. In the **Widget title** field, type CPU for queue managers and click **Rename**.

- \_\_\_ 4. Configure the chart widget to show the CPU estimation for the queue managers that are configured in this widget.

- \_\_\_ a. Click the **Configure widget** (gear) icon.



\_\_ b. Set these properties:

- **Resource type: CPU performance - running queue manager**
- **Resource element: System CPU time - percentage estimate for queue manager.**

**Resource to monitor**

Resource class: Platform central processing units      Resource type: CPU performance - running queue manager

Resource element: System CPU time - percentage estimate for queue manager

**Queue managers to monitor**

QM01      Add queue manager

**View finder**  
 Show     Hide

\_\_ c. Click **Add queue manager** and select any running queue manager.

**Queue managers to monitor**

QM01      Remove

QML01      Remove      Add queue manager

Depending on which queue managers are running on your system, your queue manager list might differ from the screen capture that is shown here.

Each queue manager is assigned a unique color for monitoring purposes.

\_\_ d. Set the **View finder** option to **Show** and click **Save**.

**View finder**  
 Show     Hide

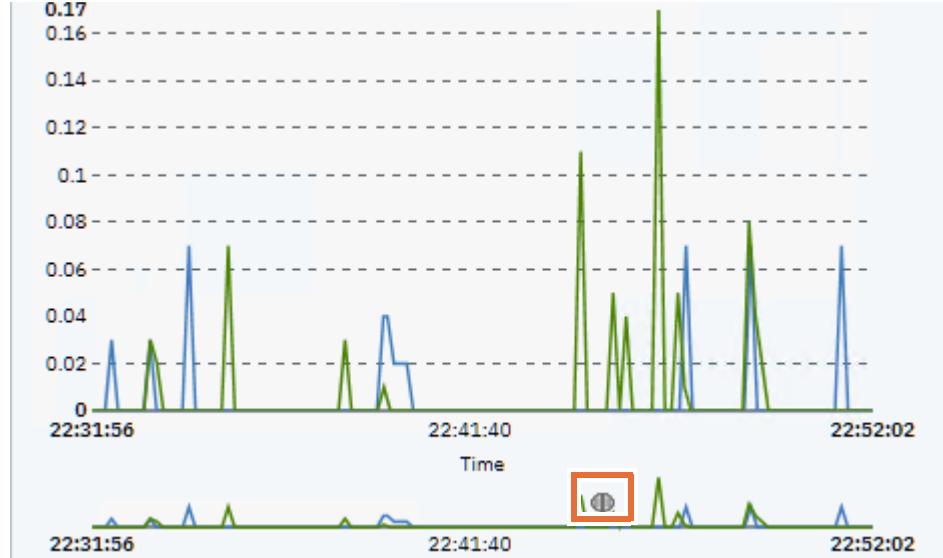
Cancel      Save



The View finder allows you to focus on specific time intervals in the chart. This widget feature can be useful when you notice sudden changes in graph activity.

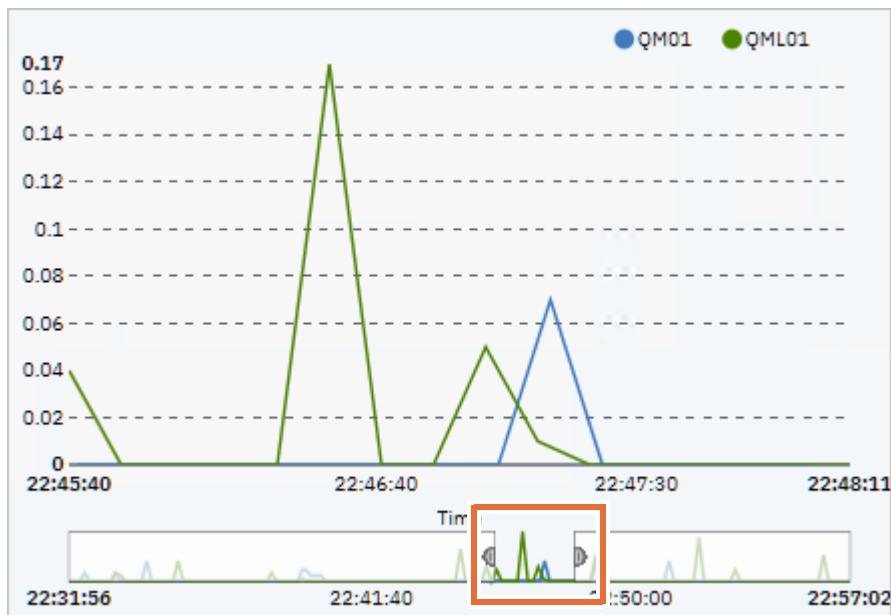
- \_\_\_ 5. Use the View finder to focus on a time interval of interest.
- \_\_\_ a. Use your mouse to click a point on the smaller graph.

