

Course Exercises Guide

Designing, Implementing, and Managing IBM MQ V9 Clusters

Course code WM253 ERC 1.0



June 2017 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 2017.

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 1. Configuring and reviewing base IBM MQ resources	1-1
Section 1: Log on to the lab image and review environment	1-2
Section 2: Create the queue managers that are used in Exercise 2	1-3
Section 3: Start and display the queue managers	1-4
Section 4: Review the object definitions and changes that are required for queue manager MQE5 and run the script by using the <code>runmqsc</code> utility	1-5
Section 5: Determine the value of the BIND attribute in the queues that were created	1-8
Section 6: Run the script to create the object definitions and changes that are required for queue manager MQG1 by using the <code>runmqsc</code> utility	1-9
Section 7: Test connectivity from queue managers MQE5 to MQG1	1-9
Section 8: Test connectivity from queue managers MQG1 to MQE5	1-10
Section 9: Review the queue manager logs for queue manager MQG1	1-12
Exercise 2. Implementing and verifying a cluster, cluster queues, and a cluster gateway	2-1
2.1. Configure and verify cluster WMADMCLS	2-4
Section 1: Review and configure the commands that are required to add queue manager MQ0A as a full repository of cluster WMADMCLS	2-4
Section 2: Review and configure the commands that are required to add queue manager MQ0B as a full repository of cluster WMADMCLS	2-6
Section 3: Run DIS CLUSQMGR command from each MQ0A and MQ0B to confirm the success of the first part of the configuration	2-8
Section 4: Review and run the commands that are required to incorporate queue manager MQG1 to cluster WMADMCLS	2-15
Section 5: Run the DIS CLUSQMGR command from MQG1 to verify a successful incorporation of queue manager MQG1 to the cluster	2-17
Section 6: Review and configure the commands that are required to incorporate queue manager MQ03 to cluster WMADMCLS	2-20
Section 7: Run DIS CLUSQMGR command from MQ03 to verify a successful incorporation of queue manager MQ03 to the cluster	2-22
2.2. Define and display cluster queues	2-26
Section 1: Define a cluster queue by the same name in queue managers MQ0A and MQ03	2-26
Section 2: Attempt to display the new cluster queues from cluster member queue manager MQG1	2-26
2.3. Configure the cluster gateway and run a workload balancing baseline	2-28
Section 1: Create required definitions in non-cluster queue manager MQE5	2-29
Section 2: Create required definitions to make cluster queue manager MQG1 the gateway queue manager to cluster WMADMCLS	2-29
Section 3: Baseline the round-robin workload balancing behavior by sending messages from the outside queue manager MQE5	2-29
Section 4: Check the number of messages that arrived at the WM253.IN queue at queue manager MQ0A	2-30
Section 5: Check the number of messages that arrived at the WM253.IN queue at queue manager MQ03	2-32
Section 6: Repeat the test with a higher number of messages	2-33
2.4. Redisplay the cluster queues	2-35
Section 1: Return to queue manager MQG1 and repeat the cluster queue display	2-35

Exercise 3. Working with cluster administration tasks	3-1
3.1. Add a queue manager that uses separate transmission queues to the cluster	3-2
Section 1: Create and start queue manager MQ0X to listen on port 1670	3-2
Section 2: Alter queue manager MQ0X to use a separate transmission queue before adding the queue manager to the cluster	3-3
Section 3: Define the MQ0X CLUSRCVR channel for cluster WMADMCLS	3-4
Section 4: Define the MQ0X CLUSSDR channel for cluster WMADMCLS	3-4
Section 5: Verify and review results of the addition of queue manager MQ0X to cluster WMADMCLS	3-4
Section 6: Review the results of the addition of queue manager MQ0X to the cluster from the perspective of queue manager MQ0A	3-8
3.2. Inhibit and restore use of a cluster queue manager by using SUSPEND and RESUME	3-12
Section 1: Create queue cluster queue MQ03.ONLY at queue manager MQ03	3-13
Section 2: Advise WMADMCLS queue managers to stop sending messages to queue manager MQ03 by using the SUSPEND command	3-13
Section 3: Check the status of the cluster channels after the SUSPEND from the perspective of the MQ03 queue manager	3-14
Section 4: Use queue manager MQG1 to send messages to cluster queue MQ03.ONLY, hosted in queue manager MQ03 exclusively	3-14
Section 5: Determine whether the messages sent to queue MQ03.ONLY arrived at suspended queue manager MQ03	3-15
Section 6: Use queue manager MQG1 to send messages to cluster queue WM253.IN, which is hosted in queue managers MQ03 and MQ0A	3-15
Section 7: Determine whether the messages sent to queue WM253.IN arrived at suspended queue manager MQ03	3-16
Section 8: Restore use of queue manager MQ03 to cluster WMADMCLS	3-16
Section 9: Review the entries in the MQ03 queue manager log	3-17
Section 10: Repeat the SUSPEND command for queue manager MQ03 with MODE(FORCE)	3-18
Section 11: Use queue manager MQG1 to send messages to cluster queue MQ03.ONLY, hosted in queue manager MQ03 exclusively	3-19
Section 12: Determine whether the messages sent to queue MQ03.ONLY arrived at suspended queue manager MQ03	3-19
Section 13: Check the cluster transmission queue for queue manager MQG1	3-20
Section 14: Check the entries in the MQG1 queue manager log	3-20
Section 15: Restore use of queue manager MQ03 to cluster WMADMCLS	3-22
Section 16: Review the status of the MQ03 cluster channels by using the DIS CLUSQMGR command	3-22
Section 17: Review the status of the dynamic WMADMCLS.MQ03 CLUSSDRA channel from queue manager MQG1	3-23
Section 18: Restart CLUSSDRA channel WMADMCLS.MQ03 from MQG1	3-24
Section 19: Repeat the DIS CLUSQMGR command with the deftype and status attributes and review the results	3-24
Section 20: Ensure that the transmission queue for queue manager MQG1 is clear of messages	3-26
3.3. Remove MQ0X from the cluster by using the alternative process with RESET	3-27
Section 1: Confirm and stop the cluster channels at queue manager MQ0X	3-27
Section 2: Confirm whether all channels stopped	3-28
Section 3: Remove the manually defined CLUSSDR channel to queue manager MQ0A from queue manager MQ0X	3-29
Section 4: Remove knowledge of queue manager MQ0X from the cluster by using the RESET command from a full repository queue manager	3-29
Section 5: Repeat the DIS CLUSQMGR for MQ0X in each member of the cluster, MQ0B, MQG1, and MQ03	3-32
Section 6: After you confirmed that activity for MQ0X ceased, remove the CLUSRCVR channel from queue manager MQ0X	3-32
3.4. Use the REFRESH CLUSTER command	3-34
Section 1: Refresh the cluster from full repository MQ0B	3-34

Section 2: Review the entries that are made in the MQ0B queue manager log as a result of the REFRESH command	3-34
3.5. Take an initial look at the IBM MQ Explorer cluster administration capabilities	3-37
Exercise 4. Working with IBM MQ security	4-1
4.1. Baseline your environment	4-4
Section 1: Confirm that the sending of messages from non-cluster queue manager MQE5 to queue WM253.IN via MQG1 is operational	4-4
Section 2: Check the cluster channels from the perspective of queue manager MQG1	4-5
4.2. Apply strict connection authentication and channel authentication to MQG1	4-7
Section 1: Review the commands to tighten security that is found in the lockSecurity.mqsc script	4-7
Section 2: Apply the security changes to MQG1 by running the lockSecurity.mqsc script	4-10
4.3. Check the environment after the security changes, identify the cause of the 2035, and implement correct adjustments	4-12
Section 1: Repeat the sending of messages from MQE5 to the WM253.IN queues via queue remote TO.GATEWAY at MQE5	4-12
Section 2: Check whether the messages reached the WM253.IN queues	4-12
Section 3: Check the status of sender channel MQE5.MQG1	4-13
Section 4: Attempt a runmqsc session for queue manager MQG1 after the security commands were applied	4-15
Section 5: Create a channel authentication rule at MQG1 to allow MQE5 to use channel MQE5.MQG1	4-17
Section 6: Determine the behavior of the cluster after MQG1 has hardened security	4-19
Section 7: Confirm the reason that the WMADMCLS.MQG1 channel is in retry by checking the MQG1 queue manager error log	4-21
Section 8: Create a channel authorization record to allow connectivity with the cluster queue managers	4-22
Section 9: Check that the new rule resolved connectivity issues in the other cluster member queue managers	4-23
4.4. Part 3: Work with basic object authorizations	4-26
Section 1: Use dspmqaut to display the authorizations of a user for a queue in a queue manager with optional connection authentication	4-26
Section 2: Use DIS AUTHREC to display the authorizations of an unprivileged user for a queue in a queue manager with optional connection authentication	4-26
Section 3: Use dspmqaut to display the authorization of user fileusr1 to connect to queue manager MQG1, which has the required connection authentication	4-28
Section 4: Grant user fileusr1 connect authority to queue manager MQG1	4-29
Section 5: Check the results of the setmqaut by using <code>dis authrec</code> from a <code>runmqsc</code> session	4-29
Exercise 5. Working with workload balancing options	5-1
Introduction	1
Requirements	2
5.1. End any active queue managers	5-4
5.2. Defining the cluster queue managers, channels, and queues	5-5
5.3. Using a round-robin scheme for workload balancing	5-10
5.4. Using channel and queue rank to control workload	5-12
5.5. Using channel priority to control workload	5-14
5.6. Using channel WEIGHT to control workload	5-15
5.7. Restricting the number of outbound cluster channels	5-16
Exercise 6. Configuring a publish/subscribe cluster	6-1
Requirements	2
6.1. Stop any active queue managers	6-4
6.2. Clustered publish/subscribe with direct routing	6-5
Section 1: Create and verify the queue managers and the cluster that is used in this exercise	6-6
Section 2: Create and review a cluster topic	6-8

Section 3: Use IBM MQ Explorer as an extra way to verify the cluster topic	6-11
Section 4: Test the publish/subscribe cluster	6-11
6.3. Testing cluster publication routing	6-13
Section 1: Create a subscription and publication	6-13
Section 2: View the message route	6-14
Section 3: Validate proxy subscriptions	6-17
6.4. Clustered publish/subscribe with topic host routing	6-18
Section 1: Replace the existing DIRECT routed topic on QMPS2 with a TOPICHOST routed topic	6-19
Section 2: Test the topic host routing by creating a test subscription on QMPS4 in IBM MQ Explorer	6-20
6.5. Exercise clean-up	6-25
Exercise 7. Configuring an overlapping cluster	7-1
Requirements	2
7.1. Creating an overlapping cluster	7-4
Section 1: Ensure that the correct queue managers are active for this exercise, and any other queue managers are stopped	7-5
Section 2: Create the CLUS2 queue managers QMCA and QMCB	7-5
Section 3: Verify the resulting CLUS1 and CLUS2 overlapping cluster from the perspective of full repository queue manager QMC1	7-6
Section 4: Add the queue ACCOUNTING to the CLUS2 queue managers QMCA and QMCB ...	7-8
7.2. Putting messages across the clusters	7-10
Section 1: Create a queue that is shared by using the cluster namelist and test across clusters .	7-10
Section 2: Create a queue alias to put messages across the two clusters and test the definitions	7-10
7.3. Exercise clean-up	7-12

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:

CICS®
Notes®
z/OS®

DB2®
RACF®

IMS™
WebSphere®

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.

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

PAR® is a trademark or registered trademark of Merge Healthcare, an IBM Company.

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

Exercises description

This course includes the following exercises:

- Configuring and reviewing base IBM MQ resources
- Implementing and verifying a cluster, cluster queues, and a cluster gateway
- Working with cluster administration tasks
- Working with IBM MQ security
- Working with workload balancing options
- Configuring a publish/subscribe cluster
- Configuring an overlapping cluster

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.

Safety practices

IBM General Safety practices must be adhered to while at the education center. Use safety glasses, hearing protection, and remove personal effects that might create a hazardous environment. If it cannot be readily done before entering areas with potential safety hazards, remove all exposed jewelry before traveling to the education center.



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. Configuring and reviewing base IBM MQ resources

Estimated time

01:00

Overview

In this exercise, you define and start the queue managers that are used in the cluster implementation lab. You also establish distributed two-way connectivity between two of the queue managers in the cluster in preparation for setting up a cluster gateway.

Objectives

After completing this exercise, you should be able to:

- Define and start IBM MQ queue managers
- Establish two-way distributed message channels between two queue managers
- Examine the channel status
- Locate the IBM MQ queue manager logs and the dead-letter queue
- Describe the queue manage cluster repository process

Requirements

IBM MQ V9 server installed.

Exercise instructions

Preface

In this lab exercise, you create the queue managers that you use to create your first cluster in Exercise 2. You also establish distributed message channel connectivity between two queue managers: MQE5, which is used as an external queue manager, and MQG1, which is used as a cluster member gateway queue manager.

Section 1: Log on to the lab image and review environment


- ___ 1. Use the directions that are provided to you to reach the lab image.
- ___ 2. Log on with `Administrator` as the user ID and `webSphere` as the password.



Attention

If you see a gray panel that requests activation of the Windows license, close it by clicking the **Cancel** button at the lower-right corner of the panel. If your VMware screen is not fully expanded, you might not see the **Cancel** button.

If you log off your VMware image, you might encounter this panel. If it surfaces again, click the **Cancel** button.

- ___ 3. Open a command prompt window by clicking the command prompt  icon on your Windows taskbar.
- ___ 4. Display the installed IBM MQ software by typing `dspmqrinst` and pressing the Enter key. Your results should resemble the display in the text box.

```
dspmqrinst
```

Expected results:

```
InstName:      IBMMQV9
InstDesc:
Identifier:    1
InstPath:      C:\Program Files\IBM\MQ
Version:       9.0.0.0
Primary:       Yes
State:         Available
MSIProdCode:   {80B9C730-0533-4060-A44B-16549476F111}
MSIMedia:      9.0 Server
MSIInstanceId: 1
```

- ___ 5. Check for any defined queue managers by typing `dspmq` and pressing the Enter key. The result is expected to be a blank line with no output.

- ___ 6. **Leave the command prompt window open through the end of this lab exercise.**

Section 2: Create the queue managers that are used in Exercise 2



Note

Throughout the steps in this section, you can type the commands as shown in the instructions, or copy and paste the command from the `Lab1_copyAndPaste.txt` file under the `C:\LabFiles\Unit1` directory.

Do not copy the commands directly from this exercise guide, as you might introduce special characters that cause errors in the commands.

- ___ 7. Create queue manager MQG1 to listen on port 1651 by typing the command as shown in the text box, and press the Enter key. Optionally, you can copy and paste the command that is located under the respective section name in the `Lab1_copyAndPaste.txt` file that is found in directory `C:\LabFiles\Unit1`.

```
crtmqm -p 1651 -u MQG1.DLQ MQG1
```

Expected results:

IBM MQ queue manager created.

Directory 'C:\ProgramData\IBM\MQ\qmgrs\MQG1' created.

The queue manager is associated with installation 'IBMMQV9'.

Creating or replacing default objects for queue manager 'MQG1'.

Default objects statistics : 87 created. 0 replaced. 0 failed.

Completing setup.

Setup completed.

- ___ 8. Create the queue managers MQ03, MQ0A, MQ0B, and MQE5 with the attributes that are provided in the text box by following the same process that you used for the first queue manager.

For each queue manager created, ensure a successful completion similar to the results that are shown in the box for queue manager MQG1.

```
crtmqm -p 1653 -u MQ03.DLQ MQ03
```

```
crtmqm -p 1657 -u MQ0A.DLQ MQ0A
```

```
crtmqm -p 1658 -u MQ0B.DLQ MQ0B
```

```
crtmqm -p 1661 -u MQE5.DLQ MQE5
```

Section 3: Start and display the queue managers

___ 9. Start the queue managers.

- ___ a. Start queue manager MQG1 by typing `strmqm MQG1` and pressing the Enter key. Expected results are displayed in the text box.

```
IBM MQ queue manager 'MQG1' starting.
The queue manager is associated with installation 'IBMMQV9'.
5 log records accessed on queue manager 'MQG1' during the log replay
phase.
Log replay for queue manager 'MQG1' complete.
Transaction manager state recovered for queue manager 'MQG1'.
IBM MQ queue manager 'MQG1' started using V9.0.0.0.
```

- ___ b. Start queue manager MQ03 by typing `strmqm MQ03` and pressing the Enter key.
Ensure that the results resemble the results that are shown for the start of queue manager MQG1.
- ___ c. Start queue manager MQ0A by typing `strmqm MQ0A` and pressing the Enter key.
Ensure that the results resemble the results that are shown for the start of queue manager MQG1.
- ___ d. Start queue manager MQ0B by typing `strmqm MQ0B` and pressing the Enter key.
Ensure that the results resemble the results that are shown for the start of queue manager MQG1.
- ___ e. Start queue manager MQE5 by typing `strmqm MQE5` and pressing the Enter key.
Ensure that the results resemble the results that are shown for the start of queue manager MQG1.
- ___ 10. Check that all queue managers are running by typing `dspmq` and pressing the Enter key. The results should resemble the display in the text box.

```
dspmq
```

Expected results:

QMNAME(MQG1)	STATUS(Running)
QMNAME(MQ03)	STATUS(Running)
QMNAME(MQ0A)	STATUS(Running)
QMNAME(MQ0B)	STATUS(Running)
QMNAME(MQE5)	STATUS(Running)

- ___ 11. If your results do not resemble the display, resolve any problems before you proceed with the lab.
- ___ a. First, ensure that you did type the `strmqm` command for any non-running queue managers by scrolling back in your command prompt window and reviewing the `strmqm` commands issued.

- ___ b. If you find that you typed a `strmqm` command for the non-running queue manager, start by reviewing the queue manager log at directory
`C:\ProgramData\IBM\MQ\qmgrs\<queueManagerName>\errors>` for any queue manager that failed to start. Replace the `<queueManagerName>` placeholder with the name of the queue manager that exhibits a non-running status.

**Note**

In a later lab exercise:

- Queue manager MQE5 serves as an “external” queue manager.
- Queue manager MQG1 plays the role of a cluster queue manager that is used in the role of “gateway queue manager”.

In this section, you establish **basic connectivity** between the two queue managers. However, you add other definitions later to complete the “gateway queue manager” functionality.

**CAUTION**

Do not forget to define the dead-letter queue that is identified for each queue manager. In this course, the dead-letter queues are added before the queue managers are used. For queue managers MQE5 and MQG1, the dead-letter queues are defined in this exercise. For queue managers MQ0A and MQ0B, the dead-letter queues are created with the cluster definition scripts.

Section 4: Review the object definitions and changes that are required for queue manager MQE5 and run the script by using the `runmqsc` utility

- ___ 12. The IBM MQ script commands in file `MQE5.mqsc` are copied in the text box for this step. Review these definitions.
- ___ a. You see how connection authentication and connection authorization are disabled.
 - ___ b. You also see standard sender-channel definitions from queue manager MQE5 to queue manager MQG1, and some test queues.
 - ___ c. The transmission queue that is used for the sender channel is set to be triggered.

Do not make any changes to the script.

```

*****
* WM253 Exercise 1 student script
* Queue manager name: MQE5
*****

ALTER QMGR +
    CHLAUTH(DISABLED) +
    CLWLUSEQ(LOCAL) +
    CONNAUTH('SYSTEM.DEFAULT.AUTHINFO.IDPWOS') +
    DEADQ('MQE5.DLQ') +
    FORCE

alter authinfo(SYSTEM.DEFAULT.AUTHINFO.IDPWOS) authtype(IDPWOS)
chkckloc1(optional) chkckclnt(optional)
refresh security type(connauth)
DEFINE QLOCAL('MQE5.DLQ') +
    REPLACE
DEFINE QLOCAL('MQE5.IN') +
    MAXDEPTH(5000) +
    REPLACE
DEFINE QREMOTE('TO.MQG1') +
    RQMNAME('MQG1') +
    RNAME('MQG1.IN') +
    XMITQ('MQG1') +
    REPLACE
DEFINE QLOCAL('MQG1') +
    INITQ('SYSTEM.CHANNEL.INITQ') +
    MAXDEPTH(5000) +
    TRIGGER +
    TRIGDATA('MQE5.MQG1') +
    USAGE(XMITQ) +
    REPLACE
DEFINE CHANNEL('MQE5.MQG1') +
    CHLTYPE(SDR) +
    CONNAME('localhost(1655)') +
    TRPTYPE(TCP) +
    XMITQ('MQG1') +
    REPLACE
DEFINE CHANNEL('MQG1.MQE5') +
    CHLTYPE(RCVR) +
    TRPTYPE(TCP) +
    REPLACE
*****

```

**Stop**

In these instructions, queue manager connection authentication is set to optional, and channel authentication is disabled. These changes are done to mitigate security distractions during the setup of the basic cluster. **However, disabling connection authentication and channel authentication in a queue manager is not to be construed as a preferred practice.**

-
- ___ 13. At the command prompt window, change to directory `C:\LabFiles\Unit1` by typing the command that is shown in the text box, and press the Enter key.
-

`cd C:\LabFiles\Unit1`

Expected results:

Your command-line prompt now shows `C:\LabFiles\Unit1>`

- ___ 14. Ensure that the expected command files are present by listing the directory by typing the command that is shown in the text box, and press the Enter key. Look for two `.mqsc` suffixed files: `MQE5.mqsc` and `MQG1.mqsc`.
-

`dir`

Expected results (abbreviated display with non-essential notations removed):

```
Directory of C:\LabFiles\Unit1
04/27/2017  02:39 AM                902 Lab1_copyAndPaste.txt
04/27/2017  02:35 AM             1,163 MQE5.mqsc
04/27/2017  02:35 AM             1,218 MQG1.mqsc
```

- ___ 15. Process the `MQE5.mqsc` script by typing the command as shown on the text box, and press the Enter key. Ensure that you capture the results in a file so that you can review that each command completed successfully. Ensure that you are in the `C:\LabFiles\Unit1>` directory before you type the command.
-

`runmqsc MQE5 < MQE5.mqsc > MQE5.out`

All expected output goes to file `MQE5.out`.

- ___ 16. From the same directory in the command prompt window, check the `runmqsc` utility results:
- ___ a. Open the `MQE5.out` file by typing `notepad MQE5.out` and pressing the Enter key.
 - ___ b. Scroll to the end of the output. You should see:


```
9 MQSC commands read.
No commands have a syntax error.
All valid MQSC commands were processed
```
 - ___ c. If the commands process without errors, continue to the next numbered step.
-

- ___ d. If the results are not as expected, review the output and correct any errors before you proceed.

Close file MQE5.out.

Section 5: Determine the value of the BIND attribute in the queues that were created



CAUTION

Applications can use MQOPEN options to alter the MQOPEN behavior. If no open options are used, the DEFBIND attribute of the local queue is used.

If you define a local queue and allow the queue attributes to use the defaults, the DEFBIND attribute from the SYSTEM.LOCAL.DEFAULT.QUEUE is used. DEFBIND for the SYSTEM.LOCAL.DEFAULT.QUEUE is set to DEFBIND(OPEN). Use of DEFBIND(OPEN) causes affinities and impacts the cluster workload management.

Keep in mind the results of the display that is obtained in this section when you need to identify and eliminate message affinities.

If you define cluster queues, ensure that you specify the NOTFIXED value for the queue DEFBIND attribute.

- ___ 17. Open a `runmqsc` session for queue manager MQE5 by typing `runmqsc MQE5` and pressing the Enter key.
- ___ 18. In the previous section, you created three local queues: MQE5.DLQ, MQE5.IN, and MQG1. Display the DEFBIND attribute for all three queues by typing `dis q(M*) defbind` and pressing the Enter key. The expected results are:

```
dis q(M*) defbind
  1 : dis q(M*) defbind
AMQ8409: Display Queue details.
      QUEUE(MQE5.DLQ)                                TYPE(QLOCAL)
      DEFBIND(OPEN)
AMQ8409: Display Queue details.
      QUEUE(MQE5.IN)                                  TYPE(QLOCAL)
      DEFBIND(OPEN)
AMQ8409: Display Queue details.
      QUEUE(MQG1)                                     TYPE(QLOCAL)
      DEFBIND(OPEN) TYPE(QLOCAL)
```

- ___ 19. Observe how, when attribute values are not specified in a queue definition, the default value for the DEFBIND attribute might lead to message affinities.
- ___ 20. Type `end` and press the Enter key to exit the `runmqsc` session.

Section 6: Run the script to create the object definitions and changes that are required for queue manager MQG1 by using the *runmqsc* utility

- ___ 21. Ensure that you are still at directory `C:\LabFiles\Unit1` in your command prompt script so that you have access to the `MQG1.mqsc` script.
- ___ 22. The `MQG1.mqsc` script that is used to do the required additions and updates to queue manager MQG1 is similar to the script that you reviewed for queue manager MQE5. The difference is that the definitions are made from MQE5 to MQG1 and the objects are created for MQE5. At your option, you can review the `MQG1.mqsc` script.
- ___ 23. Process the `MQG1.mqsc` script by typing the command as shown on the text box, and press the Enter key. Ensure that you capture the results in a file so that you can verify that each command completed successfully. Ensure that you are in the `C:\LabFiles\Unit1>` directory before you type the command.

```
runmqsc MQG1 < MQG1.mqsc > MQG1.out
```

All expected output goes to file `MQG1.out`.

- ___ 24. From the same directory in the command prompt window, check the *runmqsc* utility results:
 - ___ a. Open the `MQG1.out` file by typing `notepad MQG1.out` and pressing the Enter key.
 - ___ b. Scroll to the end of the output. You should see:


```
9 MQSC commands read.
No commands have a syntax error.
All valid MQSC commands were processed
```
 - ___ c. If you see that the commands are processed and have no errors, continue to the next numbered step.
 - ___ d. If the results are not as expected, review the output and correct any errors before you proceed.
- ___ 25. Close file `MQG1.out`.

Section 7: Test connectivity from queue managers MQE5 to MQG1

- ___ 26. Start a *runmqsc* session for queue manager MQE5 by typing `runmqsc MQE5` and pressing the Enter key.
- ___ 27. **Do not close the *runmqsc* session until explicitly instructed.**
- ___ 28. Channel MQE5.MQG1 has the transmission queue set to be trigger-started. However, you must start the channel one time to get it out of stopped status before the trigger-start of the channel works. Start the channel by typing `start chl(MQE5.MQG1)` from the *runmqsc* session, and press the Enter key. The expected response is:

```
1 : start chl(MQE5.MQG1)
AMQ8018: Start IBM MQ channel accepted.
```

- ___ 29. Display the channel status to confirm whether the channel started by typing the command `dis chs(MQ*)` and pressing the Enter key. The expected response is:

```
dis chs(MQ*)
      2 : dis chs(MQ*)
AMQ8417: Display Channel Status details.
CHANNEL(MQE5.MQG1)                CHLTYPE(SDR)
CONNAME(127.0.0.1(1651))           CURRENT
RQMNAME(MQG1)                     STATUS(RUNNING)
SUBSTATE(MQGET)                   XMITQ(MQG1)
```

- ___ 30. Close the `runmqsc` session by typing `end` and pressing the Enter key.
- ___ 31. Use the `amqspout` sample program to send a test message from MQE5 to MQG1 by typing the command as indicated in the text box, and press the Enter key **one time**. If you press Enter twice, you end the sample without sending the message.

```
amqspout TO.MQG1 MQE5
```

Expected results:

```
Sample AMQSPUT0 start
target queue is TO.MQG1
```

- ___ 32. Type any data such as `abcd` and press the Enter key two times to end `amqspout`. Expected results are:

```
abcd
```

```
Sample AMQSPUT0 end
```

- ___ 33. Check that the data was received at queue MQG1.IN of queue manager MQG1 by using the `amqsget` command as shown in the display.

```
amqsget MQG1.IN MQG1
```

Expected results (if you typed `abcd` for data):

```
Sample AMQSGET0 start
message <abcd>
```

- ___ 34. Sample `amqsget` runs for 15 seconds and then stops. After you see your message, you can either wait until it stops, or press CTL-C to break out of the command. If you wait for `amqsget` to stop, the results are as shown.

```
no more messages
Sample AMQSGET0 end
```

- ___ 35. If your message was sent from MQE5 and arrived at the MQG1.IN queue at queue manager MQG1, proceed to the next step. If the message did not arrive, investigate and resolve the problem before you proceed.

Section 8: Test connectivity from queue managers MQG1 to MQE5

- ___ 36. Start a `runmqsc` session for queue manager MQG1 by typing `runmqsc MQG1` and pressing the Enter key.

- ___ 37. **Do not close the `runmqsc` session until explicitly instructed.**
- ___ 38. Channel MQG1.MQE5 has the transmission queue set to be trigger-started. However, you must start the channel one time to get it out of stopped status before the trigger-start of the channel works. Start the channel by typing `start chl(MQG1.MQE5)` from the `runmqsc` session, and press the Enter key. Expected response is:
- ```
1 : start chl(MQG1.MQE5)
AMQ8018: Start IBM MQ channel accepted.
```
- \_\_\_ 39. Display the channel status to confirm whether the channel started by typing the command `dis chs(MQ*)` and pressing the Enter key. The expected response is:
- ```
dis chs(MQ*)
  2 : dis chs(MQ*)
AMQ8417: Display Channel Status details.
CHANNEL(MQE5.MQG1)                CHLTYPE(RCVR)
CONNNAME(127.0.0.1)                CURRENT
RQMNAME(MQE5)                      STATUS(RUNNING)
SUBSTATE(RECEIVE)
AMQ8417: Display Channel Status details.
CHANNEL(MQG1.MQE5)                CHLTYPE(SDR)
CONNNAME(127.0.0.1(1661))          CURRENT
RQMNAME(MQE5)                      STATUS(RUNNING)
SUBSTATE(MQGET)
```
- ___ 40. For the channel status display from queue manager MQG1, you now see that two channels are running, the MQG1 receiver MQE5.MQG1, and the MQG1 sender MQG1.MQE5. **If your results do not show that the channels are running, investigate and resolve the problems before you proceed.**
- ___ 41. Close the `runmqsc` session by typing `end` and pressing the Enter key.
- ___ 42. Use the `amqspout` sample program to send a test message from MQE5 to MQG1 by typing the command as indicated in the text box, and press the Enter key **one time**. If you press Enter twice, you end the sample without sending the message.

```
amqspout TO.MQE5 MQG1
```

Expected results:

```
Sample AMQSPUT0 start
target queue is TO.MQE5
```

- ___ 43. Type any data such as `xyzd` and press the Enter key two times to end `amqspout`. Expected results are:

```
xyzd
```

```
Sample AMQSPUT0 end
```

- ___ 44. Check that the data was received at queue MQE5.IN of queue manager MQE5 by using the `amqsget` command as shown in the display.

```
amqsget MQE5.IN MQE5
```

Expected results (if you typed `xyzd` for data):

```
Sample AMQSGET0 start
message <xyzd>
```

Sample `amqsget` runs for 15 seconds and then stops. After you see your message, you can either wait until it stops, or press CTL-C to break out of the command.

Section 9: Review the queue manager logs for queue manager MQG1

- ___ 45. From the command prompt, navigate to the MQG1 queue manager error log by typing `cd C:\ProgramData\IBM\MQ\qmgrs\MQG1\errors` and pressing the Enter key.
- ___ 46. Use Notepad to open the `AMQERR01.LOG` file by typing `notepad AMQERR01.LOG` and pressing the Enter key.
- ___ 47. Ensure that you are at the start of the file.
- ___ 48. Confirm whether the cluster repository manager started. From the Notepad file, select **Edit > Find**.
- ___ 49. Type `repository` in the Find box, and press the Find button. The expected result is displayed in the text box.

```
-----
4/27/2017 01:01:50 - Process(1568.1) User(MUSR_MQADMIN)
Program(amqrrmfa.exe)
                        Host(WS2008R2X64) Installation(IBM MQV9)
                        VRMF(9.0.0.0) QMgr(MQG1)
```

```
AMQ9410: Repository manager started.
```

```
EXPLANATION:
```

```
The repository manager started successfully.
```

```
ACTION:
```

```
None.
```

- ___ 50. A log entry that matches the search for `repository` surfaces. However, if you do not locate the word “repository” the first time, you might need to repeat the search. You might continue to see a “Not found” response when in actuality the entry does exist in the log. **Leave the queue log file open.**

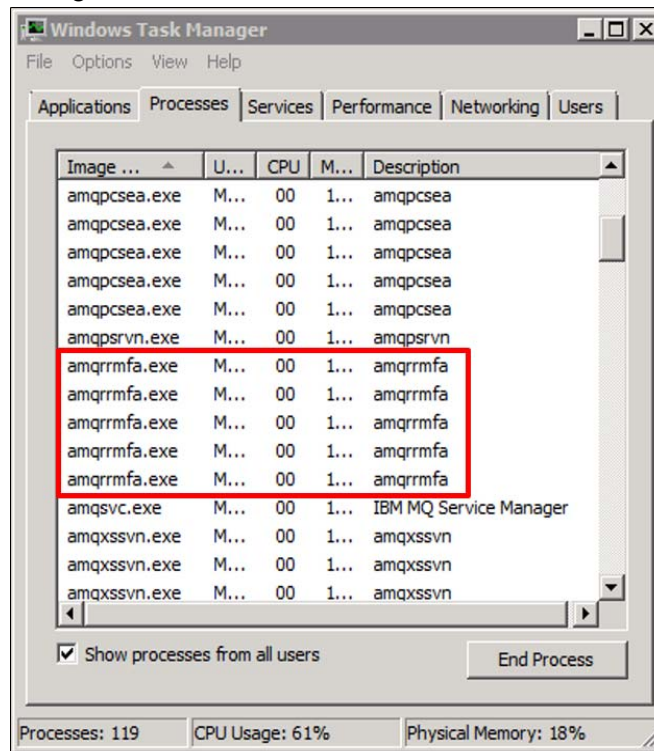


Note

In the log entry that was found when you searched for “repository”, take note of the program name, `amqrrmfa.exe`. This program is the cluster repository manager for distributed platforms. In a z/OS queue manager, the cluster repository manager starts as part of the channel initiator started task.

Message `AMQ9410` confirms that the repository manager is running for queue manager `MQG1`.

- ___ 51. You created five queue managers. Check the Windows task manager to confirm whether you see that five `amqrrmfa.exe` processes are running by right-clicking the Windows taskbar and selecting **Task Manager**.
- ___ 52. Select the Windows Task Manager **Processes** tab and scroll down until you find the instances of the `amqrrmfa.exe` process. Five instances of the process are expected, one for each queue manager.



Troubleshooting

Make a note of the `amqrrmfa.exe` process name. When you work with a clustered environment, if “odd” behaviors are happening, yet no apparent problems exist, the repository manager process might be stopped. Changes made to active cluster channels might bring down the repository manager.

To find the process in a UNIX environment, type `ps -ef | grep amqrrmfa` from a command window.

To determine whether the repository manager is running in z/OS platforms, you can check the IBM MQ channel initiator started task for the following message:

CSQX410I MQ03 CSQXREPO **Repository manager** started.

-
- __ 53. Close or minimize the Task Manager window.
 - __ 54. Return to the queue manager log and find information about channel MQE5.MQG1 by repeating another search for the channel name, MQE5.MQG1. The expected result is partially displayed in the text box.

```
-----
4/27/2017 03:02:10 - Process(5216.3) User(MUSR_MQADMIN) Program(amqrmppa.exe)
                    Host(WS2008R2X64) Installation(IBM MQV9)
                    VRMF(9.0.0.0) QMgr(MQG1)
```

AMQ9002: Channel 'MQE5.MQG1' is starting.

EXPLANATION:

Channel 'MQE5.MQG1' is starting.

ACTION:

None.

```
-----
4/27/2017 03:04:30 - Process(5016.1) User(MUSR_MQADMIN) Program(runmqchl.exe)
                    Host(WS2008R2X64) Installation(IBM MQV9)
                    VRMF(9.0.0.0) QMgr(MQG1)
```

AMQ9002: Channel 'MQG1.MQE5' is starting.

EXPLANATION:

Channel 'MQG1.MQE5' is starting.

ACTION:

None.

-
- __ 55. Close the queue manager log file.



Note

When you work with distributed message channels, you review the queue manager log if you suspect a channel problem or need to check what happened with a channel. In a later lab exercise, you discover that when you work with IBM MQ clusters, although you might use other cluster commands to display information, you also go to the queue manager logs. You use these logs to obtain other details about the cluster channels. The only difference is that in some situations, you might be checking logs for more than one queue manager.

End of exercise

Exercise summary

In this exercise, you:

- Created and started the queue managers to be used in the first cluster definition.
- Reviewed defaulted queue attributes which, if not handled properly for cluster queues, might introduce affinities that disrupt the cluster workload balancing.
- Started selected message channels and confirmed that the channels were running.
- Verified that messages flow between two queue managers to be used as the base for an external queue manager and a cluster queue manager gateway in a later exercise.
- Examine the channel status.

Overall in this exercise, you took a first look at some cluster considerations and details as you worked through the tasks of the exercise.

Before you configure the first cluster, the next unit teaches key facts to consider before you embark on a cluster implementation, whether as an IBM MQ administrator, an architect, or a developer.

Exercise 2. Implementing and verifying a cluster, cluster queues, and a cluster gateway

Estimated time

01:30

Overview

In this exercise, you implement a basic cluster. As you incorporate extra queue managers, you verify that each addition is successful by using the DIS CLUSQMGR command incrementally. As you review the displayed information, you learn the differences in the dynamically generated CLUSSDRA or CLUSSDRB channels. You also learn how to implement a cluster gateway, and how to do a simple workload distribution baseline.

Objectives

After completing this exercise, you should be able to:

- Describe the MQSC commands that are used to create an IBM MQ cluster
- Implement an IBM MQ cluster by using MQSC commands
- Use the DIS CLUSQMGR command to verify a new cluster by identifying the channel types, status, and repository type
- Create cluster queues
- Identify problems that are found in the cluster
- Observe the cluster “need-to-know basis” behavior by using the DIS QCLUSTER command from a partial configuration queue manager before any messages are exchanged
- Configure and test a cluster gateway
- Send messages from a queue manager external to the cluster, to a cluster gateway queue manager
- Implement baseline default message workload distribution

Requirements

- Completion of Exercise 1 with queue managers MQG1, MQ0A, MQ0B, MQ05, and MQ0E started
- Two-way channels between queue managers MQ0E and MQG1 defined

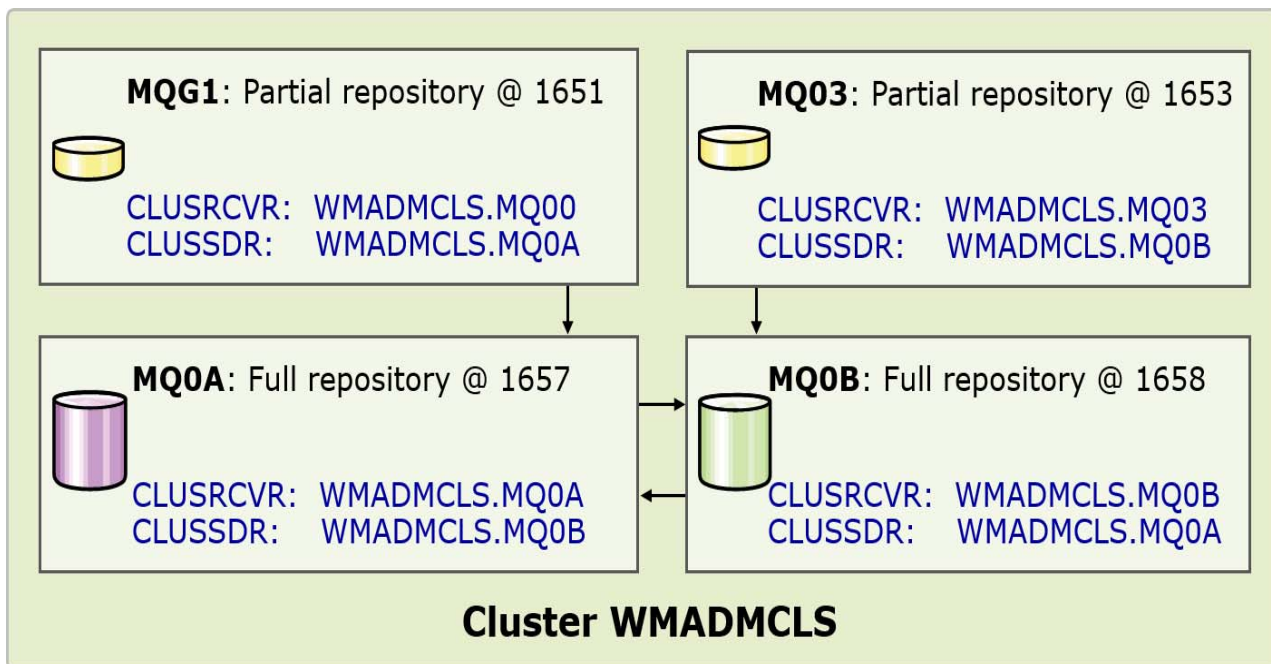
- Connection authorization must be set to optional for local and client connections, and channel authentication must be disabled

Exercise instructions

Preface

The graphic in this page denotes the cluster organization that you are about to define. The process that you follow is:

- Update queue manager MQ0A as needed to make it a full repository cluster member.
- Update queue manager MQ0A as needed to make it the second full repository cluster member.
- Use the `dis clusqmgr` command to verify that the base configuration was successful.
- Add the MQG1 queue manager as a partial repository cluster member. Point the CLUSSDR channel for the MQG1 queue manager to the MQ0A full repository.
- Use the `dis clusqmgr` command to verify that the incorporation of queue manager MQG1 to the cluster was successful.
- Add the MQ03 queue manager as a partial repository cluster member. Point the CLUSSDR channel for the MQ03 queue manager to the MQ0B full repository.
- Use the `dis clusqmgr` command to verify that the incorporation of queue manager MQG1 to the cluster was successful.