You see that a small circle added to the graph.



- \_\_\_ b. With your mouse on the small circle, drag your mouse to the left or right to select a time interval.

The larger graph zooms in on the section that you select in the lower graph.



- \_\_\_ 6. Add another chart widget to the **Monitoring** tab that is named RAM usage estimation
- \_\_\_ a. Change the title to: RAM usage estimation
- \_\_\_ b. Click the **Configure widget** (gear) icon.
- \_\_\_ c. Click **Add queue manager** and add all the running queue managers.

\_\_ 7. Explore the monitoring options.

- \_\_ a. In the **Resource class** list, click each option to see how the select affects the **Resource type** and **Resource element** lists.

You see that the options available in the other lists depend on your **Resource class** selection.

- \_\_ b. Complete the widget by setting the following properties:

- **Queue managers to monitor:** Include all running queue managers
- Properties to monitor:
  - **Resource class:** Platform central processing units
  - **Resource type:** CPU performance - running queue manager
  - **Resource element:** RAM total bytes - estimate for queue manager
- **View finder:** Show

### Chart

#### Resource to monitor

|                   |                                                                           |                |                                                                      |
|-------------------|---------------------------------------------------------------------------|----------------|----------------------------------------------------------------------|
| Resource class:   | <input type="text" value="Platform central processing units"/>            | Resource type: | <input type="text" value="CPU performance - running queue manager"/> |
| Resource element: | <input type="text" value="RAM total bytes - estimate for queue manager"/> |                |                                                                      |

#### Queue managers to monitor

|      |                                       |
|------|---------------------------------------|
| QM01 | <input type="button" value="Remove"/> |
| QM02 | <input type="button" value="Remove"/> |
| QM17 | <input type="button" value="Remove"/> |

#### View finder

Show  Hide

## Section 2. Testing security with the basic registry

In [Exercise 1, "Getting started with IBM MQ"](#), you configured IBM MQ Console to use the basic registry for user access. You are currently working in the Console as mqadmin, which is an administrative user role (MQWebAdmin). In this part of the exercise, you test the mqreader user role (MQWebUser).



### Information

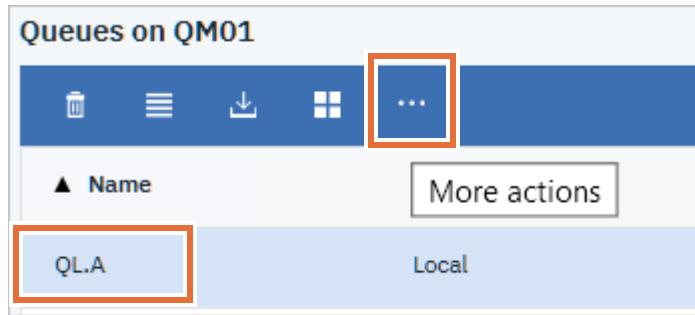
With the basic registry, any users that are assigned to either the MQWebAdmin or MQWebAdminRO roles can access MQ objects, according to the access that is granted to the user ID that the mqweb server is running under. Console users that resolve to the MQWebUser role have access according to what is defined in the queue manager for the operating system user ID that they signed in with.