2.1. Configure and verify cluster WMADMCLS

- ___ 1. If you logged off the lab environment, follow the instructions to reach your image and log on by using `Administrator` as the user ID and `weblsphere` as the password.
- ___ 2. Open a command prompt window. Leave the command prompt window open until explicitly instructed to close it. You use this window to create and verify the cluster configuration.
- ___ 3. Type `cd C:\LabFiles\Unit3` and press the Enter key to navigate to the directory where the “copy-and-paste” commands are stored. *Note: `Unit3` is correct. Although this exercise is the second lab exercise in this course, the lab is for Unit 3.*

Section 1: Review and configure the commands that are required to add queue manager MQ0A as a full repository of cluster WMADMCLS

- ___ 4. Review the contents of the definitions from file `1_MQ0A.txt` at `C:\LabFiles\Unit3`, and the comments that continue after the display.

1+1=2 Example

The contents of script file 1_MQ0A.txt are shown in the example display.

```
*****
* 1_MQ0A.txt start
*****

* Define the dead-letter queue for MQ0A
DEF QL(MQ0A.DLQ) REPLACE

* Make MQ0A a full repository queue manager and adjust message length
ALTER QMGR REPOS(WMADMCLS) MAXMSGL(32768)

* Adjust the max message length of the system and dead-letter queues.
ALTER QLOCAL(SYSTEM.DEFAULT.LOCAL.QUEUE) MAXMSGL(32768)
ALTER QLOCAL(SYSTEM.CLUSTER.TRANSMIT.QUEUE) MAXMSGL(32768)
ALTER QLOCAL(MQ0A.DLQ) MAXMSGL(32768)

* Define the CLUSRCVR channel
DEF CHL(WMADMCLS.MQ0A) CHLTYPE(CLUSRCVR) TRPTYPE(TCP) +
  CONNAME('localhost(1657)') MAXMSGL(32768) CLUSTER(WMADMCLS) REPLACE

* Define the CLUSSDR channel to the second full repos queue manager, MQ0B
DEF CHL(WMADMCLS.MQ0B) CHLTYPE(CLUSSDR) TRPTYPE(TCP) +
  CONNAME('localhost(1658)') MAXMSGL(32768) CLUSTER(WMADMCLS) REPLACE

*****
* 1_MQ0A.txt end
*****
```

- ___ a. When you created queue manager MQ0A in Exercise 1, MQ0A.DLQ was specified as the queue to use as the dead-letter queue. You define QLOCAL MQ0A.DLQ now. It is not directly related to the cluster, but must be defined.
- ___ b. The ALTER QMGR command changes two fields. First, REPOS(WMADMCLS) makes queue manager MQ0A a full repository. Second is the adjustment of the maximum length of messages that are allowed in the queue manager, or MAXMSGL.

! Important

Always confirm the maximum size of messages that are expected to travel through the cluster and adjust all SYSTEM queues, cluster queues, and the dead-letter queue to reflect the largest expected message size.

Larger messages require adequate capacity, and might impact performance expectations. Always question the need for larger messages.

In this lab exercise, the default size is being decreased rather than increased just to make the point and raise awareness that along with the cluster implementation, message length changes might be needed.

-
- ___ c. The third group of ALTER QL commands adjusts the message sizes for cluster-related system queues and the dead-letter queue.
 - ___ d. The first DEF CHL command defines the cluster receiver command, which provides connectivity information for the MQ0A queue manager. Other cluster queue manager members create dynamic CLUSSDRA or CLUSSDRB channels to this CLUSRCVR channel.
 - ___ e. Last is the CLUSSDR channel definition to the second full cluster repository queue manager, MQ0B. When queue manager MQ0A defines a dynamic sender channel to queue manager MQ0B, it is a CLUSSDRB type channel. If queue manager MQ0A defines a dynamic sender channel to any other queue manager in the cluster other than MQ0B, it is a CLUSSDRA channel.
- ___ 5. From the command prompt window, ensure that you are at the `C:\LabFiles\Unit3` directory.
 - ___ 6. Implement the additions and changes in the script by using a `runmqsc` session by typing the command as shown in the text box, and press the Enter key.
-

```
runmqsc MQ0A < 1_MQ0A.txt > MQ0A.out
```

- ___ 7. Open output file `MQ0A.out` to confirm that the commands completed successfully by typing `notepad MQ0A.out` and pressing the Enter key.
 - ___ 8. Scroll to the end of the file and review the results. The expected result is displayed in the text box.
-

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

- ___ 9. ***If your results are different from the results that are displayed in the text box***, review the rest of the `MQ0A.out` file to determine the problem, and make corrections before you continue.
- ___ 10. If you see that seven commands were read, and no commands have errors, you can continue with the next section.
- ___ 11. Close the `MQ0A.out` file.

Section 2: Review and configure the commands that are required to add queue manager MQ0B as a full repository of cluster WMADMCLS

- ___ 12. Review the contents of the definitions from file `2_MQ0B.txt` at `C:\LabFiles\Unit3`, and the comments that continue after the display.

1+1=2 Example

The contents of script file 2_MQ0B.txt are shown in the example display.

```
*****
* 2_MQ0B.txt start
*****

* Define the dead-letter queue for MQ0B
DEF QL(MQ0B.DLQ) REPLACE

* Make MQ0B a full repository queue manager and adjust the maximum length
of messages allowed
ALTER QMGR REPOS(WMADMCLS) MAXMSGL(32768)

* Adjust the queue manager and queue sizes. Normally this is needed when
messages are larger than the default length.
* The changes are implemented in this exercise exclusively to make the
point that you must take
* potential size of messages traveling through the cluster into
consideration.
* Larger message sizes impact queue manager resources and must be
questioned and documented.
ALTER QLOCAL(SYSTEM.DEFAULT.LOCAL.QUEUE) MAXMSGL(32768)
ALTER QLOCAL(SYSTEM.CLUSTER.TRANSMIT.QUEUE) MAXMSGL(32768)
ALTER QLOCAL(MQ0B.DLQ) MAXMSGL(32768)

* Define the CLUSRCVR channel
DEF CHL(WMADMCLS.MQ0B) CHLTYPE(CLUSRCVR) TRPTYPE(TCP) +
  CONNAME('localhost(1658)') MAXMSGL(32768) CLUSTER(WMADMCLS) REPLACE

* Define the CLUSSDR channel to the other full repos queue manager, MQ0A
DEF CHL(WMADMCLS.MQ0A) CHLTYPE(CLUSSDR) TRPTYPE(TCP) +
  CONNAME('localhost(1657)') MAXMSGL(32768) CLUSTER(WMADMCLS) REPLACE

*****
* 2_MQ0B.txt end
*****
```

- ___ 13. In your review of the definitions that were made in the MQ0B queue manager, you find:
- ___ a. The creation of the dead-letter queue, changes to the queue manager, and adjustment of the queue sizes are similar to the commands that ran for queue manager MQ0B.
 - ___ b. The CLUSRCVR channel now points to port 1658, where queue manager MQ0B listens.
 - ___ c. The CLUSSDR channel in MQ0B points to the second full repository queue manager, MQ0A.

- ___ 14. From the command prompt window, ensure that you are at the `C:\LabFiles\Unit3` directory.
- ___ 15. Implement the additions and changes in the script by using a `runmqsc` session by typing the command as shown in the text box, and press the Enter key.

```
runmqsc MQ0B < 2_MQ0B.txt > MQ0B.out
```

- ___ 16. Open output file `MQ0B.out` to confirm that the commands completed successfully by typing `notepad MQ0B.out` and pressing the Enter key.
- ___ 17. Scroll to the end of the file and review the results. The expected result is displayed in the text box.

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

- ___ 18. **If your results are different from the results that are displayed in the text box**, review the rest of the `MQ0B.out` file to determine the problem, and make corrections before you continue.
- ___ 19. If you see that seven commands were read, and no commands have errors, you can continue with the next section.
- ___ 20. Close the `MQ0B.out` file.

Section 3: Run `DIS CLUSQMGR` command from each `MQ0A` and `MQ0B` to confirm the success of the first part of the configuration



Hint

The output of the `DIS CLUSQMGR(*) ALL` command spans more than one command prompt screen. Type the command from outside the `runmqsc` session and direct the output to a file to optimize viewing.

-
- ___ 21. From the command prompt window, type the command as shown in the text box, and press the Enter key. Optionally you can copy and paste the command from the `Lab2_copyAndPaste.txt` file that is found in the `Unit3` directory.
-

```
echo dis clusqmgr(*) all | runmqsc MQ0A > disA.txt
```

- ___ 22. Open file `disA.txt` to review the results of the display by typing `notepad disA.txt` and pressing the Enter key.

**Important**

Your results are expected to resemble the display in this lab guide **except** for date and time information, and other fields that the queue manager generates dynamically, such as the QMID attributes.

__ 23. Review the notes that directly follow the display.

```

1 : dis clusqmgr(*) all
AMQ8441: Display Cluster Queue Manager details.

CLUSQMGR(MQ0A)
ALTIME(09.35.50)
BATCHINT(0)
BATCHSZ(50)
CLUSDATE(2017-04-28)
CLUSTIME(09.35.50)
CLWLRANK(0)
COMPHDR(NONE)
CONNAME(localhost(1657))
DEFTYPE(CLUSRCVR)
DISCINT(6000)
KAINI(AUTO)
LONGRTY(999999999)
MAXMSGL(32768)
MCATYPE(THREAD)
MODENAME( )
MREXIT( )
MRIMR(1000)
MSGEXIT( )
NPMSPEED(FAST)
PROPCTL(COMPAT)
QMID(MQ0A_2017-04-27_00.52.51)
RCVDATA( )
SCYDATA( )
SENDDATA( )
SEQWRAP(999999999)
SHORTIMR(60)
SSLCIPH( )
STATUS(RUNNING)
TPNAME( )
USEDLQ(YES)
VERSION(09000000)

ALTDAT(2017-04-28)
BATCHHB(0)
BATCHLIM(5000)
CHANNEL(WMADMCLS.MQ0A)
CLUSTER(WMADMCLS)
CLWLPRTY(0)
CLWLWGHT(50)
COMPMSG(NONE)
CONVERT(NO)
DESCR( )
HBINT(300)
LOCLADDR( )
LONGIMR(1200)
MCANAME( )
MCAUSER( )
MRDATA( )
MRRTY(10)
MSGDATA( )
NETPRTY(0)
PASSWORD( )
PUTAUT(DEF)
QMTYPE(REPOS)
RCVEXIT( )
SCYEXIT( )
SENDEXIT( )
SHORTRTY(10)
SSLCAUTH(REQUIRED)
SSLPEER( )
SUSPEND(NO)
TRPTYPE(TCP)
USERID( )
XMITQ( )

```


AMQ8441: Display Cluster Queue Manager details.

CLUSQMGR(MQ0B)	ALTDATE(2017-04-28)
ALTTIME(09.43.17)	BATCHHB(0)
BATCHINT(0)	BATCHLIM(5000)
BATCHSZ(50)	CHANNEL(WMADMCLS.MQ0B)
CLUSDATE(2017-04-28)	CLUSTER(WMADMCLS)
CLUSTIME(09.43.17)	CLWLPRTY(0)
CLWLRANK(0)	CLWLWGHT(50)
COMPHDR(NONE)	COMPMSG(NONE)
CONNNAME(localhost(1658))	CONVERT(NO)
DEFTYPE(CLUSSDRB)	DESCR()
DISCINT(6000)	HBINT(300)
KAINT(AUTO)	LOCLADDR()
LONGRTY(999999999)	LONGTMR(1200)
MAXMSGL(32768)	MCANAME()
MCATYPE(THREAD)	MCAUSER()
MODENAME()	MRDATA()
MREXIT()	MRRTY(10)
MRTMR(1000)	MSGDATA()
MSGEXIT()	NETPRTY(0)
NPMSPEED(FAST)	PASSWORD()
PROPCTL(COMPAT)	PUTAUT(DEF)
QMID(MQ0B_2017-04-27_00.52.52)	QMTYPE(REPOS)
RCVDATA()	RCVEXIT()
SCYDATA()	SCYEXIT()
SENDDATA()	SENDEXIT()
SEQWRAP(999999999)	SHORTRTY(10)
SHORTTMR(60)	SSLCAUTH(REQUIRED)
SSLCIPH()	SSLPEER()
STATUS(RUNNING)	SUSPEND(NO)
TPNAME()	TRPTYPE(TCP)
USEDLQ(YES)	USERID()
VERSION(09000000)	

XMITQ(SYSTEM.CLUSTER.TRANSMIT.QUEUE)

One MQSC command read.

No commands have a syntax error.

All valid MQSC commands were processed.

-
- ___ a. Two **AMQ8441** cluster queue manager details entries exist, the first entry for queue manager MQ0A, the second entry for queue manager MQ0B. The display shows the cluster view *from the perspective of the MQ0A queue manager*.

- ___ b. Look at the fields that are marked in bold. The first entry for CLUSQMGR(MQ0A)
- Shows that it is for the CLUSRCVR channel type **WMADMCLS.MQ0A**
 - Confirms that MQ0A is a full repository with QMTYPE(REPOS)
 - Shows that it is in running status.
 - The transmission queue, or XMITQ attribute is empty because it is a “receiver” type channel.
- ___ c. Look at the fields that are marked in bold for the second entry for CLUSQMGR(MQ0B).
- You see that it is for a CLUSSDRB type channel **WMADMCLS.MQ0B** to the MQ0B second full repository queue manager.
 - CLUSSDRB is a dynamic channel from MQ0A to MQ0B, The “B” suffix in CLUSSDRB indicates that this channel is a dynamic channel, but queue manager MQ0A also has an explicit CLUSSDR channel definition to queue manager MQ0B. The “B” denotes that it is **both** a dynamic and explicitly defined channel.
 - You see that queue manager MQ0B is also a full repository queue manager, with QMTYPE(REPOS).
 - Look at the difference in the XMITQ attribute. Since a CLUSSDRB channel is a “sender” type, it indicates that it uses the cluster default transmit queue.
- ___ d. You are **not** expected to find any **SYSTEM.TEMPQMGR** prefixed names in the CLUSQMGR attribute or any **SYSTEM.TEMPUUID** prefixed names in the QMID attribute of any queue manager entries. If you find any **SYSTEM.TEMP*** entries as described, investigate and correct the error.
- ___ 24. Close file `disA.txt`.
- ___ 25. From the command prompt window, type the command as shown in the text box, and press the Enter key. Optionally you can copy and paste the command from the `Lab2_copyAndPaste.txt` file that is found in the `Unit3` directory.

```
echo dis clusqmgr(*) all | runmqsc MQ0B > disB.txt
```

- ___ 26. Open file `disB.txt` to review the results of the display by typing `notepad disB.txt` and pressing the Enter key.
- ___ 27. Review the notes that directly follow the expected display. You now look at the cluster from the perspective of the MQ0B queue manager.

```
1 : dis clusqmgr(*) all
```

AMQ8441: Display Cluster Queue Manager details.

CLUSQMGR(MQ0A)	ALTDATE(2017-04-28)
ALTTIME(09.35.50)	BATCHHB(0)
BATCHINT(0)	BATCHLIM(5000)
BATCHSZ(50)	CHANNEL(WMADMCLS.MQ0A)
CLUSDATE(2017-04-28)	CLUSTER(WMADMCLS)
CLUSTIME(09.38.50)	CLWLPRTY(0)
CLWLRANK(0)	CLWLWGHT(50)
COMPHDR(NONE)	COMPMSG(NONE)
CONNAME(localhost(1657))	CONVERT(NO)
DEFTYPE(CLUSSDRB)	DESCR()
DISCINT(6000)	HBINT(300)
KAINT(AUTO)	LOCLADDR()
LONGRTY(999999999)	LONGTMR(1200)
MAXMSGL(32768)	MCANAME()
MCAATYPE(THREAD)	MCAUSER()
MODENAME()	MRDATA()
MREXIT()	MRRTY(10)
MRTMR(1000)	MSGDATA()
MSGEXIT()	NETPRTY(0)
NPMSPEED(FAST)	PASSWORD()
PROPCTL(COMPAT)	PUTAUT(DEF)
QMID(MQ0A_2017-04-27_00.52.51)	QMTYPE(REPOS)
RCVDATA()	RCVEXIT()
SCYDATA()	SCYEXIT()
SENDDATA()	SENDEXIT()
SEQWRAP(999999999)	SHORTRTY(10)
SHORTTMR(60)	SSLCAUTH(REQUIRED)
SSLCIPH()	SSLPEER()
STATUS(RUNNING)	SUSPEND(NO)
TPNAME()	TRPTYPE(TCP)
USEDLQ(YES)	USERID()
VERSION(09000000)	

XMITQ(SYSTEM.CLUSTER.TRANSMIT.QUEUE)

AMQ8441: Display Cluster Queue Manager details.

CLUSQMGR(MQ0B)	ALTDATE(2017-04-28)
ALTTIME(09.43.17)	BATCHHB(0)
BATCHINT(0)	BATCHLIM(5000)
BATCHSZ(50)	CHANNEL(WMADMCLS.MQ0B)
CLUSDATE(2017-04-28)	CLUSTER(WMADMCLS)
CLUSTIME(09.43.17)	CLWLPRTY(0)
CLWLRANK(0)	CLWLWGHT(50)
COMPHDR(NONE)	COMPMSG(NONE)
CONNAME(localhost(1658))	CONVERT(NO)
DEFTYPE(CLUSRCVR)	DESCR()

```

DISCINT(6000)
Kaint(AUTO)
LONGRTY(999999999)
MAXMSGL(32768)
MCATYPE(THREAD)
MODENAME( )
MREXIT( )
MRIMR(1000)
MSGEXIT( )
NPMSPEED(FAST)
PROPCTL(COMPAT)
QMID(MQ0B_2017-04-27_00.52.52)
RCVDATA( )
SCYDATA( )
SENDDATA( )
SEQWRAP(999999999)
SHORTIMR(60)
SSLCIPH( )
STATUS(RUNNING)
TPNAME( )
USEDLQ(YES)
VERSION(09000000)

HBINT(300)
LOCLADDR( )
LONGTMR(1200)
MCANAME( )
MCAUSER( )
MRDATA( )
MRRTY(10)
MSGDATA( )
NETPRTY(0)
PASSWORD( )
PUTAUT(DEF)
QMTYPE(REPOS)
RCVEXIT( )
SCYEXIT( )
SENDEXIT( )
SHORTRTY(10)
SSLCAUTH(REQUIRED)
SSLPEER( )
SUSPEND(NO)
TRPTYPE(TCP)
USERID( )
XMITQ( )

```

One MQSC command read.
No commands have a syntax error.
All valid MQSC commands were processed.

-
- ___ a. The observations of the DIS CLUSQMGR display from queue manager MQ0B are similar to the display from queue manager MQ0A. The exception is that in this MQ0B display, the CLUSSDRB type channel to MQ0A appears before the MQ0B CLUSRCVR channel.
- ___ b. Both channels are in running status.
- ___ c. The WMADMCLS.MQ0A channel to MQ0A is a dynamic CLUSSDRB channel because queue manager MQ0B has an explicit CLUSSDR channel definition to the first full repository queue manager, MQ0A. As you add other queue managers, you start to see some CLUSSDR dynamic channels, but not expected now with two queue managers in the cluster, that point the explicitly defined CLUSSDR channel to each other.
- ___ 28. Close the disB.txt file.
-



Note

As you add each extra queue manager to the cluster, you run two DIS CLUSQMGR commands to verify each new queue manager:

- First display from the new queue manager

- Second display from the full repository queue manager that the new queue manager points its CLUSSDR channel to.
-

Section 4: Review and run the commands that are required to incorporate queue manager MQG1 to cluster WMADMCLS

- ___ 29. Review the contents of the definitions from file 3_MQG1.txt at C:\LabFiles\Unit3.
- ___ a. One difference in the script file to add MQG1 is that the ALTER QMGR command is not there because queue manager MQG1 does not hold a full repository.
 - ___ b. The message length in the dead-letter queue is handled at the DEFINE QL definition.
 - ___ c. The CLUSSDR channel for queue manager MQG1 points to full repository queue manager MQ0A.

1+1=2 Example

The contents of script file 3_MQG1.txt are shown in the example display.

```
*****
* 3_MQG1.txt start
*****

* Define the dead-letter queue for MQG1
DEF QL(MQG1.DLQ) MAXMSGL(32768) REPLACE

* MQG1 does not hold a full repository, no need for an ALTER QMGR command

* Adjust the queue manager and queue sizes. Normally this is needed when
messages are larger than the default length.
* The changes are implemented in this exercise exclusively to make the
point that you must take
* potential size of messages traveling through the cluster into
consideration.
* Larger message sizes impact queue manager resources and must be
questioned and documented.
ALTER QLOCAL(SYSTEM.DEFAULT.LOCAL.QUEUE) MAXMSGL(32768)
ALTER QLOCAL(SYSTEM.CLUSTER.TRANSMIT.QUEUE) MAXMSGL(32768)
ALTER QLOCAL(MQG1.DLQ) MAXMSGL(32768)

* Define the CLUSRCVR channel
DEF CHL(WMADMCLS.MQG1) CHLTYPE(CLUSRCVR) TRPTYPE(TCP) +
  CONNAME('localhost(1653)') MAXMSGL(32768) CLUSTER(WMADMCLS) REPLACE

* Define the CLUSSDR channel to the other full repos queue manager, MQ0A
DEF CHL(WMADMCLS.MQ0A) CHLTYPE(CLUSSDR) TRPTYPE(TCP) +
  CONNAME('localhost(1657)') MAXMSGL(32768) cluster(WMADMCLS) REPLACE

*****
* 3_MQG1.txt end
*****
```

-
- ___ 30. From the command prompt window, ensure that you are at the C:\LabFiles\Unit3 directory.
- ___ 31. Implement the additions and changes in the script by using a runmqsc session by typing the command as shown in the text box, and press the Enter key.
-

```
runmqsc MQG1 < 3_MQG1.txt > MQG1.out
```

- ___ 32. Open output file `MQG1.out` to confirm that the commands completed successfully by typing `notepad MQG1.out` and pressing the Enter key.
- ___ 33. Scroll to the end of the file and review the results. The expected result is displayed in the text box.

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

- ___ 34. ***If your results are different from the results that are displayed in the text box***, review the rest of the `MQG1.out` file to determine the problem, and make corrections before you continue.
- ___ 35. If you see that six commands were read, and no commands have errors, you can continue with the next section.
- ___ 36. Close the `MQG1.out` file.

Section 5: Run the DIS CLUSQMGR command from MQG1 to verify a successful incorporation of queue manager MQG1 to the cluster

- ___ 37. From the command prompt window, type the command as shown in the text box, and press the Enter key. Optionally you can copy and paste the command from the `Lab2_copyAndPaste.txt` file that is found in the `Unit3` directory.

```
echo dis clusqmgr(*) all | runmqsc MQG1 > disG.txt
```

- ___ 38. Open file `disG.txt` to review the results of the display by typing `notepad disG.txt` and pressing the Enter key.



Important

Your results are expected to resemble the display in this lab guide except for date and time information, and other fields that the queue manager generates dynamically, such as the QMID attributes. However, the different types of channels might display in a different order than the display shown in the text box.

-
- ___ 39. Review the notes that directly follow the display.

```

1 : dis clusqmgr(*) all
AMQ8441: Display Cluster Queue Manager details.
  CLUSQMGR(MQ0A)                                ALTDAT(2017-04-28)
  ALTIME(09.35.50)                              BATCHHB(0)
  BATCHINT(0)                                    BATCHLIM(5000)
  BATCHSZ(50)                                    CHANNEL(WMADMCLS.MQ0A)
  CLUSDATE(2017-04-28)                          CLUSTER(WMADMCLS)
  CLUSTIME(12.27.32)                             CLWLPRTY(0)
  CLWLRANK(0)                                    CLWLWGHT(50)
  COMPHDR(NONE)                                  COMPMSG(NONE)
  CONNAME(localhost(1657))                      CONVERT(NO)
  DEFTYPE(CLUSSDRB)                             DESCR( )
  DISCINT(6000)                                 HBINT(300)
  KAIN(TAUTO)                                   LOCLADDR( )
  LONGRTY(999999999)                           LONGTMR(1200)
  MAXMSGL(32768)                               MCANAME( )
  MCATYPE(THREAD)                             MCAUSER( )
  MODENAME( )                                  MRDATA( )
  MREXIT( )                                    MRRTY(10)
  MRTMR(1000)                                  MSGDATA( )
  MSGEXIT( )                                   NETPRTY(0)
  NPMSPEED(FAST)                              PASSWORD( )
  PROPCTL(COMPAT)                             PUTAUT(DEF)
  QMID(MQ0A_2017-04-27_00.52.51)              QMTYPE(REPOS)
  RCVDATA( )                                   RCVEXIT( )
  SCYDATA( )                                   SCYEXIT( )
  SENDDATA( )                                  SENDEXIT( )
  SEQWRAP(999999999)                          SHORTRTY(10)
  SHORTTMR(60)                                SSLCAUTH(REQUIRED)
  SSLCIPH( )                                   SSLPEER( )
  STATUS(RUNNING)                             SUSPEND(NO)
  TPNAME( )                                    TRPTYPE(TCP)
  USEDLO(YES)                                  USERID( )
  VERSION(09000000)

```

XMITQ(SYSTEM.CLUSTER.TRANSMIT.QUEUE)

```

AMQ8441: Display Cluster Queue Manager details.
  CLUSQMGR(MQ0B)                                ALTDAT(2017-04-28)
  ALTIME(09.43.17)                              BATCHHB(0)
  BATCHINT(0)                                    BATCHLIM(5000)
  BATCHSZ(50)                                    CHANNEL(WMADMCLS.MQ0B)
  CLUSDATE(2017-04-28)                          CLUSTER(WMADMCLS)
  CLUSTIME(12.27.32)                             CLWLPRTY(0)
  CLWLRANK(0)                                    CLWLWGHT(50)
  COMPHDR(NONE)                                  COMPMSG(NONE)
  CONNAME(localhost(1658))                      CONVERT(NO)
  DEFTYPE(CLUSSDRA)                             DESCR( )
  DISCINT(6000)                                 HBINT(300)

```


KAINT(AUTO)	LOCLADDR()
LONGRTY(999999999)	LONGTMR(1200)
MAXMSGL(32768)	MCANAME()
MCATYPE(THREAD)	MCAUSER()
MODENAME()	MRDATA()
MREXIT()	MRRTY(10)
MRTMR(1000)	MSGDATA()
MSGEXIT()	NETPRTY(0)
NPMSPEED(FAST)	PASSWORD()
PROPCTL(COMPAT)	PUTAUT(DEF)
QMID(MQ0B_2017-04-27_00.52.52)	QMTYPE(REPOS)
RCVDATA()	RCVEXIT()
SCYDATA()	SCYEXIT()
SENDDATA()	SENDEXIT()
SEQWRAP(999999999)	SHORTRTY(10)
SHORTTMR(60)	SSLCAUTH(REQUIRED)
SSLCIPH()	SSLPEER()
STATUS(RUNNING)	SUSPEND(NO)
TPNAME()	TRPTYPE(TCP)
USEDLQ(YES)	USERID()
VERSION(09000000)	
XMITQ(SYSTEM.CLUSTER.TRANSMIT.QUEUE)	
AMQ8441: Display Cluster Queue Manager details.	
CLUSQMGR(MQG1)	ALTDATE(2017-04-28)
ALTTIME(12.27.32)	BATCHHB(0)
BATCHINT(0)	BATCHLIM(5000)
BATCHSZ(50)	CHANNEL(WMADMCLS.MQG1)
CLUSDATE(2017-04-28)	CLUSTER(WMADMCLS)
CLUSTIME(12.27.32)	CLWLPRTY(0)
CLWLRANK(0)	CLWLWGHT(50)
COMPHDR(NONE)	COMPMSG(NONE)
CONNAME(localhost(1651))	CONVERT(NO)
DEFTYPE(CLUSRCVR)	DESCR()
DISCINT(6000)	HBINT(300)
KAINT(AUTO)	LOCLADDR()
LONGRTY(999999999)	LONGTMR(1200)
MAXMSGL(32768)	MCANAME()
MCATYPE(THREAD)	MCAUSER()
MODENAME()	MRDATA()
MREXIT()	MRRTY(10)
MRTMR(1000)	MSGDATA()
MSGEXIT()	NETPRTY(0)
NPMSPEED(FAST)	PASSWORD()
PROPCTL(COMPAT)	PUTAUT(DEF)
QMID(MQG1_2017-04-27_00.45.50)	QMTYPE(NORMAL)
RCVDATA()	RCVEXIT()
SCYDATA()	SCYEXIT()
SENDDATA()	SENDEXIT()

```

SEQWRAP(999999999)
SHORTIMR(60)
SSLCIPH( )
STATUS(RUNNING)
TPNAME( )
USEDLQ(YES)
VERSION(09000000)

SHORTTRY(10)
SSLCAUTH(REQUIRED)
SSLPEER( )
SUSPEND(NO)
TRPTYPE(TCP)
USERID( )
XMITQ( )

```

One MQSC command read.
No commands have a syntax error.
All valid MQSC commands were processed.

You are now viewing the cluster information from the perspective of queue manager MQG1. You find three entries in the display.

- ___ a. The first entry is for channel `WMADMCLS.MQ0A` to queue manager MQ0A, which is the queue manager to which the full repository queue manager that the CLUSSDR channel for this MQG1 queue manager points. Since MQG1 has an explicit CLUSSDR definition to queue manager MQ0A, the dynamic channel to MQ0A is of type CLUSSDRB because it is both a dynamically and an explicitly defined channel. You are now familiar with the other fields. Ensure that the status is running, and then check the next entry.
 - ___ b. The second entry for channel `WMADMCLS.MQ0B` to queue manager MQG1 is a type CLUSSDRA channel. Therefore, queue manager MQG1 creates a dynamic CLUSSDR channel to queue manager MQ0B, but MQG1 does not have an explicit CLUSSDR definition to MQ0B, which makes the CLUSSDR type "A", or CLUSSDRA. All other attributes are familiar. Ensure that the channel is in running status.
 - ___ c. The last entry is the CLUSRCVR channel for MQG1, `WMADMCLS.MQ0B`. The results are similar to the other CLUSRCVR displays with one exception, the queue manager type, or QMTYPE. This entry is the first entry for a partial repository queue manager, and you see a QMTYPE(NORMAL), which denotes that MQG1 does not hold a full repository.
- ___ 40. Close file `disG.txt`.

Section 6: Review and configure the commands that are required to incorporate queue manager MQ03 to cluster WMADMCLS

- ___ 41. Review the contents of the definitions from file `4_MQ03.txt` at `C:\LabFiles\Unit3`.
 - ___ a. One difference in the script file to add MQG1 is that no `ALTER QMGR` command is there because queue manager MQG1 does not hold a full repository.
 - ___ b. The message length in the dead-letter queue is handled at the `DEFINE QL` definition.
 - ___ c. The CLUSSDR channel for queue manager MQG1 points to full repository queue manager MQ0A.

1+1=2 Example

The contents of script file 4_MQ03.txt are shown in the example display.

```
*****
* 4_MQ03.txt start
*****

* Define the dead-letter queue for MQ03
DEF QL(MQ03.DLQ) REPLACE

* Make MQ03 a full repository queue manager and adjust the maximum length
of messages allowed
ALTER QMGR REPOS(WMADMCLS) MAXMSGL(32768)

* Adjust the queue manager and queue sizes. Normally this is needed when
messages are larger than the default length.
* The changes are implemented in this exercise exclusively to make the
point that you must take
* potential size of messages traveling through the cluster into
consideration.
* Larger message sizes impact queue manager resources and must be
questioned and documented.
ALTER QLOCAL(SYSTEM.DEFAULT.LOCAL.QUEUE) MAXMSGL(32768)
ALTER QLOCAL(SYSTEM.CLUSTER.TRANSMIT.QUEUE) MAXMSGL(32768)
ALTER QLOCAL(MQ03.DLQ) MAXMSGL(32768)

* Define the CLUSRCVR channel
DEF CHL(WMADMCLS.MQ03) CHLTYPE(CLUSRCVR) TRPTYPE(TCP) +
  CONNAME('localhost(1657)') MAXMSGL(32768) CLUSTER(WMADMCLS) REPLACE

* Define the CLUSSDR channel to the other full repos queue manager, MQ0B
DEF CHL(WMADMCLS.MQ0B) CHLTYPE(CLUSSDR) TRPTYPE(TCP) +
  CONNAME('localhost(1658)') MAXMSGL(32768) cluster(WMADMCLS) REPLACE

*****
* 4_MQ03.txt end
*****
```

-
- __ 42. From the command prompt window, ensure that you are at the C:\LabFiles\Unit3 directory.
- __ 43. Implement the additions and changes in the script by using a runmqsc session by typing the command as shown in the text box, and press the Enter key.
-

```
runmqsc MQ03 < 4_MQ03.txt > MQ03.out
```

- ___ 44. Open output file `MQ03.out` to confirm that the commands completed successfully by typing `notepad MQ03.out` and pressing the Enter key.
- ___ 45. Scroll to the end of the file and review the results. The expected result is displayed in the text box.

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

- ___ 46. ***If your results are different from the results that are displayed in the text box***, review the rest of the `MQ03.out` file to determine the problem, and make corrections before you continue.
- ___ 47. If you see that six commands were read, and no commands have errors, you can continue with the next section.
- ___ 48. Close the `MQ03.out` file.

Section 7: Run DIS CLUSQMGR command from MQ03 to verify a successful incorporation of queue manager MQ03 to the cluster

- ___ 49. From the command prompt window, type the command as shown in the text box, and press the Enter key. Optionally you can copy and paste the command from the `Lab2_copyAndPaste.txt` file that is found in the `Unit3` directory.

```
echo dis clusqmgr(*) all | runmqsc MQ03 > dis3.txt
```

- ___ 50. Open file `dis3.txt` to review the results of the display by typing `notepad disG.txt` and pressing the Enter key.



Important

Your results are expected to resemble the display in this lab guide except for date and time information, and other fields that the queue manager generates dynamically, such as the QMID attributes. However, the different types of channels might display in a different order than the display shown in the text box.

-
- ___ 51. Review the notes that directly follow the display.

```
1 : dis clusqmgr(*) all
```

```
AMQ8441: Display Cluster Queue Manager details.
```

CLUSQMR(MQ03)	ALTDATE(2017-04-28)
ALTIME(16.29.39)	BATCHHB(0)
BATCHINT(0)	BATCHLIM(5000)
BATCHSZ(50)	CHANNEL(WMADMCLS.MQ03)
CLUSDATE(2017-04-28)	CLUSTER(WMADMCLS)
CLUSTIME(16.23.34)	CLWLPRTY(0)
CLWLRANK(0)	CLWLWGHT(50)
COMPHDR(NONE)	COMPMSG(NONE)
CONNNAME(localhost(1653))	CONVERT(NO)
DEFTYPE(CLUSRCVR)	DESCR()
DISCINT(6000)	HBINT(300)
KAINT(AUTO)	LOCLADDR()
LONGRTY(999999999)	LONGTMR(1200)
MAXMSGL(32768)	MCANAME()
MCATYPE(THREAD)	MCAUSER()
MODENAME()	MRDATA()
MREXIT()	MRRTY(10)
MRTMR(1000)	MSGDATA()
MSGEXIT()	NETPRTY(0)
NPMSPEED(FAST)	PASSWORD()
PROPCTL(COMPAT)	PUTAUT(DEF)
QMID(MQ03_2017-04-27_00.51.51)	QMTYPE(NORMAL)
RCVDATA()	RCVEXIT()
SCYDATA()	SCYEXIT()
SENDDATA()	SENDEXIT()
SEQWRAP(999999999)	SHORTRTY(10)
SHORTTMR(60)	SSLCAUTH(REQUIRED)
SSLCIPH()	SSLPEER()
STATUS(RUNNING)	SUSPEND(NO)
TPNAME()	TRPTYPE(TCP)
USEDLQ(YES)	USERID()
VERSION(09000000)	XMITQ()

```
AMQ8441: Display Cluster Queue Manager details.
```

CLUSQMR(MQ0A)	ALTDATE(2017-04-28)
ALTIME(09.35.50)	BATCHHB(0)
BATCHINT(0)	BATCHLIM(5000)
BATCHSZ(50)	CHANNEL(WMADMCLS.MQ0A)
CLUSDATE(2017-04-28)	CLUSTER(WMADMCLS)
CLUSTIME(16.48.51)	CLWLPRTY(0)
CLWLRANK(0)	CLWLWGHT(50)
COMPHDR(NONE)	COMPMSG(NONE)
CONNNAME(localhost(1657))	CONVERT(NO)
DEFTYPE(CLUSSDRA)	DESCR()
DISCINT(6000)	HBINT(300)

KAIN(T(AUTO)	LOCLADDR()
LONGRTY(999999999)	LONGTMR(1200)
MAXMSGL(32768)	MCANAME()
MCATYPE(THREAD)	MCAUSER()
MODENAME()	MRDATA()
MREXIT()	MRRTY(10)
MRIMR(1000)	MSGDATA()
MSGEXIT()	NETPRTY(0)
NPMSPEED(FAST)	PASSWORD()
PROPCTL(COMPAT)	PUTAUT(DEF)
QMID(MQ0A_2017-04-27_00.52.51)	QMTYPE(REPOS)
RCVDATA()	RCVEXIT()
SCYDATA()	SCYEXIT()
SENDDATA()	SENDEXIT()
SEQWRAP(999999999)	SHORTRTY(10)
SHORTTMR(60)	SSLCAUTH(REQUIRED)
SSLCIPH()	SSLPEER()
STATUS(RUNNING)	SUSPEND(NO)
TPNAME()	TRPTYPE(TCP)
USEDLQ(YES)	USERID()
VERSION(09000000)	

XMITQ(SYSTEM.CLUSTER.TRANSMIT.QUEUE)

AMQ8441: Display Cluster Queue Manager details.

CLUSQMGR(MQ0B)	ALTDATE(2017-04-28)
ALTTIME(09.43.17)	BATCHHB(0)
BATCHINT(0)	BATCHLIM(5000)
BATCHSZ(50)	CHANNEL(WMADMCLS.MQ0B)
CLUSDATE(2017-04-28)	CLUSTER(WMADMCLS)
CLUSTIME(16.30.34)	CLWLPRTY(0)
CLWLRANK(0)	CLWLWGHT(50)
COMPHDR(NONE)	COMPMSG(NONE)
CONNAME(localhost(1658))	CONVERT(NO)
DEFTYPE(CLUSSDRB)	DESCR()
DISCINT(6000)	HBINT(300)
KAIN(T(AUTO)	LOCLADDR()
LONGRTY(999999999)	LONGTMR(1200)
MAXMSGL(32768)	MCANAME()
MCATYPE(THREAD)	MCAUSER()
MODENAME()	MRDATA()
MREXIT()	MRRTY(10)
MRIMR(1000)	MSGDATA()
MSGEXIT()	NETPRTY(0)
NPMSPEED(FAST)	PASSWORD()
PROPCTL(COMPAT)	PUTAUT(DEF)
QMID(MQ0B_2017-04-27_00.52.52)	QMTYPE(REPOS)
RCVDATA()	RCVEXIT()
SCYDATA()	SCYEXIT()
SENDDATA()	SENDEXIT()

```

SEQWRAP(999999999)
SHORTIMR(60)
SSLCIPH( )
STATUS(RUNNING)
TPNAME( )
USEDLQ(YES)
VERSION(09000000)

SHORTTRY(10)
SSLCAUTH(REQUIRED)
SSLPEER( )
SUSPEND(NO)
TRPTYPE(TCP)
USERID( )

```

XMITQ(SYSTEM.CLUSTER.TRANSMIT.QUEUE)

One MQSC command read.

No commands have a syntax error.

All valid MQSC commands were processed.

You are now viewing the cluster information from the perspective of queue manager MQ03. You find three entries in the display.

- ___ a. The first entry is for channel `WMADMCLS.MQ03`, which is the `CLUSRCVR` channel for the MQ03 queue manager. The channel is in running status.
 - ___ b. In the second entry, you see dynamic channel `WMADMCLS.MQ0A`, which is a `CLUSSDRA` to MQ0A. Queue manager MQ03 does not have an explicit `CLUSSDR` definition to queue manager MQ0A; therefore, the dynamic channel is of type `CLUSSDRA`. The channel is in running status.
 - ___ c. The last entry is for dynamic channel `WMADMCLS.MQ0B`, which is a dynamic channel to MQ0B. MQ03 has **both** a dynamic and an explicit definition to the MQ0A full repository; therefore, the dynamic channel is of type `CLUSDRB`.
- ___ 52. Close file `dis3.txt`.

2.2. Define and display cluster queues

Section 1: Define a cluster queue by the same name in queue managers MQ0A and MQ03

- ___ 53. Type `runmqsc MQ0A` and press the Enter key to start a `runmqsc` session for queue manager MQ0A.
- ___ 54. Define queue WM253.IN by typing the command as shown in the example, and press the Enter key. Optionally you can copy and paste the command from the `Lab2_copyAndPaste.txt` file that is found in the `Unit3` directory.