### 2.1. Review the permissions for administrative users.

- \_\_\_ 1. Click the **Overview** tab in the console.
- \_\_\_ 2. Open the **Queues on QM01** widget.

If you no longer have the **Queues on QM01** widget, create one as described in [Section 3.6, "Add a widget"](#) in [Exercise 1, "Getting started with IBM MQ"](#).

- \_\_\_ 3. In the **Queues on QM01** widget, select a queue and click the **More actions (...)** menu.



- \_\_\_ 4. Click **Manage authority records**.

- \_\_\_ 5. Click the Administrator entity and notice which permissions are set for this user role.

Manage authority records for 'QL.A' on QM01

| Delete                  |             | 1 item selected | <a href="#">Deselect</a> |
|-------------------------|-------------|-----------------|--------------------------|
| ▲ Entity name           | Entity type |                 |                          |
| Administrator@WS2016X64 | User        |                 |                          |
| mqm@WS2016X64           | Group       |                 |                          |

Total: 2      Last updated: 12:20:41 AM

| Administration                                                                                                                                                                       | Context                                                                                                                                                                                                                              | MQI                                                                                                                                                                                                                        |
|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <input checked="" type="checkbox"/> Change<br><input checked="" type="checkbox"/> Clear<br><input checked="" type="checkbox"/> Delete<br><input checked="" type="checkbox"/> Display | <input checked="" type="checkbox"/> Pass all context<br><input checked="" type="checkbox"/> Pass identity context<br><input checked="" type="checkbox"/> Set all context<br><input checked="" type="checkbox"/> Set identity context | <input checked="" type="checkbox"/> Browse<br><input checked="" type="checkbox"/> Inquire<br><input checked="" type="checkbox"/> Get<br><input checked="" type="checkbox"/> Put<br><input checked="" type="checkbox"/> Set |

[Check all](#) [Uncheck all](#) [Close](#) [Save](#)

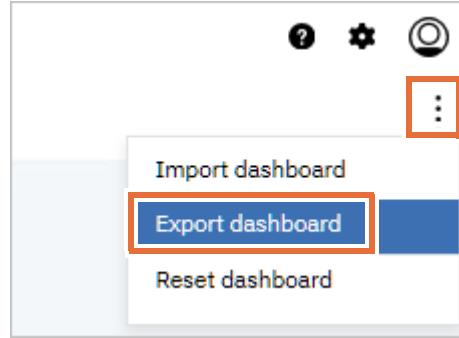
You signed in to the VMware image for you lab environment with the user name: Administrator. You also started the mqweb server while logged in as Administrator. The entity name associated with the authority record is the Windows Administrator user ID.

- \_\_\_ 6. Click **Close**.

## 2.2. Export the dashboard

- \_\_\_ 1. Save your dashboard layout by exporting it.

- \_\_\_ 2. Click the control menu in the upper-right corner of the dashboard and click **Export**.

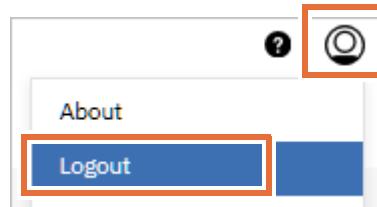


- \_\_\_ 3. When prompted to save the file, click **OK**.

The dashboard is saved as a JSON file in the browser's **Downloads** directory.

## 2.3. Test mqreader user permissions for MQ objects

- \_\_\_ 1. In the upper-right corner of the dashboard, click the profile icon and click **Logout** to return to the IBM MQ Console login page.