Example

```
def ql(WM253.IN) cluster(WMADMCLS) defbind(NOTFIXED)
```

- ___ 55. Ensure that the completion message confirms that the queue was created. If you have any syntax errors, make the corrections and ensure that the queue is created successfully before you continue with any other steps.
- ___ 56. Type `end` and press the Enter key to close the `runmqsc` session for queue manager MQ0A.
- ___ 57. Type `runmqsc MQ03` and press the Enter key to start a `runmqsc` session for queue manager MQ03.
- ___ 58. Define queue WM253.IN by typing the same `def ql` command from the example text box, and press the Enter key. Optionally you can copy and paste the command from the `Lab2_copyAndPaste.txt` file that is found in the `Unit3` directory.
- ___ 59. Ensure that the completion message confirms that the queue was created. If you have any syntax errors, make the corrections and ensure that the queue is created successfully before you continue with any other steps.
- ___ 60. Type `end` and press the Enter key to close the `runmqsc` session for queue manager MQ03.
- ___ 61. ***Do not attempt to place any messages to these queues at this time.***

Section 2: Attempt to display the new cluster queues from cluster member queue manager MQG1



Note

This test introduces the “need-to-know” behavior with cluster queues. ***If the queues that you attempt to display are not found now, do not be concerned. You are not expected to find them now.***

- ___ 62. From the command prompt window, start a `runmqsc` session for queue manager MQG1 by typing `runmqsc MQG1` and pressing the Enter key. Ensure that you use MQG1 to start the session.
 - ___ 63. Display the cluster queues by typing `dis qcluster(*)` and pressing the Enter key.
-



Stop

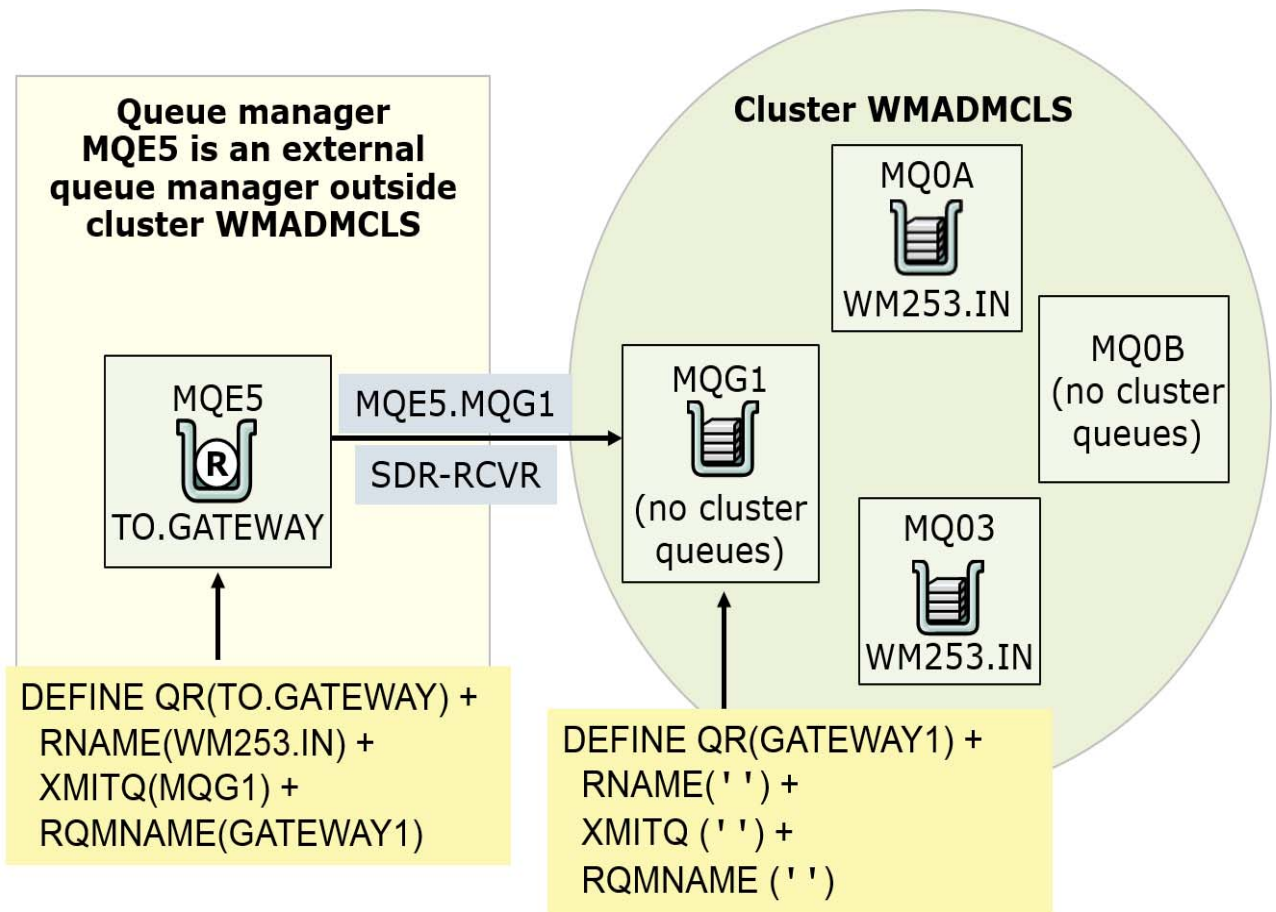
If your results indicate that the object was not found, ***you obtained the expected response***. You just observed the cluster normal “need-to-know” behavior. You repeat this test after you use these queues from queue manager MQG1 later in this lab exercise.

However, if before you reached this step you did any work where the MQG1 “needed to know” about these queues, then you might find them in the display. The step works if you did not do any work.

Whether the queues show or not is not significant to continue this exercise. You can continue regardless of outcome.

- ___ 64. Type `end` and press the Enter key to close the `runmqsc` session for queue manager MQG1.

2.3. Configure the cluster gateway and run a workload balancing baseline



Note

In Exercise 1, you configured distributed message channels between queue managers MQE5 and MQG1. Later, queue manager MQG1 was added to cluster WMADMCLS. In this section, you work through steps to:

- Set up queue manager MQG1 to be used as a cluster gateway named GATEWAY1.
- Create remote queue TO.GATEWAY on queue manager MQE5 to use GATEWAY1 to send messages to the WM253.IN cluster queues.
- Observe that remote queue TO.GATEWAY uses the MQE5.MQG1 sender-receiver channels you created and tested in Exercise 1.

In your workload balance baseline, you put messages to queue TO.GATEWAY in queue manager MQE5.

Section 1: Create required definitions in non-cluster queue manager MQE5

- ___ 65. Type `runmqsc MQE5` and press the Enter key to start a `runmqsc` session for queue manager MQE5.
- ___ 66. Create remote queue TO.GATEWAY by typing the definition command as shown in the text box, and press the Enter key. **Type the command in the same line before you press the Enter key.** Optionally you can copy and paste the command from the `Lab2_copyAndPaste.txt` file that is found in the `Unit3` directory.

```
DEFINE QR(TO.GATEWAY) RNAME(WM253.IN)XMITQ(MQG1) RQMNAME(GATEWAY1)
```

Expected response:

```
AMQ8006: IBM MQ queue created.
```

- ___ 67. Ensure that the queue was created successfully.
- ___ 68. Type `end` and press the Enter key to close the `runmqsc` session for queue manager MQE5.

Section 2: Create required definitions to make cluster queue manager MQG1 the gateway queue manager to cluster WMADMCLS

- ___ 69. Type `runmqsc MQG1` and press the Enter key to start a `runmqsc` session for queue manager MQG1.
- ___ 70. Create the gateway queue manager alias definition by defining **remote queue** GATEWAY1. Type the definition command as shown in the text box, and press the Enter key. **This definition requires the queue name by itself, with the other remote queue attributes left blank.**

```
DEFINE QR(GATEWAY1)
```

Expected response:

```
AMQ8006: IBM MQ queue created.
```

- ___ 71. Ensure that the queue was created successfully.
- ___ 72. Type `end` and press the Enter key to close the `runmqsc` session for queue manager MQG1.

Section 3: Baseline the round-robin workload balancing behavior by sending messages from the outside queue manager MQE5

- ___ 73. File `many_messages.txt` is found in the `C:\LabFiles\Unit3` directory. Review the display in the text box that contains the contents of file `many_messages.txt`. The file contains 20 lines. **Each line becomes one separate message when you use sample program `amqspu`.**

one
two
three
four
five
six
seven
eight
nine
ten
eleven
twelve
thirteen
fourteen
fifteen
sixteen
seventeen
eighteen
nineteen
twenty

- __ 74. From your command prompt window, ***ensure that you are still at directory C:\LabFiles\Unit3.***
- __ 75. Use sample program `amqspout` to send the messages from queue manager E5 to the G1 gateway, to reach the cluster queues, by typing the command exactly as shown. Be careful to use the correct queue and queue manager names.
-

```
amqspout TO.GATEWAY MQE5 < many_messages.txt
```

Expected results:

```
Sample AMQSPUT0 start  
target queue is TO.GATEWAY  
Sample AMQSPUT0 end
```

Section 4: Check the number of messages that arrived at the WM253.IN queue at queue manager MQ0A

- __ 76. Type `runmqsc MQ0A` and press the Enter key to start a `runmqsc` session for queue manager MQ0A.

- ___ 77. Check how many messages arrived at queue manager MQ0A by typing the command as shown in the display box, and press the Enter key. **Ensure that you use the CURDEPTH attribute.**

```
dis q(WM*) curdepth
```

Expected results:

```
1 : dis q(WM*) curdepth
AMQ8409: Display Queue details.
      QUEUE(WM253.IN)                                TYPE(QLOCAL)
      CURDEPTH(10) <=== out of 20 messages sent, for the iteration of the test in this
example, 10 arrived at the MQ0A hosted queue.
```



CAUTION

Your results might not be the same as in this example. Normally, you should see an even distribution. However, if you followed all steps, it is possible that IBM MQ might decide on different candidate queue managers for the cluster queues. If you see a total of 20 messages for the WM253.IN queues hosted by MQ0A and MQ03, you can proceed to the next steps.

- ___ 78. Type `end` and press the Enter key to close the `runmqsc` session for queue manager MQ0A.
- ___ 79. Use sample program `amqsget` to retrieve the messages in the MQ0A WM253.IN queue by typing the command as displayed in the text box.
-

```
amqsget WM253.IN MQ0A
```

Expected reply (Note: you might see either the even or the odd number messages)

```
Sample AMQSGET0 start
message <two>
message <four>
message <six>
message <eight>
message <ten>
message <twelve>
message <fourteen>
message <sixteen>
message <eighteen>
message <twenty>
no more messages
Sample AMQSGET0 end
```

- __ 80. You can use CTL-C after the messages display, or wait a few seconds for `amqsget` to stop by itself.

Section 5: Check the number of messages that arrived at the WM253.IN queue at queue manager MQ03

- __ 81. Type `runmqsc MQ03` and press the Enter key to start a `runmqsc` session for queue manager MQ03.
- __ 82. Check how many messages arrived at queue manager MQ03 by typing the command as shown in the display box, and press the Enter key. **Ensure that you use the CURDEPTH attribute.**

```
dis q(WM*) curdepth
```

Expected results:

```
1 : dis q(WM*) curdepth
AMQ8409: Display Queue details.
      QUEUE(WM253.IN)                TYPE(QLOCAL)
      CURDEPTH(10) <=== out of 20 messages sent, 10 might have arrived
```

- __ 83. How many messages are reported in the CURDEPTH attribute?
-



CAUTION

Your results might not be the same as in this example. Normally, you should see an even distribution. However, if you followed all steps, it is possible that IBM MQ might decide on different candidate queue managers for the cluster queues. If you see a total of 20 messages for the WM253.IN queues hosted by MQ0A and MQ03, you can proceed to the next steps.

- __ 84. Type `end` and press the Enter key to close the `runmqsc` session for queue manager MQ03.
- __ 85. Use sample program `amqsget` to retrieve the messages in the MQ03 WM253.IN queue by typing the command as displayed in the text box.

```
amqsget WM253.IN MQ03
```

Expected reply (Note: you might see either the even or the odd numbered messages)

Sample AMQSGETO start

```
message <one>
message <three>
message <five>
message <seven>
message <nine>
message <eleven>
message <thirteen>
message <fifteen>
message <seventeen>
message <nineteen>
^C
```

- __ 86. Ensure that you empty both clustered queues by using the `amqsget` command.

Section 6: Repeat the test with a higher number of messages

- __ 87. Repeat the sending of messages via the gateway **three times** by typing the command `amqsput TO.GATEWAY MQE5 < many_messages.txt` and pressing the Enter key. This step might lead to a more even message distribution. This step puts a total of 60 messages to the cluster queues.
-



Hint

After you type `amqsput TO.GATEWAY MQE5 < many_messages.txt` the first time, you can use the “up” arrow key to recall the command, and then press the Enter key to repeat execution of the command.

- __ 88. Start a `runmqsc` session for queue manager MQ0A and then display the current message depth for the clustered queue by typing `dis q(WM*) curdepth`. How many messages do you see in the second test? **Leave the `runmqsc` session open.**
- __ 89. Type `clear ql(WM253.IN)` and press the Enter key to clear the messages off queue WM253.IN at queue manager MQ0A.
- __ 90. Type `end` and press the Enter key to exit the MQ0A `runmqsc` session.
- __ 91. Start a `runmqsc` session for queue manager MQ03 and then display the current message depth for the clustered queue by typing: `dis q(WM*) curdepth`
- How many messages do you see in the second test? **Leave the `runmqsc` session open.**
- __ 92. Type `clear ql(WM253.IN)` and press the Enter key to clear the messages off queue WM253.IN at queue manager MQ03.

__ 93. Type `end` and press the Enter key to exit the MQ03 `runmqsc` session.



Note

You might notice a more even distribution of messages in the second test.

2.4. Redisplay the cluster queues

Section 1: Return to queue manager MQG1 and repeat the cluster queue display



Important

In an earlier step, you attempted to display the cluster queues from queue manager MQG1 by using the command `DIS QCLUSTER`. When you ran the earlier display, queue manager MQG1 did not encounter any need to look for clustered queues. However, when you sent messages to MQG1 from queue manager MQE5, queue manager MQG1 had to find queues WM253.IN by contacting one of the full repository queue managers. Now queue manager MQG1 “knows” about the clustered queues.

-
- ___ 94. From the command prompt window, start a `runmqsc` session for queue manager MQG1 by typing `runmqsc MQG1` and pressing the Enter key. Ensure that you use MQG1 to start the session.
 - ___ 95. Type the `DIS QCLUSTER` command as shown in the text box, and press the Enter key. Ensure that you include the `clusqmgr` attribute at the end of the command so that you see the queue manager to which each queue belongs.
-

```
dis qcluster(*) clusqmgr
```

Expected results:

```
2 : dis qcluster(*) clusqmgr
AMQ8409: Display Queue details.
      QUEUE(WM253.IN)                                TYPE(QCLUSTER)
      CLUSQMGR(MQ03)
AMQ8409: Display Queue details.
      QUEUE(WM253.IN)                                TYPE(QCLUSTER)
      CLUSQMGR(MQ0A)
```

- ___ 96. Review your results. The response shows both cluster queues, and the queue manager to which they belong.
- ___ 97. Type `end` and press the Enter key to close the `runmqsc` session for queue manager MQG1.

**Reminder**

When IBM MQ administrators start to work with clusters, a frequent mistake is to try to list the cluster queues with the `DIS QCLUSTER` command right after the queues are defined. Then, when the queues are not found, they try to refresh the cluster, and still do not find the queues.

Each queue manager member of the cluster “learns” about cluster queues the first time that messages are put in these queues. In your future work, remember that a non-full repository cluster queue manager learns about cluster queues after a put is attempted.

End of exercise

Exercise 3. Working with cluster administration tasks

Estimated time

01:30

Overview

In this exercise, you follow processes to accomplish selected cluster administration tasks. As part of the tasks, you use various cluster-specific administration commands.

Objectives

After completing this exercise, you should be able to:

- Add a queue manager that uses a separate transmission queue to the cluster
- Inhibit use of a queue manager by using the SUSPEND command
- Contrast the results and possible actions that the various SUSPEND modes require
- Describe how some actions that you take to administer cluster message channels resemble actions that you take to administer distributed message channels
- Restore use of a suspended queue manager by using the RESUME command
- Remove a queue manager from the cluster by using the RESET command
- Use the REFRESH CLUSTER command and review the entries that are generated in the queue manager log for this command
- Describe IBM MQ Explorer cluster administration capabilities

Requirements

Successful completion of Exercises 1 and 2.

3.1. Add a queue manager that uses separate transmission queues to the cluster



Note

The instructions in this section are derived from the IBM Information Center procedure that was shown in the lecture that accompanies this exercise. The command to add the dead-letter local queue was added to the instructions in the step that changes the queue manager.

Section 1: Create and start queue manager MQ0X to listen on port 1670

- ___ 1. Open a command prompt window by clicking the command prompt icon in the Windows taskbar.
- ___ 2. Create queue manager MQ0X to listen on port 1681 by typing the command as shown in the text box, and press the Enter key.

```
crtmqm -p 1681 -u MQ0X.DLQ MQ0X
```

Expected response:

```
IBM MQ queue manager created.
Directory 'C:\ProgramData\IBM\MQ\qmgrs\MQ0X' created.
The queue manager is associated with installation 'IBMMQV9'.
Creating or replacing default objects for queue manager 'MQ0X'.
Default objects statistics : 87 created. 0 replaced. 0 failed.
Completing setup.
Setup completed.
```

- ___ 3. Start the queue manager by typing the command as shown in the text box, and press the Enter key.

```
strmqm MQ0X
```

Expected response:

```
IBM MQ queue manager 'MQ0X' starting.
The queue manager is associated with installation 'IBMMQV9'.
5 log records accessed on queue manager 'MQ0X' during the log replay
phase.
Log replay for queue manager 'MQ0X' complete.
Transaction manager state recovered for queue manager 'MQ0X'.
IBM MQ queue manager 'MQ0X' started using V9.0.0.0.
```

Section 2: Alter queue manager MQ0X to use a separate transmission queue before adding the queue manager to the cluster



Important

The queue manager `DEFCLXQ` attribute must be changed before the queue manager is incorporated to the cluster. Do not define the `CLUSRCVR` or `CLUSDR` channels until after the `DEFCLXQ` attribute is changed.

- ___ 4. Start a `runmqsc` session for queue manager MQ0X by typing `runmqsc MQ0X` and pressing the Enter key.
- ___ 5. Change the `DEFCLXQ` attribute of the queue manager by typing the command as shown in the text box, and press the Enter key.

```
alter qmgr DEFCLXQ(CHANNEL)
```

Expected response:

```
1: alter qmgr DEFCLXQ(CHANNEL)
AMQ8005: IBM MQ queue manager changed.
```

- ___ 6. **Do not exit** the `runmqsc` session.
- ___ 7. Add the dead-letter queue that is specified when you created the queue manager by typing the command as shown on the text box, and press the Enter key. **Leave the `runmqsc` session open.**

```
def ql(MQ0X.DLQ)
```

Expected response:

```
2 : def ql(MQ0X.DLQ)
AMQ8006: IBM MQ queue created.
```



Note

In an earlier exercise, you added the queue that was identified as the dead-letter queue and altered the maximum message length of the cluster support queues and the dead-letter queue. However, queue manager MQ0X is deleted later in this exercise without any other use; therefore, the extra steps are not warranted and are deliberately skipped for brevity.

Section 3: Define the MQ0X CLUSRCVR channel for cluster WMADMCLS

- ___ 8. Create the CLUSRCVR channel by typing the command as shown in the text box, and press the Enter key. Optionally, you can copy and paste the command that is located under the respective section name in the `Lab3_copyAndPaste.txt` file that is found in directory `C:\LabFiles\Unit4`. If you opt to type the command, ensure that you type it as one contiguous line, regardless of whether it wraps over to a second line.

```
define chl(WMADMCLS.MQ0X) chltype(CLUSRCVR) trptype(TCP)
conname('localhost(1681)') cluster(WMADMCLS)
```

Expected response:

```
2 : define chl(WMADMCLS.MQ0X) chltype(CLUSRCVR) trptype(TCP)
conname('localhost(1681)') cluster(WMADMCLS)
AMQ8014: IBM MQ channel created..
```

- ___ 9. **Do not exit** the `runmqsc` session.

Section 4: Define the MQ0X CLUSSDR channel for cluster WMADMCLS

- ___ 10. Create the CLUSSDR channel to use the MQ0A full repository by typing the command as shown in the text box, and press the Enter key. Optionally, you can copy and paste the command that is located under the respective section name in the `Lab3_copyAndPaste.txt` file that is found in directory `C:\LabFiles\Unit4`. If you opt to type the command, ensure that you type it as one contiguous line, regardless of whether it wraps over to a second line.

```
def chl (WMADMCLS.MQ0A) chltype(CLUSSDR) trptype(TCP)
conname('localhost(1657)') cluster(WMADMCLS)
```

Expected response:

```
3: def chl (WMADMCLS.MQ0A) chltype(CLUSSDR) trptype(TCP)
conname('localhost(1657)') cluster(WMADMCLS)
AMQ8014: IBM MQ channel created.
```

- ___ 11. Type `end` and press the Enter key to end the `runmqsc` session.

Section 5: Verify and review results of the addition of queue manager MQ0X to cluster WMADMCLS

- ___ 12. Check the outcome of the previous configuration from the perspective of the MQ0X queue manager by directing the output of a DIS CLUSQMGR session to a Notepad file. Type the command in your command prompt window as shown on the text box. You can optionally copy the command from the copy-and-paste file that is used in earlier steps in this exercise.

```
echo dis clusqmgr(*) all | runmqsc MQ0X > disX.txt
```

- ___ 13. Type `notepad disX.txt` and press the Enter key to open the file with the results of the `DIS CLUSQMGR` command.
- ___ 14. Your results should resemble the display that is shown in the text box. However, the channels in your display might appear in different order. Read the observations noted directly after the display.
-

5724-H72 (C) Copyright IBM Corp. 1994, 2016.

Starting MQSC for queue manager MQ0X.

1 : dis clusqmgr(*) all

AMQ8441: Display Cluster Queue Manager details.