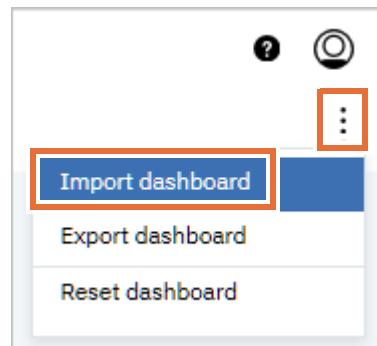


- \_\_\_ 2. Sign in to the console with these credentials:

- **User name:** mqreader
- **Password:** mqreader

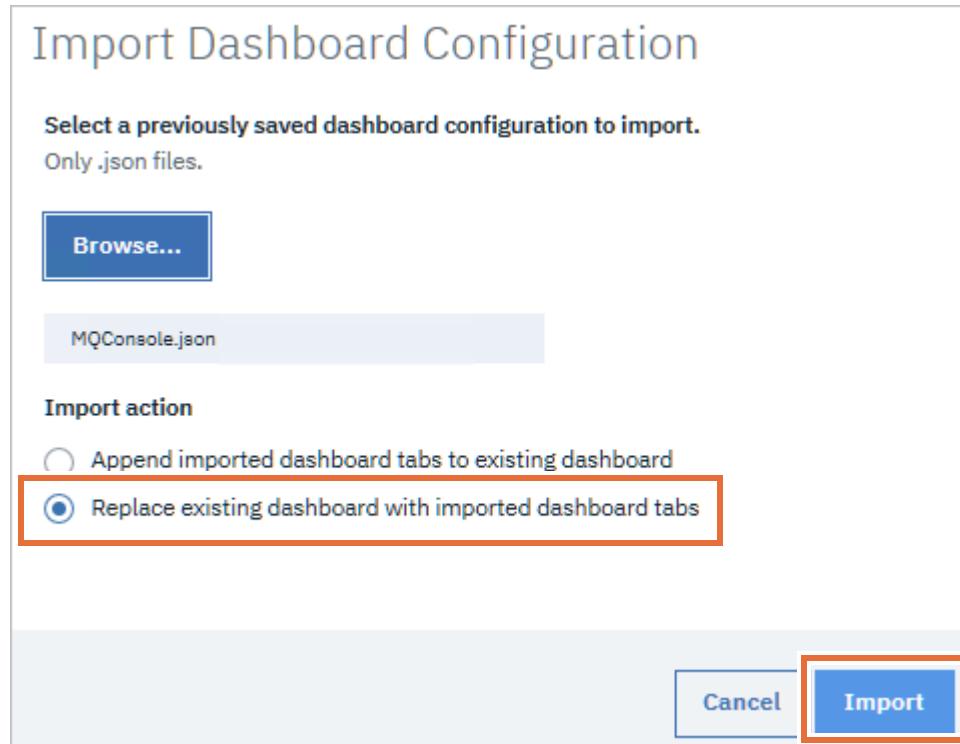
Because mqreader user has not customized the IBM MQ Console, you see the default layout.

- \_\_\_ 3. Click the control menu icon and click **Import dashboard**.



- \_\_\_ 4. Click **Browse**, navigate to the `MQConsole.json` file in the **ThisPC > Downloads** folder, and click **Open**.

- \_\_\_ 5. Select **Replace existing dashboard with imported dashboard tabs** and click **Import**.



The customized dashboard opens.

- \_\_\_ 6. In the **Local Queue Managers** widget, select a queue manager and notice which menu options are available.
- Can you delete the queue manager?
  - Can you create a new queue manager?

The mqreader privileges include limited permissions to manage queue managers.

- \_\_\_ 7. In the **Queues on QM01** widget, try to create a queue.
- \_\_\_ a. Make sure that you do not have a queue selected, and click the **Create** icon.
  - \_\_\_ b. When prompted for a name, type a name, such as `myQueue` and click **Create**.

An error message opens because mqreader does not have authority to create MQ objects, only to display them.

- \_\_\_ 8. Open the **Monitoring** tab and notice that you can view the chart data.  
 \_\_\_ 9. Sign out of IBM MQ Console when you are done.