CLUSQMGR(MQ0A)	ALTDATE(2017-04-28)
ALTTIME(09.35.50)	BATCHHB(0)
BATCHINT(0)	BATCHLIM(5000)
BATCHSZ(50)	CHANNEL(WMADMCLS.MQ0A)
CLUSDATE(2017-05-04)	CLUSTER(WMADMCLS)
CLUSTIME(06.50.08)	CLWLPRTY(0)
CLWLRANK(0)	CLWLWGHT(50)
COMPHDR(NONE)	COMPMSG(NONE)
CONNNAME(localhost(1657))	CONVERT(NO)
DEFTYPE(CLUSSDRB)	DESCR()
DISCINT(6000)	HBINT(300)
KAINT(AUTO)	LOCLADDR()
LONGRTY(999999999)	LONGTMR(1200)
MAXMSGL(32768)	MCANAME()
MCATYPE(THREAD)	MCAUSER()
MODENAME()	MRDATA()
MREXIT()	MRRTY(10)
MRTMR(1000)	MSGDATA()
MSGEXIT()	NETPRTY(0)
NPMSPEED(FAST)	PASSWORD()
PROPCTL(COMPAT)	PUTAUT(DEF)
QMID(MQ0A_2017-04-27_00.52.51)	QMTYPE(REPOS)
RCVDATA()	RCVEXIT()
SCYDATA()	SCYEXIT()
SENDDATA()	SENDEXIT()
SEQWRAP(999999999)	SHORTRTY(10)
SHORTTMR(60)	SSLCAUTH(REQUIRED)
SSLCIPH()	SSLPEER()
STATUS(RUNNING)	SUSPEND(NO)
TPNAME()	TRPTYPE(TCP)
USEDLQ(YES)	USERID()
VERSION(09000000)	
XMITQ(SYSTEM.CLUSTER.TRANSMIT.WMADMCLS.MQ0A)	

AMQ8441: Display Cluster Queue Manager details.

CLUSQMGR(MQ0B)	ALTDATE(2017-04-28)
ALTTIME(09.43.17)	BATCHHB(0)
BATCHINT(0)	BATCHLIM(5000)
BATCHSZ(50)	CHANNEL(WMADMCLS.MQ0B)
CLUSDATE(2017-05-04)	CLUSTER(WMADMCLS)
CLUSTIME(06.50.08)	CLWLPRTY(0)
CLWLRANK(0)	CLWLWGHT(50)
COMPHDR(NONE)	COMPMSG(NONE)
CONNAME(localhost(1658))	CONVERT(NO)
DEFTYPE(CLUSSDRA)	DESCR()
DISCINT(6000)	HBINT(300)
KAINT(AUTO)	LOCLADDR()
LONGRTY(999999999)	LONGTMR(1200)
MAXMSGL(32768)	MCANAME()
MCATYPE(THREAD)	MCAUSER()
MODENAME()	MRDATA()
MREXIT()	MRRTY(10)
MRTMR(1000)	MSGDATA()
MSGEXIT()	NETPRTY(0)
NPMSPEED(FAST)	PASSWORD()
PROPCTL(COMPAT)	PUTAUT(DEF)
QMID(MQ0B_2017-04-27_00.52.52)	QMTYPE(REPOS)
RCVDATA()	RCVEXIT()
SCYDATA()	SCYEXIT()
SENDDATA()	SENDEXIT()
SEQWRAP(999999999)	SHORTRTY(10)
SHORTTMR(60)	SSLCAUTH(REQUIRED)
SSLCIPH()	SSLPEER()
STATUS(RUNNING)	SUSPEND(NO)
TPNAME()	TRPTYPE(TCP)
USEDLQ(YES)	USERID()
VERSION(09000000)	
XMITQ(SYSTEM.CLUSTER.TRANSMIT.WMADMCLS.MQ0B)	

AMQ8441: Display Cluster Queue Manager details.

CLUSQMGR(MQ0X)	ALTDATE(2017-05-04)
ALTTIME(06.33.51)	BATCHHB(0)
BATCHINT(0)	BATCHLIM(5000)
BATCHSZ(50)	CHANNEL(WMADMCLS.MQ0X)
CLUSDATE(2017-05-04)	CLUSTER(WMADMCLS)
CLUSTIME(06.33.51)	CLWLPRTY(0)
CLWLRANK(0)	CLWLWGHT(50)
COMPHDR(NONE)	COMPMSG(NONE)
CONNAME(localhost(1681))	CONVERT(NO)
DEFTYPE(CLUSRCVR)	DESCR()
DISCINT(6000)	HBINT(300)
KAINT(AUTO)	LOCLADDR()

LONGRTY(999999999)	LONGTMR(1200)
MAXMSGL(4194304)	MCANAME()
MCATYPE(THREAD)	MCAUSER()
MODENAME()	MRDATA()
MREXIT()	MRRTY(10)
MRTMR(1000)	MSGDATA()
MSGEXIT()	NETPRTY(0)
NPMSPEED(FAST)	PASSWORD()
PROPCTL(COMPAT)	PUTAUT(DEF)
QMID(MQ0X_2017-05-03_05.53.55)	QMTYPE(NORMAL)
RCVDATA()	RCVEXIT()
SCYDATA()	SCYEXIT()
SENDDATA()	SENDEXIT()
SEQWRAP(999999999)	SHORTRTY(10)
SHORTTMR(60)	SSLCAUTH(REQUIRED)
SSLCIPH()	SSLPEER()
STATUS(RUNNING)	SUSPEND(NO)
TPNAME()	TRPTYPE(TCP)
USEDLQ(YES)	USERID()
VERSION(09000000)	XMITQ()

One MQSC command read.

No commands have a syntax error.

All valid MQSC commands were processed.

-
- ___ a. The display shows three channels. The first channel in the display is queue manager MQ0A.
 - ___ b. MQ0A is the full repository queue manager to which queue manager MQ0X has an explicitly defined CLUSSDR channel. Since channel WMADMCLS.MQ0X is both dynamically and explicitly defined, the channel is of type CLUSSDRB.
 - ___ c. Many of the attributes for channel WMADMCLS.MQ0X look familiar. However, notice the transmission queue that is used for this first channel, `SYSTEM.CLUSTER.TRANSMIT.WMADMCLS.MQ0A`. The CLUSSDRB channels that you reviewed earlier used `z`. However, you changed the MQ0X queue manager `DEFCLXQ` attribute to `CHANNEL`. As a result, a new, separate transmission is created and used when a dynamic channel, either CLUSSDRA or CLUSSDRB, is created from the MQ0X queue manager.
 - ___ d. You now look at the second channel in the display, WMADMCLS.MQ0B, which is for queue manager MQ0B. Since MQ0X does **not** have an explicit CLUSSDR definition to MQ0B, the DEFTYPE for this channel is CLUSSDRA.
 - ___ e. Look at the transmission, or XMITQ parameter for channel WMADMCLS.MQ0B. It is also a separate dynamic transmission queue.
 - ___ f. The last channel is the DEFTYPE CLUSRCVR channel for queue manager MQ0X.

Section 6: Review the results of the addition of queue manager MQ0X to the cluster from the perspective of queue manager MQ0A

- ___ 15. Check the outcome of the previous configuration from the perspective of the MQ0A queue manager by directing the output of a DIS CLUSQMGR session to a Notepad file. Type the command in your command prompt window as shown on the text box. You can optionally copy the command from the copy-and-paste file that was used in earlier steps in this exercise.

```
echo dis clusqmgr(*) all | runmqsc MQ0A > disA.txt
```

- ___ 16. Type `notepad disA.txt` and press the Enter key to open the file with the results of the DIS CLUSQMGR command.
- ___ 17. Your results should resemble the display that is shown in the text box. However, the channels in your display might appear in different order. Read the observations that are noted directly after the display. In this display, you concentrate on the entry for queue manager MQ0X, and the rest of the queue manager displays are partial.

5724-H72 (C) Copyright IBM Corp. 1994, 2016.
Starting MQSC for queue manager MQ0A.

1 : dis clusqmgr(*) all

AMQ8441: Display Cluster Queue Manager details.

CLUSQMGR(MQ03) ALTDATE(2017-04-28)

... ..

AMQ8441: Display Cluster Queue Manager details.

CLUSQMGR(MQ0A) ALTDATE(2017-04-28)

... ..

DEFTYPE(CLUSRCVR) DESCR()

... ..

AMQ8441: Display Cluster Queue Manager details.

CLUSQMGR(MQ0B) ALTDATE(2017-04-28)

... ..

AMQ8441: Display Cluster Queue Manager details.

CLUSQMGR(MQ0X)	ALTDATE(2017-05-04)
ALTTIME(06.33.51)	BATCHHB(0)
BATCHINT(0)	BATCHLIM(5000)
BATCHSZ(50)	CHANNEL(WMADMCLS.MQ0X)
CLUSDATE(2017-05-04)	CLUSTER(WMADMCLS)
CLUSTIME(06.50.08)	CLWLPRTY(0)
CLWLRANK(0)	CLWLWGHT(50)
COMPHDR(NONE)	COMPMSG(NONE)
CONNNAME(localhost(1681))	CONVERT(NO)
DEFTYPE(CLUSSDRA)	DESCR()
DISCINT(6000)	HBINT(300)
KAINT(AUTO)	LOCLADDR()
LONGRTY(999999999)	LONGTMR(1200)
MAXMSGL(4194304)	MCANAME()
MCATYPE(THREAD)	MCAUSER()
MODENAME()	MRDATA()
MREXIT()	MRRTY(10)
MRTMR(1000)	MSGDATA()
MSGEXIT()	NETPRTY(0)
NPMSPEED(FAST)	PASSWORD()
PROPCTL(COMPAT)	PUTAUT(DEF)
QMID(MQ0X_2017-05-03_05.53.55)	QMTYPE(NORMAL)
RCVDATA()	RCVEXIT()
SCYDATA()	SCYEXIT()
SENDDATA()	SENDEXIT()
SEQWRAP(999999999)	SHORTRTY(10)
SHORTTMR(60)	SSLCAUTH(REQUIRED)
SSLCIPH()	SSLPEER()
STATUS(RUNNING)	SUSPEND(NO)

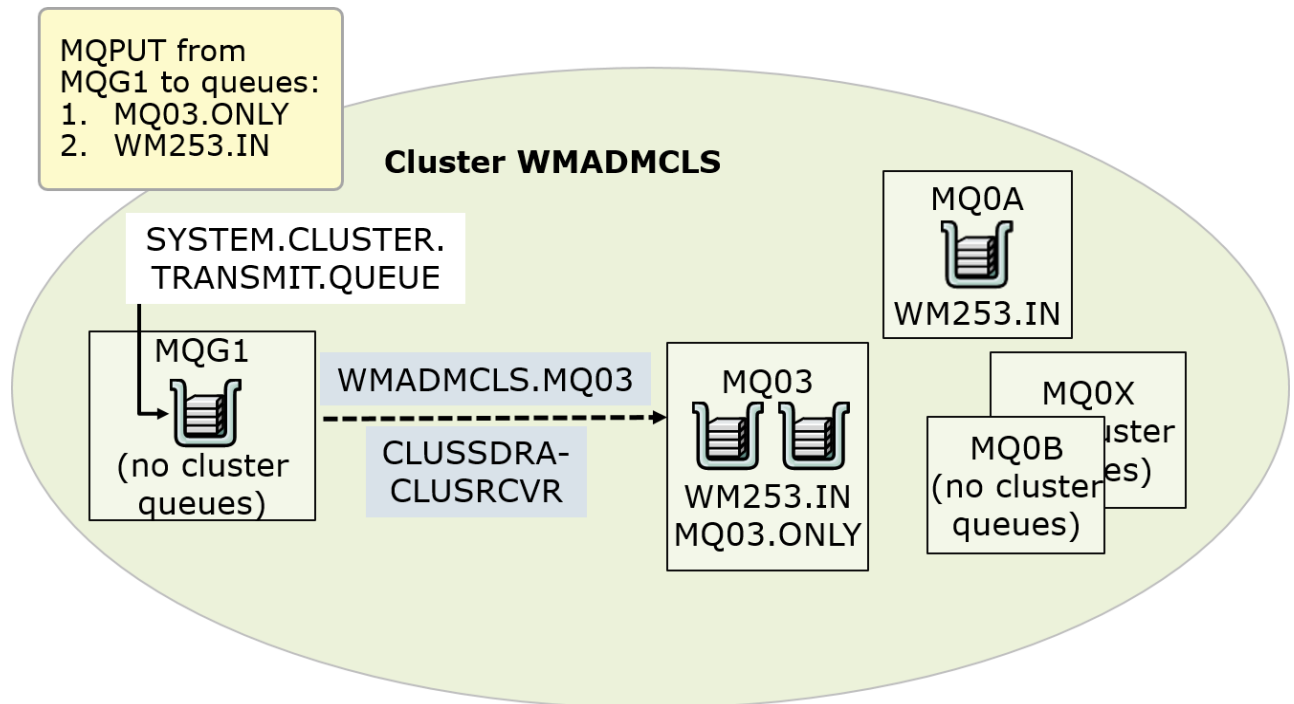
```

TPNAME( )                      TRPTYPE(TCP)
USEDLQ(YES)                    USERID( )
VERSION(09000000)
XMITQ(SYSTEM.CLUSTER.TRANSMIT.QUEUE)
AMQ8441: Display Cluster Queue Manager details.
      CLUSQMGR(MQG1)            ALTDAT(2017-04-28)
... ..
One MQSC command read.
No commands have a syntax error.
All valid MQSC commands were processed.

```

- ___ a. For purposes of this review, you look at the channel from MQ0A to MQ0X. You can disregard the other channels, which are partially shown.
- ___ b. Channel WMADMCLS.MQ0X for queue manager MQ0X is a DEFTYPE CLUSSDRA channel. The CLUSSDR channel for full repository queue manager MQ0A points to the second full repository queue manager MQ0B. Therefore, the dynamic channel to MQ0X is a type CLUSSDRA.
- ___ c. Look at the transmit queue. Channel WMADMCLS.MQ0X from MQ0A to MQ0X uses the SYSTEM.CLUSTER.TRANSMIT queue. The change that was made to queue manager MQ0X before it joined the cluster, `alter qmgr DEFCLXQ(CHANNEL)`, impacts dynamic channels outbound from MQ0X exclusively. Channels inbound to MQ0X continue to use the cluster transmission queue channels that were originally used by each queue manager.

3.2. Inhibit and restore use of a cluster queue manager by using SUSPEND and RESUME



Note

In the figure, the highlighted cluster channels are between queue manager MQG1 and MQ03. Other cluster channels pairs are omitted because they are not needed for purposes of this exercise.



Information

In this part of the exercise, you:

- Work with suspend modes QUIESCE and FORCE, and observe the differences in behavior between the two modes
- Find your messages for a suspended queue manager, and compare in which ways cluster channels are similar to distributed message channels
- Test the MQPUT from queue manager MQG1
- Use MQGET and other utilities from different cluster queue managers to determine where the messages are

Cluster queue MQ03.ONLY, which you create next, exists in queue manager MQ03 exclusively.

**Stop**

Ensure that you read this section before you proceed with this part of the exercise.

The main objective of this part of the exercise is to learn how to use the `suspend` and `resume` commands. It is important to be aware that after you suspend and then resume a queue manager, you must ensure that the cluster channels are not in stopped status. In some cases, you might need to manually start a channel.

Due to possible changes in queue manager activity for each student, the results of your steps might not be exactly like the results that are shown in the instructions. When it comes to channel states in this part of the exercise, do not be concerned if the results are not exactly like the expected results that are shown in the text boxes. You can proceed.

However, after you “resume” a queue manager, type `DIS CLUSQMGR(*) deftype status` in each cluster member queue manager, and ensure that the channels are not in stopped status.

If you have any problems that appear to be due to cluster connectivity, use the `DIS CLUSQMGR` method that is explained earlier, and if needed, start or resolve any channel. If a channel in `RETRYING` status is not resolved after you restart it, you need to investigate. Look in the queue manager log for the queue manager that the channel is attempting to connect to so that you can determine any other problems.

Section 1: Create queue cluster queue MQ03.ONLY at queue manager MQ03

- ___ 18. Start a `runmqsc` session for queue manager MQ03 by typing `runmqsc MQ03` and pressing the Enter key.
- ___ 19. Create the cluster local queue by typing `def ql(MQ03.ONLY) cluster(WMADMCLS)` and pressing the Enter key.
- ___ 20. Do not close the `runmqsc` session.

Section 2: Advise WMADMCLS queue managers to stop sending messages to queue manager MQ03 by using the SUSPEND command

- ___ 21. Limit queue manager MQ03 from receiving messages in cluster WMADMCLS by typing the `SUSPEND` command with `MODE(QUIESCE)` as shown in the text box.

```
suspend qmgr cluster(WMADMCLS)
```

Expected response:

```
1 : suspend qmgr cluster(WMADMCLS)
AMQ8557: SUSPEND QUEUE MANAGER accepted.
```

**Reminder**

QUIESCE is the default MODE value in the SUSPEND command. You do not need to specify the MODE value for QUIESCE mode.

__ 22. Keep the `runmqsc` session open.

Section 3: Check the status of the cluster channels after the SUSPEND from the perspective of the MQ03 queue manager

__ 23. Type the `DIS CLUSQMGR` command in the MQ03 `runmqsc` session with the options that are shown in the text box, and press the Enter key.

```
dis clusqmgr(*) deftype status
```

Expected response:

```
3 : dis clusqmgr(*) deftype status
AMQ8441: Display Cluster Queue Manager details.
CLUSQMGR(MQ03)                                CHANNEL(WMADMCLS.MQ03)
CLUSTER(WMADMCLS)                             DEFTYPE(CLUSRCVR)
STATUS(RUNNING)
AMQ8441: Display Cluster Queue Manager details.
CLUSQMGR(MQ0A)                                CHANNEL(WMADMCLS.MQ0A)
CLUSTER(WMADMCLS)                             DEFTYPE(CLUSSDRA)
STATUS(RUNNING)
AMQ8441: Display Cluster Queue Manager details.
CLUSQMGR(MQ0B)                                CHANNEL(WMADMCLS.MQ0B)
CLUSTER(WMADMCLS)                             DEFTYPE(CLUSSDRB)
STATUS(RUNNING)
```

**Information**

Notice how the CLUSRCVR and other CLUSSDRA and CLUSSDRB channels continue to run after the SUSPEND with MODE(QUIESCE).

__ 24. Type `end` and press the Enter key to exit the `runmqsc` session.

Section 4: Use queue manager MQG1 to send messages to cluster queue MQ03.ONLY, hosted in queue manager MQ03 exclusively

__ 25. From your command prompt window, change to the `Unit4` directory by typing `cd C:\LabFiles\Unit4` and pressing the Enter key.

- ___ 26. Ensure that file `many_messages.txt` is present in the directory by typing `dir` and pressing the Enter key.
- ___ 27. Send a few messages to cluster queue `MQ03.ONLY` from queue manager `MQG1` by using the `amqspout` sample program as shown in the text box, and press the Enter key.

```
amqspout MQ03.ONLY MQG1 < many_messages.txt
```

Expected results:

```
Sample AMQSPUT0 start
target queue is MQ03.ONLY
Sample AMQSPUT0 end
```

Section 5: Determine whether the messages sent to queue `MQ03.ONLY` arrived at suspended queue manager `MQ03`

- ___ 28. Use the `amqsget` sample program to get messages from queue `MQ03.ONLY` at queue manager `MQ03` by typing the command as shown in the text box, and press the Enter key.

```
amqsget MQ03.ONLY MQ03
```

Expected results (partial, you see all 20 messages arrived):

```
Sample AMQSGET0 start
message <one
message <two>
message <three>
... ..
```

- ___ 29. Either type CTL-C to exit `amqsget`, or wait for `amqsget` to stop.
-



Information

When the suspended queue manager was the only queue manager to host the cluster queue, all 20 messages were sent to the cluster queue, regardless of the `SUSPEND MODE(QUIESCE)`.

Section 6: Use queue manager `MQG1` to send messages to cluster queue `WM253.IN`, which is hosted in queue managers `MQ03` and `MQ0A`

- ___ 30. Send a few messages to cluster queue `MQ03.ONLY` from queue manager `MQG1` by using the `amqspout` sample program as shown in the text box, and press the Enter key.

```
amqspout WM253.IN MQ01 < many_messages.txt
```

Expected results:

```
Sample AMQSPUT0 start
target queue is WM253.IN
Sample AMQSPUT0 end
```

Section 7: Determine whether the messages sent to queue WM253.IN arrived at suspended queue manager MQ03

- ___ 31. Use the `amqsget` sample program to get messages from queue WM253.IN. Since WM253.IN is hosted in both queue managers MQ03 and MQ0A, try the `amqsget` command for queue manager MQ0A first. Type the command as shown in the text box, and press the Enter key.
-

```
amqsget WM253.IN MQ0A
```

Expected results (partial, you see all 20 messages arrived):

```
Sample AMQSGET0 start
message <one>
message <two>
message <three>
... ..
```

- ___ 32. All 20 messages are expected to arrive at queue manager MQ0A. It is not necessary to repeat the `amqsget` at queue manager MQ03.
- ___ 33. You can type CTL-C to exit `amqsget`, or wait for `amqsget` to stop.
-



Information

When a cluster queue is hosted in more than one queue manager, including the suspended queue manager, messages get routed to other instances of the same queue in an available, or not suspended, queue manager. In this case, WM253.IN was also available at queue manager MQ0A.