## End of exercise

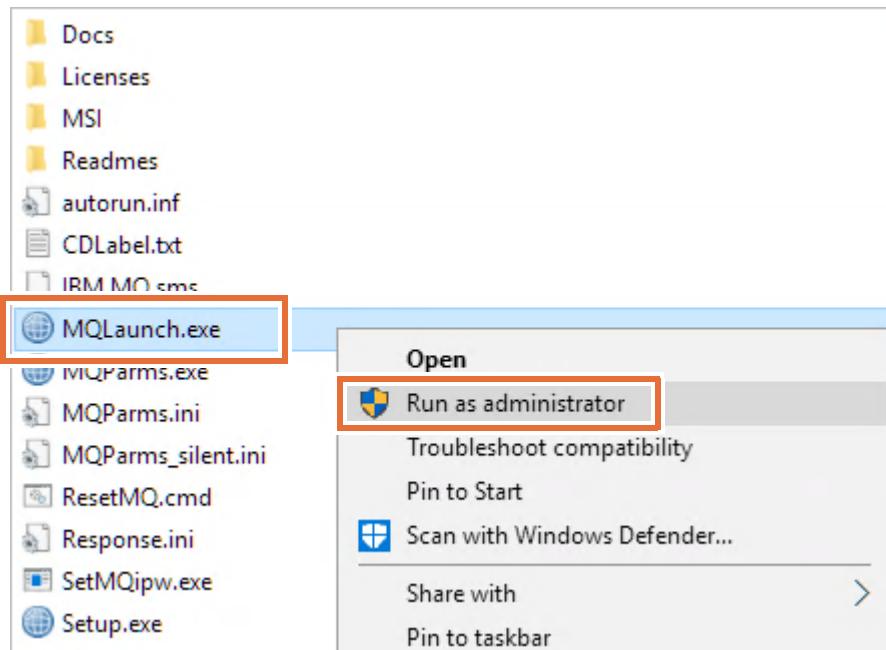
## Exercise review and wrap-up

This exercise demonstrated how to use the IBM MQ Console for monitoring system resources and configuring dashboard layouts.

# Appendix A. Installing IBM MQ V9.1.0.1

The following steps were used to install IBM MQ V9.1.0.1 for this course.

- 1. Run the Launchpad using the **Run as administrator** option.
  - a. In the directory where you downloaded the installation files, right-click **MQLaunch.exe** and click **Run as administrator**.



- \_\_ b. From the Welcome page, click **IBM MQ Installation**, click **Launch IBM MQ Installer**, and when prompted, click **Run**.



- \_\_ c. Accept the license agreement, choose **Typical** as the setup type, and click **Install**.  
This option installs all components (client, server, and IBM MQ Explorer), and uses the default installation name: Installation1  
After the installation completes, the Prepare IBM MQ Wizard opens.
- \_\_ 2. In Prepare IBM MQ Wizard, click **Next**, and when prompted about domain controllers, select **No** and click **Next**.
- \_\_ 3. After the Prepare IBM MQ Wizard completes, close IBM MQ Explorer and any other open windows.
- \_\_ 4. Set the PATH environment variable.
- \_\_ a. Add the following paths to the PATH environment variable:
- C:\Program Files\IBM\MQ\Tools\C\Samples\Bin64
  - C:\Program Files\IBM\MQ\java\jre\bin
- \_\_ b. Open a command prompt and verify that the paths were set correctly by typing:
- For the Java path, type:  
`java -version`
  - For the MQ sample program path, type:  
`amqspput`

- The command should return a syntax error about a missing parameter.
- \_\_\_ c. Close the command prompt.
  - \_\_\_ 5. In Windows Explorer, make sure that **Hidden Items** on the **View** menu is selected so that the C:\ProgramData\IBM\MQ directory is visible.



IBM Training



© Copyright International Business Machines Corporation 2020.