Section 8: Restore use of queue manager MQ03 to cluster WMADMCLS

- ___ 34. Start a `runmqsc` session for queue manager MQ03 by typing `runmqsc MQ03` and pressing the Enter key.
- ___ 35. Restore queue manager MQ03 to full use in cluster WMADMCLS by typing the `RESUME` command as shown in the text box.

```
resume qmgr cluster(WMADMCLS)
```

Expected response:

```
1 : resume qmgr cluster(WMADMCLS)
```

```
AMQ8556: RESUME QUEUE MANAGER accepted.
```

- __ 36. Type `end` and press the Enter key to exit the `runmqsc` session.

Section 9: Review the entries in the MQ03 queue manager log

- __ 37. Use Windows Explorer to navigate to the queue manager log for MQ03, which is found at directory `C:\ProgramData\IBM\MQ\qmgrs\MQ03\errors`.
- __ 38. Double-click **AMQERR01.LOG** to open the file.
- __ 39. Scroll to the end of the file, and then by working backwards in the file, review the entries regarding the `SUSPEND` and `RESUME` commands. What process is involved in the two commands?

```
-----
5/5/2017 03:59:59 - Process(1740.1) User(MUSR_MQADMIN)
Program(amqrrmfa.exe)
```

```
Host(WS2008R2X64) Installation(IBM MQV9)
```

```
VRMF(9.0.0.0) QMgr(MQ03)
```

```
AMQ9443: Suspend Qmgr Cluster command processed.
```

```
EXPLANATION:
```

```
The Suspend Qmgr Cluster(WMADMCLS) command completed. 4 objects
suspended. In
```

```
the case of suspending a list of clusters, the cluster name shown is the
first
```

```
name in the list.
```

```
ACTION:
```

```
None.
```

```
-----
5/5/2017 05:06:06 - Process(1740.1) User(MUSR_MQADMIN)
Program(amqrrmfa.exe)
```

```
Host(WS2008R2X64) Installation(IBM MQV9)
```

```
VRMF(9.0.0.0) QMgr(MQ03)
```

```
AMQ9444: Resume Qmgr Cluster command processed.
```

```
EXPLANATION:
```

```
The Resume Qmgr Cluster(WMADMCLS) command completed. 4 objects resumed.
In the
```

```
case of resuming a list of clusters, the cluster name shown is the first
name
```

```
in the list.
```

```
ACTION:
```

```
None.
```



Information

Answer: You see that process `amqrmfa.exe`, the cluster repository manager, is involved in the SUSPEND and RESUME process.

__ 40. Close the `AMQERR01.LOG` file.

Section 10: Repeat the SUSPEND command for queue manager MQ03 with MODE(FORCE)

- __ 41. Start a `runmqsc` session for queue manager MQ03 by typing `runmqsc MQ03` and pressing the Enter key.
- __ 42. Force-inhibit queue manager MQ03 from receiving messages in cluster WMADMCLS by typing the `SUSPEND` command with `MODE(FORCE)` as shown in the text box. Keep the `runmqsc` session open.

```
suspend qmgr cluster(WMADMCLS) MODE(FORCE)
```

Expected response:

```
1 : suspend qmgr cluster(WMADMCLS) MODE(FORCE)
AMQ8557: SUSPEND QUEUE MANAGER accepted.
```

- __ 43. From the `runmqsc` session for queue manager MQ03, repeat the `DIS CLUSQMGR` display as shown in the text box, and press the Enter key.

```
dis clusqmgr(*) deftype status
```

Expected results:

```
2 : dis clusqmgr(*) deftype status
AMQ8441: Display Cluster Queue Manager details.
CLUSQMGR(MQ03)                                CHANNEL(WMADMCLS.MQ03)
CLUSTER(WMADMCLS)                              DEFTYPE(CLUSRCVR)
STATUS(STOPPED)
AMQ8441: Display Cluster Queue Manager details.
CLUSQMGR(MQ0A)                                CHANNEL(WMADMCLS.MQ0A)
CLUSTER(WMADMCLS)                              DEFTYPE(CLUSSDRA)
STATUS(RUNNING)
AMQ8441: Display Cluster Queue Manager details.
CLUSQMGR(MQ0B)                                CHANNEL(WMADMCLS.MQ0B)
CLUSTER(WMADMCLS)                              DEFTYPE(CLUSSDRB)
STATUS(RUNNING)
```

- __ 44. Type `end` and press the Enter key to exit the `runmqsc` session.

**Information**

Notice how the CLUSRCVR channel for MQ03, WMADMCLS.MQ03, is now in STOPPED status. Other cluster queue managers cannot establish dynamic CLUSSDRA or CLUSSDRB to MQ03.

Section 11: Use queue manager MQG1 to send messages to cluster queue MQ03.ONLY, hosted in queue manager MQ03 exclusively

- ___ 45. From your command prompt window, change to the Unit 4 directory by typing `cd C:\LabFiles\Unit4` and pressing the Enter key.
- ___ 46. Ensure that file `many_messages.txt` is present in the directory by typing `dir` and pressing the Enter key.
- ___ 47. Pause for a minute or so before you proceed to the next step to ensure that the suspend process was completed.
- ___ 48. Send a few messages to cluster queue MQ03.ONLY from queue manager MQG1 by using the `amqspout` sample program as shown in the text box, and press the Enter key.

```
amqspout MQ03.ONLY MQG1 < many_messages.txt
```

Expected results:

```
Sample AMQSPUT0 start
target queue is MQ03.ONLY
Sample AMQSPUT0 end
```

Section 12: Determine whether the messages sent to queue MQ03.ONLY arrived at suspended queue manager MQ03

- ___ 49. Use the `amqsget` sample program to get messages from queue MQ03.ONLY at queue manager MQ03 by typing the command as shown in the text box, and press the Enter key.
- ___ 50. **Wait for `amqsget` to end. Do not use CTL-C.**

```
amqsget MQ03.ONLY MQ03
```

Expected results:

```
Sample AMQSGET0 start
no more messages
Sample AMQSGET0 end
```



Information

When SUSPEND with MODE(FORCE) is specified, the channels to the queue manager are stopped, and no messages arrive at a queue even when the suspended queue manager is the only host for the cluster queue.

You apply some basic IBM MQ skills to determine where the messages intended for queue MQ03.ONLY at queue manager MQ03 might be. For example, from basic administration, you know that when a channel is not running, messages are held in the transmission queue of the sending queue manager.

- What queue manager was used to put the messages to the MQ03.ONLY queue, as specified at the end of the `amqspout` command?
- What transmission queue does queue manager MQG1 use when it sends messages to other queue managers in the cluster?

You now find where the messages are.

Section 13: Check the cluster transmission queue for queue manager MQG1

- ___ 51. Start a `runmqsc` session for queue manager MQG1 by typing `runmqsc MQG1` and pressing the Enter key.
- ___ 52. Check the number of messages in the `SYSTEM.CLUSTER.TRANSMIT.QUEUE` for queue manager MQG1 by typing the command as shown in the text box, and press the Enter key. As specified, the command displays the current message depth in the queue.

```
dis q(SYSTEM.CLUSTER.TRANSMIT.QUEUE) curdepth
```

Expected results:

```
1 : dis q(SYSTEM.CLUSTER.TRANSMIT.QUEUE) curdepth
AMQ8409: Display Queue details.
      QUEUE(SYSTEM.CLUSTER.TRANSMIT.QUEUE)      TYPE(QLOCAL)
      CURDEPTH(20)
```

- ___ 53. Exit the `runmqsc` session by typing `end` and pressing the Enter key.

Section 14: Check the entries in the MQG1 queue manager log

- ___ 54. Use Windows Explorer to navigate to the queue manager log for MQG1 found at directory `C:\ProgramData\IBM\MQ\qmgrs\MQG1\errors`.
- ___ 55. Double-click `AMQERR01.LOG` to open the file.

- ___ 56. Scroll to the end of the file, and review the last set of three entries that pertain to channel WMADMCLS.MQ03.

```
----- amqrccca.c : 1090 -----
5/5/2017 05:42:52 - Process(5216.60) User(MUSR_MQADMIN) Program(amqrmppa.exe)
                        Host(WS2008R2X64) Installation(IBM MQV9)
                        VRMF(9.0.0.0) QMgr(MQG1)
AMQ9002: Channel 'WMADMCLS.MQ03' is starting. <== attempted to start for amqsput
EXPLANATION:
Channel 'WMADMCLS.MQ03' is starting.
ACTION:
None.
-----

5/5/2017 05:42:52 - Process(5216.60) User(MUSR_MQADMIN) Program(amqrmppa.exe)
                        Host(WS2008R2X64) Installation(IBM MQV9)
                        VRMF(9.0.0.0) QMgr(MQG1)
AMQ9558: The remote channel 'WMADMCLS.MQ03' on host 'ws2008r2x64
(127.0.0.1)(1653)' is not currently available.
EXPLANATION:
The channel program ended because an instance of channel 'WMADMCLS.MQ03' could
not be started on the remote system. This could be for one of the following
reasons:
The channel is disabled.
The remote system does not have sufficient resources to run another instance of
the channel.
In the case of a client-connection channel, the limit on the number of
instances configured for the remote server-connection channel was reached.
ACTION:
Check the remote system to ensure that the channel is able to run. Try the
operation again.
----- cmqxrft.c : 573 -----
5/5/2017 05:42:52 - Process(5216.60) User(MUSR_MQADMIN) Program(amqrmppa.exe)
                        Host(WS2008R2X64) Installation(IBM MQV9)
                        VRMF(9.0.0.0) QMgr(MQG1)
AMQ9999: Channel 'WMADMCLS.MQ03' to host 'localhost(1653)' ended abnormally.
EXPLANATION:
The channel program running under process ID 5216(5588) for channel
'WMADMCLS.MQ03' ended abnormally. The host name is 'localhost(1653)'; in some
cases the host name cannot be determined and so is shown as '????'.
ACTION:
Look at previous error messages for the channel program in the error logs to
determine the cause of the failure. Note that this message can be excluded
completely or suppressed by tuning the "ExcludeMessage" or "SuppressMessage"
attributes under the "QMErrorLog" stanza in qm.ini. Further information can be
found in the System Administration Guide.
----- amqrccca.c : 1090 -----
```



Information

The messages in the queue manager logs of other active cluster queue manager members might be inconclusive when a cluster queue manager is suspended with FORCE. There are no problems with the channel or insufficient resources. The messages clear up after you resume the queue manager.

Section 15: Restore use of queue manager MQ03 to cluster WMADMCLS

- ___ 57. Start a `runmqsc` session for queue manager MQ03 by typing `runmqsc MQ03` and pressing the Enter key.
- ___ 58. Restore queue manager MQ03 to full use in cluster WMADMCLS by typing the `RESUME` command as shown in the text box. **Keep the `runmqsc` session open.**

```
resume qmgr cluster(WMADMCLS)
```

Expected response:

```
1 : resume qmgr cluster(WMADMCLS)
AMQ8556: RESUME QUEUE MANAGER accepted.
```

Section 16: Review the status of the MQ03 cluster channels by using the DIS CLUSQMGR command

- ___ 59. From the `runmqsc` session for queue manager MQ03, repeat the DIS CLUSQMGR display as shown in the text box, and press the Enter key.

```
dis clusqmgr(*) deftype status
```

Expected results:

```
2 : dis clusqmgr(*) deftype status
AMQ8441: Display Cluster Queue Manager details.
      CLUSQMGR(MQ03)                                CHANNEL(WMADMCLS.MQ03)
      CLUSTER(WMADMCLS)                              DEFTYPE(CLUSRCVR)
      STATUS(INACTIVE) <== Check any channel not in running status
AMQ8441: Display Cluster Queue Manager details.
      CLUSQMGR(MQ0A)                                CHANNEL(WMADMCLS.MQ0A)
      CLUSTER(WMADMCLS)                              DEFTYPE(CLUSSDRA)
      STATUS(RUNNING)
AMQ8441: Display Cluster Queue Manager details.
      CLUSQMGR(MQ0B)                                CHANNEL(WMADMCLS.MQ0B)
      CLUSTER(WMADMCLS)                              DEFTYPE(CLUSSDRB)
      STATUS(RUNNING)
```

- ___ 60. What do you see as the status of the MQ03 CLUSRCVR channel named WMADMCLS.MQ03?



Information

Your results are expected to resemble the display. The status of the CLUSRCVR channel for MQ03 is INACTIVE, which denotes that other CLUSSDRA or CLUSSDRB channels from other cluster member queue managers to WMADMCLS.M03 did not automatically start after the RESUME.

Treat the CLUSSDRx to CLUSRCVR channels as you treat a SENDER-RECEIVER distributed message channel, by looking at the sending, or CLUSSDRx side. In this case, you know that you were using the MQG1 queue manager. Check queue manager MQG1.

__ 61. Type `end` and press the Enter key to exit the `runmqsc` session.

Section 17: Review the status of the dynamic WMADMCLS.MQ03 CLUSSDRA channel from queue manager MQG1

- __ 62. Start a `runmqsc` session for queue manager MQG1 by typing `runmqsc MQG1` and pressing the Enter key.
- __ 63. From the `runmqsc` session for queue manager MQG1, repeat the `DIS CLUSQMGR` display as shown in the text box, and press the Enter key. **Keep the MQG1 `runmqsc` session open.**

```
dis clusqmgr(*) deftype status
```

Expected results:

```
2 : dis clusqmgr(*) deftype status
AMQ8441: Display Cluster Queue Manager details.
      CLUSQMGR(MQ03)                      CHANNEL(WMADMCLS.MQ03)
      CLUSTER(WMADMCLS)                   DEFTYPE(CLUSSDRA)
      STATUS(RETRYING)
AMQ8441: Display Cluster Queue Manager details.
      CLUSQMGR(MQ0A)                      CHANNEL(WMADMCLS.MQ0A)
      CLUSTER(WMADMCLS)                   DEFTYPE(CLUSSDRB)
      STATUS(INACTIVE)
AMQ8441: Display Cluster Queue Manager details.
      CLUSQMGR(MQ0B)                      CHANNEL(WMADMCLS.MQ0B)
      CLUSTER(WMADMCLS)                   DEFTYPE(CLUSSDRA)
      STATUS(INACTIVE)
AMQ8441: Display Cluster Queue Manager details.
      CLUSQMGR(MQG1)                      CHANNEL(WMADMCLS.MQG1)
      CLUSTER(WMADMCLS)                   DEFTYPE(CLUSRCVR)
      STATUS(RUNNING)
```


**Information**

CLUSSDRA channel WMADMCLS.MQ03 did not get restarted with the `RESUME` command and requires manual intervention.

A status of INACTIVE for the other CLUSSDRA and CLUSSDRB type dynamic channels is acceptable.

You now correct the WMADMCLS.MQ03 CLUSSDRA channel that is in RETRYING status by restarting the WMADMCLS.MQ003 channel from queue manager MQG1.

Section 18: Restart CLUSSDRA channel WMADMCLS.MQ03 from MQG1

- ___ 64. From queue manager MQG1, restart the WMADMCLS.MQ03 channel by typing the command as shown in the text box, and press the Enter key. **Keep the MQG1 runmqsc session open.**

**CAUTION**

Your results might vary depending on previous actions. The objective is for any CLUSSDRx channel to be running. If your results are not exactly as shown, check the queue manager error log and use normal channel restore actions. For example, if a channel is INDOUBT, you might need to use the `RESOLVE CHANNEL` command. Refer to the IBM Information Center for details on the use of the different commands.

```
start chl(WMADMCLS.MQ03)
```

Expected results:

```
3 : start chl(WMADMCLS.MQ03)
AMQ8018: Start IBM MQ channel accepted.
```

Section 19: Repeat the DIS CLUSGMGR command with the deftype and status attributes and review the results

- ___ 65. Type the DIS CLUSQMGR display as shown in the text box, and press the Enter key. **Keep the MQG1 runmqsc session open.**

```
dis clusqmgr(*) deftype status
```

Expected results:

```
4 : dis clusqmgr(*) deftype status
AMQ8441: Display Cluster Queue Manager details.
      CLUSQMGR(MQ03)                      CHANNEL(WMADMCLS.MQ03)
      CLUSTER(WMADMCLS)                   DEFTYPE(CLUSSDRA)
      STATUS(RUNNING)
AMQ8441: Display Cluster Queue Manager details.
      CLUSQMGR(MQ0A)                      CHANNEL(WMADMCLS.MQ0A)
      CLUSTER(WMADMCLS)                   DEFTYPE(CLUSSDRB)
      STATUS(INACTIVE)
AMQ8441: Display Cluster Queue Manager details.
      CLUSQMGR(MQ0B)                      CHANNEL(WMADMCLS.MQ0B)
      CLUSTER(WMADMCLS)                   DEFTYPE(CLUSSDRA)
      STATUS(INACTIVE)
AMQ8441: Display Cluster Queue Manager details.
      CLUSQMGR(MQG1)                      CHANNEL(WMADMCLS.MQG1)
      CLUSTER(WMADMCLS)                   DEFTYPE(CLUSRCVR)
      STATUS(RUNNING)
```

___ 66. Review the results.

- ___ a. Look for the status of CLUSSDRA channel WMADMCLS.MQ03. It is expected to be in RUNNING status. **Keep the MQG1 runmqsc session open.**
- ___ b. If WMADMCLS.MQ03 is not running, check the MQG1 queue manager log, and possibly the MQ03 queue manager log for messages about channel WMADMCLS.MQ03. Resolve the channel problem before you continue.
- ___ c. For the other channels in the display, a status of INACTIVE is not of concern. You are working with a small configuration, and the channels normally stopped due to inactivity.
- ___ d. For the other channels in the display, you need to follow up on any channels with a status of RETRYING or STOPPED.

Section 20: Ensure that the transmission queue for queue manager MQG1 is clear of messages



Note

In a busy queue manager, you might see some messages on the queue manager transmission queue. However, if the count of messages in the transmission queue keeps incrementing, then a problem exists.

Your lab environment has minimal activity. If channel WMADMCLS.MQ03 started successfully, the message count, or CURDEPTH field display for the SYSTEM.CLUSTER.TRANSMIT.QUEUE, is expected to be 0. If your display shows other than 0, resolve the problem before you continue.

-
- ___ 67. Type the `DIS Q` display as shown in the text box, and press the Enter key. Ensure that you include the `CURDEPTH` parameter in the `DIS Q` command.
-

```
dis q(SYSTEM.CLUSTER.TRANSMIT.QUEUE) curdepth
```

Expected results:

```
5 : dis q(SYSTEM.CLUSTER.TRANSMIT.QUEUE) curdepth
AMQ8409: Display Queue details.
      QUEUE(SYSTEM.CLUSTER.TRANSMIT.QUEUE)      TYPE(QLOCAL)
      CURDEPTH(0)
```

- ___ 68. Repeat the `amqsget` command to queue MQ03 from queue manager MQ03 now. All 20 messages are expected to be available.
-

```
amqsget MQ03.ONLY MQ03
```

Expected results (partial messages are shown):

```
Sample AMQSGETO start
message <one>
message <two>
message <three>
... ..
```

3.3. Remove MQ0X from the cluster by using the alternative process with RESET

Section 1: Confirm and stop the cluster channels at queue manager MQ0X

- ___ 69. Start a `runmqsc` session for queue manager MQ0X by typing `runmqsc MQ0X` and pressing the Enter key.
- ___ 70. Confirm what channels need to be stopped. From the `runmqsc` session for queue manager MQ0X, display the cluster channels by using the `DIS CLUSQMGR` display as shown in the text box, and press the Enter key. **Keep the MQ0X `runmqsc` session open.**

```
dis clusqmgr(*) deftype status
```

Expected results:

```
1 : dis clusqmgr(*) deftype status
AMQ8441: Display Cluster Queue Manager details.
CLUSQMGR(MQ0A)                CHANNEL(WMADMCLS.MQ0A)
CLUSTER(WMADMCLS)             DEFTYPE(CLUSSDRB)
STATUS(INACTIVE)
AMQ8441: Display Cluster Queue Manager details.
CLUSQMGR(MQ0B)                CHANNEL(WMADMCLS.MQ0B)
CLUSTER(WMADMCLS)             DEFTYPE(CLUSSDRA)
STATUS(INACTIVE)
AMQ8441: Display Cluster Queue Manager details.
CLUSQMGR(MQ0X)                CHANNEL(WMADMCLS.MQ0X)
CLUSTER(WMADMCLS)             DEFTYPE(CLUSRCVR)
STATUS(INACTIVE)
```



Note

The channel status for the channels that are displayed is expected to be INACTIVE, since you did not use this queue manager and a few minutes elapsed after your last actions.

- ___ 71. Stop the cluster channels that are listed in the display by using the command as shown in the text box, and press the Enter key. Be sure to stop the CLUSRCVR type channel with force to immediately stop incoming messages. **Keep the MQ0X `runmqsc` session open.**

The three stop channel commands are followed by the corresponding expected responses. For this display, the channels were not stopped in the order that they were displayed. The CLUSRCVR type channel was stopped first, with the force option. You can stop the channels in any order, but you must use the force option for the CLUSRCVR channel.

```
stop chl(WMADMCLS.MQ0X) mode(force)
  2 : stop chl(WMADMCLS.MQ0X) mode(force)
AMQ8019: Stop IBM MQ channel accepted.
stop chl(WMADMCLS.MQ0A)
  3 : stop chl(WMADMCLS.MQ0A)
AMQ8019: Stop IBM MQ channel accepted.
stop chl(WMADMCLS.MQ0B)
  4 : stop chl(WMADMCLS.MQ0B)
AMQ8019: Stop IBM MQ channel accepted.
```



CAUTION

It is possible that the response to each `stop chl` command might be `AMQ9533: Channel 'WMADMCLS.MQ0X' is not currently active`. If you received the `AMQ9533` message, you can proceed with the next step.

Section 2: Confirm whether all channels stopped



Note

In a busy environment, before you display the channels, you might elect to wait a few moments for all the channels to stop. The lab environment for this course has no activity, so the channels stop right away.

- ___ 72. From the `runmqsc` session for MQ0X, repeat the `DIS CLUSQMGR` by typing the command as shown in the text box, and press the Enter key. **Keep the MQ0X `runmqsc` session open.**

```
dis clusqmgr(*) deftype status
```

Expected response:

```
5 : dis clusqmgr(*) deftype status
AMQ8441: Display Cluster Queue Manager details.
      CLUSQMGR(MQ0A)                      CHANNEL(WMADMCLS.MQ0A)
      CLUSTER(WMADMCLS)                   DEFTYPE(CLUSSDRB)
      STATUS(STOPPED)
AMQ8441: Display Cluster Queue Manager details.
      CLUSQMGR(MQ0B)                      CHANNEL(WMADMCLS.MQ0B)
      CLUSTER(WMADMCLS)                   DEFTYPE(CLUSSDRA)
      STATUS(STOPPED)
AMQ8441: Display Cluster Queue Manager details.
      CLUSQMGR(MQ0X)                      CHANNEL(WMADMCLS.MQ0X)
      CLUSTER(WMADMCLS)                   DEFTYPE(CLUSRCVR)
      STATUS(STOPPED)
```

Section 3: Remove the manually defined CLUSSDR channel to queue manager MQ0A from queue manager MQ0X

- ___ 73. From the `runmqsc` session for MQ0X, delete channel WMADMCLS.MQ0A by typing the command as shown in the text box, and press the Enter key.

```
delete chl(WMADMCLS.MQ0A)
```

Expected response:

```
6 : delete chl(WMADMCLS.MQ0A)
AMQ8015: IBM MQ channel deleted.
```

- ___ 74. Type `end` and press the Enter key to exit the MQ0X `runmqsc` session.

Section 4: Remove knowledge of queue manager MQ0X from the cluster by using the RESET command from a full repository queue manager



Note

Under most circumstances, you can use the `RESET` command with the queue manager name. However, the queue manager ID, or QMID, is a unique value that mitigates the possibility of purging the wrong queue manager from the cluster. In this section, you use `RESET` by QMID.

**CAUTION**

Enclose the QMID in single quotation marks to prevent syntax errors when you issue the RESET command.

-
- ___ 75. Start a `runmqsc` session for full repository queue manager MQ0A by typing `runmqsc MQ0A` and pressing the Enter key.
- ___ 76. From the `runmqsc` session for MQ0A, obtain the QMID for queue manager MQ0X by typing the `DIS CLUSQMGR` as shown in the text box, and press the Enter key.
-

```
dis clusqmgr(MQ0X) QMID
```

Expected response:

```
1: dis clusqmgr(MQ0X) QMID
```

```
AMQ8441: Display Cluster Queue Manager details.
```

```
CLUSQMGR(MQ0X)
```

```
CHANNEL(WMADMCLS.MQ0X)
```

```
CLUSTER(WMADMCLS)
```

```
QMID(MQ0X_2017-05-03_05.53.55)
```

- ___ 77. You need to include the resulting QMID value, which is enclosed in single quotation marks, in the `RESET CLUSTER` command.
- ___ a. You can either type the value or copy and paste.
 - ___ b. If you type the value, be careful to differentiate between the underscore characters, the hyphen that is used in the date, and the periods that are used in the time.
 - ___ c. If you opt to copy and paste, right-click the Administrator Command Prompt window heading, and select **Edit > Mark**. Highlight the QMID value, right-click, and select **Edit > Copy**. If you are not familiar with this method, type the QMID value instead.
- ___ 78. From the `runmqsc` session for MQ0A, purge knowledge of queue manager MQ0X from the cluster by typing the `RESET CLUSTER` command all in one line, as shown in the text box. Press the Enter key.
-

**Important**

Whether you type or paste the QMID value, **ensure that you enclosed the QMID value that you copied from your display in single quotation marks**, or the `RESET CLUSTER` results in a syntax error. Ensure that you include the `QUEUES` parameter. **Do not use the QMID in the example command.**

Most errors that are experienced in this section are due to an incorrectly typed QMID. The copy-and-paste method is preferred.

**Stop**

Carefully check the QMID you typed or copied. If you have any typographical errors and provide the wrong QMID, the response is still AMQ8559: RESET CLUSTER accepted. The message states that the command is accepted, but not whether the command succeeded or failed, or whether the QMID is valid.

If you do not type the correct QMID that you obtained from the `DIS CLUSQMGR(MQ0X) QMID` display, MQ0X is not removed, and the remaining steps in this section do not work as expected.

```
reset cluster(WMADMCLS) QMID('MQ0X_2017-05-03_05.53.55')
action(forceremove) queues(no)
```

Expected response:

```
2 : reset cluster(WMADMCLS) QMID('MQ0X_2017-05-03_05.53.55')
action(forceremove) queues(no)
AMQ8559: RESET CLUSTER accepted.
```

-
- __ 79. Check that your results contain the AMQ8559: RESET CLUSTER accepted message.
 - __ 80. If the RESET CLUSTER command did not complete successfully:
 - __ a. Ensure that you are in a session for full repository queue manager MQ0A when you try to run the RESET CLUSTER command.
 - __ b. Correct any syntax errors and repeat until your response is as expected. **Keep the MQ0A running session open.**
 - __ 81. Confirm that queue manager MQ0X no longer shows in the MQ0A repository by typing the command as shown in the text box, and press the Enter key.
-

```
dis clusqmgr(MQ0X)
```

Expected result:

```
3 : dis clusqmgr(MQ0X)
AMQ8147: IBM MQ object MQ0X not found. <== "not found" is expected
```

-
- __ 82. Review the results.

**Stop**

If queue manager MQ0X displayed after you typed `dis clusqmgr(MQ0X)`, it means that an error occurred with the `reset cluster` command even if you saw “RESET CLUSTER accepted” in the results. Repeat the `reset cluster` command by taking care to specify the queue manager ID (QMID) field exactly as you saw in the display, and within apostrophes.

__ 83. Type `end` and press the Enter key to exit the MQ0A `runmqsc` session.

Section 5: Repeat the DIS CLUSQMGR for MQ0X in each member of the cluster, MQ0B, MQG1, and MQ03

- __ 84. Wait a few moments, such as 30 seconds to 1 minute, to allow the notification that queue manager MQ0X is no longer in the cluster to reach cluster member queue managers.
- __ 85. Start a `runmqsc` session for full repository queue manager MQ0B by typing `runmqsc MQ0B` and pressing the Enter key.
- __ 86. Check that queue manager MQ0X does not show by typing `dis clusqmgr(MQ0X)` and pressing the Enter key. Results should indicate “IBM MQ object MQ0X not found.”
- __ 87. Exit the `runmqsc` session for MQ0B by typing `end` and pressing the Enter key.
- __ 88. Start a `runmqsc` session for queue manager MQG1 by typing `runmqsc MQG1` and pressing the Enter key.
- __ 89. Check that queue manager MQ0X does not show by typing `dis clusqmgr(MQ0X)` and pressing the Enter key. Results should indicate “IBM MQ object MQ0X not found.”
- __ 90. Exit the `runmqsc` session for MQG1 by typing `end` and pressing the Enter key.
- __ 91. Start a `runmqsc` session for queue manager MQ03 by typing `runmqsc MQ03` and pressing the Enter key.
- __ 92. Check that queue manager MQ0X does not show by typing `dis clusqmgr(MQ0X)` and pressing the Enter key. Results should indicate “IBM MQ object MQ0X not found.”
- __ 93. Exit the `runmqsc` session for MQ03 by typing `end` and pressing the Enter key.

**Note**

In an actual situation, you might complete the DIS CLUSQMGR displays a few moments after the `RESET CLUSTER` command to allow all activity to complete. The amount of time to wait depends on the amount of activity in the cluster.

Section 6: After you confirmed that activity for MQ0X ceased, remove the CLUSRCVR channel from queue manager MQ0X

- __ 94. Start a `runmqsc` session for full repository queue manager MQ0B by typing `runmqsc MQ0B` and pressing the Enter key.

- ___ 95. Remove the WMADMCLS.MQ0X CLUSRCVR channel by typing the command as shown in the text box, and press the Enter key.
-

```
delete chl(WMADMCLS.MQ0X)
```

Expected results:

```
1 : delete chl(WMADMCLS.MQ0  
AMQ8015: IBM MQ channel deleted.
```

- ___ 96. Exit the `runmqsc` session for MQ03 by typing `end` and pressing the Enter key.
- ___ 97. Close the command prompt window by typing `exit` and pressing the Enter key.

3.4. Use the REFRESH CLUSTER command

Section 1: Refresh the cluster from full repository MQ0B



CAUTION

It is acceptable to use REFRESH CLUSTER from a full repository when the REPOS(NO) option, which is the default, is used. It is invalid to use REFRESH CLUSTER with REPOS(YES) from a full repository.

-
- ___ 98. Start a `runmqsc` session for full repository queue manager MQ0B by typing `runmqsc MQ0B` and pressing the Enter key.
 - ___ 99. Refresh the cluster by typing the command as shown in the text box, and press the Enter key.
-

```
refresh cluster(*)
```

Expected response:

```
1 : refresh cluster(*)
AMQ8558: REFRESH CLUSTER accepted.
```

- ___ 100.Exit `runmqsc` by typing `end` and pressing the Enter key.

Section 2: Review the entries that are made in the MQ0B queue manager log as a result of the REFRESH command

- ___ 101.Use Windows Explorer to navigate to the MQ0B queue manager log at `C:\ProgramData\IBM\MQ\qmgrs\MQ0B\errors`.
- ___ 102.Double-click file `AMQERR01.LOG` to open it.
- ___ 103.Scroll to the end of the file, and then scroll back to locate the point where the REFRESH command started. Your results should resemble the display.

- ___ 104. Review the messages that the REFRESH CLUSTER command generates, and note all the warnings on what a significant impact the use of this command can take in a busy cluster.

```
-----
5/9/2017 07:47:31 - Process(3944.1) User(MUSR_MQADMIN) Program(amqrrmfa.exe)
                    Host(WS2008R2X64) Installation(IBM MQV9)
                    VRMF(9.0.0.0) QMgr(MQ0B)
```

AMQ9875: REFRESH CLUSTER processing started for cluster.

EXPLANATION:

Refresh cluster processing started for cluster(*) A REFRESH CLUSTER command has been issued on this queue manager. In phase one this will discard all locally cached information for the cluster and request new information from other members of the cluster when necessary. Phase two processes the information received. **For large cluster configurations this process can take a significant time**, especially on full repository queue managers, and during this time applications attempting to access cluster resources may see failures to resolve cluster resources. In addition, cluster configuration changes made on this queue manager may not be processed until the refresh process has completed.

ACTION:

Defer any cluster related work on this queue manager until both phases are complete. Message AMQ9442 or message AMQ9404 will be issued to this log at the end of phase one. Completion of phase two can be determined when SYSTEM.CLUSTER.COMMAND.QUEUE has reached a consistently empty state.

```
-----
5/9/2017 07:47:31 - Process(3944.1) User(MUSR_MQADMIN) Program(amqrrmfa.exe)
                    Host(WS2008R2X64) Installation(IBM MQV9)
                    VRMF(9.0.0.0) QMgr(MQ0B)
```

AMQ9442: Phase one of REFRESH CLUSTER has completed.

EXPLANATION:

Phase one of REFRESH CLUSTER has completed. The Refresh Cluster(*) command caused 10 objects to be refreshed and republished to 3 queue managers. Applications attempting to access cluster resources may see failures to resolve cluster resources until phase two of REFRESH CLUSTER is complete. Phase two is complete once all new information has been received from other members of the cluster. Monitor your SYSTEM.CLUSTER.COMMAND.QUEUE to determine when it has reached a consistently empty state to indicate that the refresh process has completed.

ACTION:

None.

```
-----
5/9/2017 07:47:31 - Process(5724.82) User(MUSR_MQADMIN) Program(amqrmppa.exe)
                    Host(WS2008R2X64) Installation(IBM MQV9)
                    VRMF(9.0.0.0) QMgr(MQ0B)
```

AMQ9002: Channel 'WMADMCLS.MQ0B' is starting.

EXPLANATION:

Channel 'WMADMCLS.MQ0B' is starting.

ACTION:

None.



Note

The larger the cluster, the more significant the impact of the `REFRESH CLUSTER` command. You are directed to check the number of messages in the `SYSTEM.CLUSTER.COMMAND.QUEUE` to determine whether phase 2 of the refresh process ended. However, in the class environment, the cluster is small and without much activity, so `SYSTEM.CLUSTER.COMMAND.QUEUE` empties too fast to monitor.



CAUTION


The `REFRESH CLUSTER` command cannot fix configuration errors. Do not use this command unless you fully understand its use and impact. Refrain from using the `REFRESH CLUSTER` command to rectify problems. Use regular IBM MQ channel capabilities and the `DIS CLUSQMGR` cluster command to investigate and correct cluster problems.

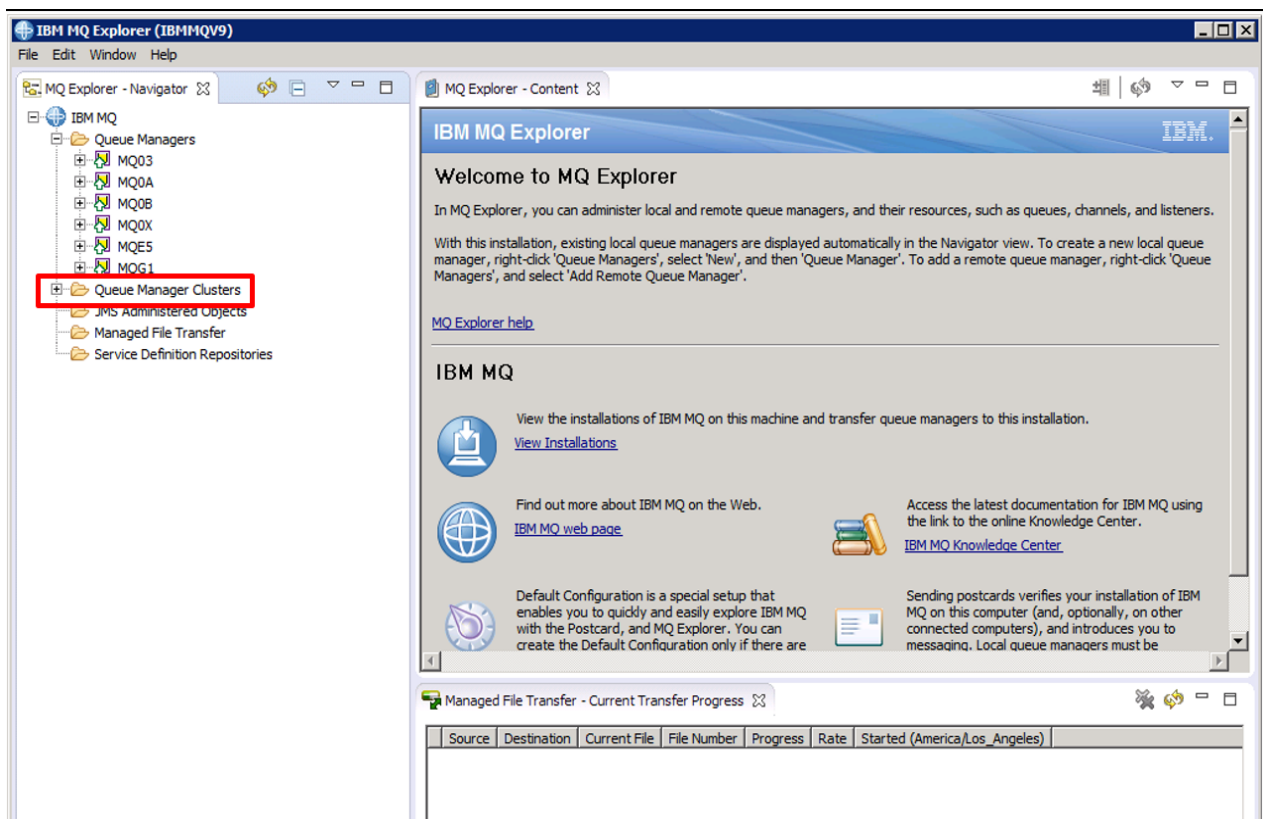
3.5. Take an initial look at the IBM MQ Explorer cluster administration capabilities



Information

You use IBM MQ Explorer to do some of the tasks in later units. In this exercise, you take a first look at the cluster administrative capabilities in IBM MQ Explorer.

- ___ 105. Start an IBM MQ Explorer screen by clicking the  icon in the Windows taskbar.
- ___ 106. Wait a few moments until IBM MQ Explorer completes startup.



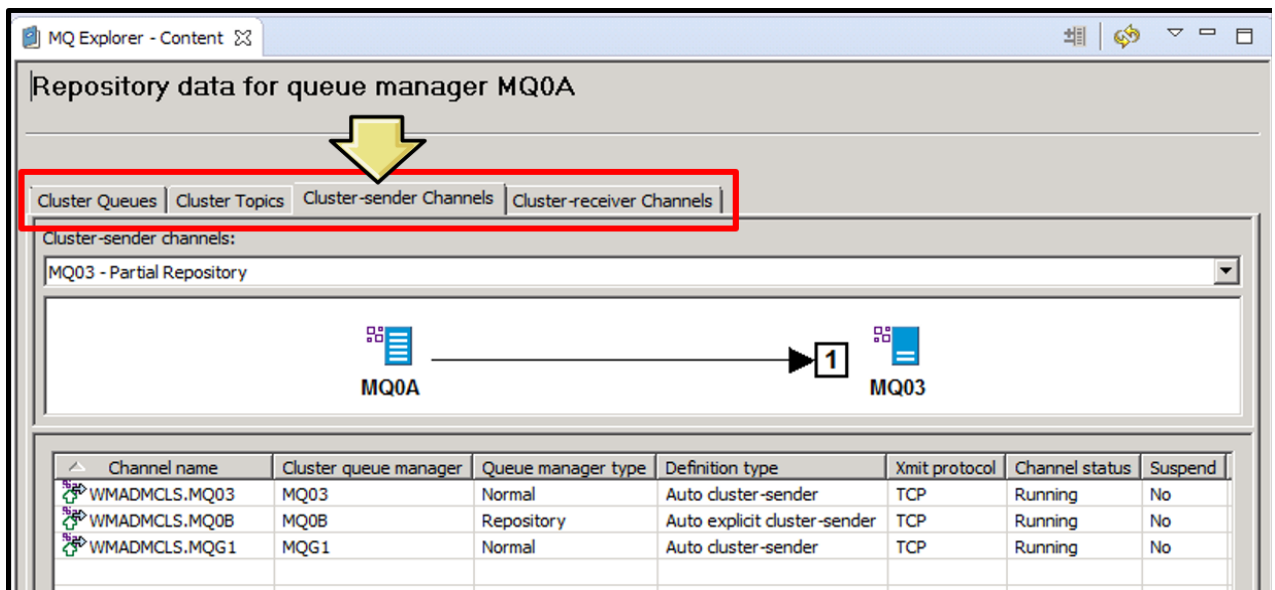
- ___ 107. After IBM MQ Explorer completes startup:
 - ___ a. Look for the Queue Manager Clusters menu, which is found in the upper-left side MQ Explorer - Navigator section.
 - ___ b. Expand the WMADMCLS menu by clicking the +. You see a list of full repository queue managers.

- ___ 108. In the next display, examine the **MQ Explorer - Content** pane on the upper right of IBM MQ Explorer.
- ___ a. This view is the view from full repository queue manager MQ0A. You see a number of tabs.
- ___ b. Select the **Cluster-sender** tab. Your view should resemble the screen display.



Note

You might find some of the channels in Inactive status, which is acceptable. In this display, the REFRESH command that is typed in the last section caused the channels to return to RUNNING status.



CAUTION

You can browse the different tabs. However, do not stop any channels or make any changes to the existing environment. You need to use the environment for subsequent exercises.

In later exercises, you use IBM MQ Explorer for various tasks. The work in this task is a first look.



Troubleshooting

Sometimes when you leave IBM MQ Explorer running in the VMware image, and return to use it, the program might be found to be unresponsive, with all displays showing as empty or garbled boxes.

- **To prevent this IBM MQ Explorer behavior in the VMware image**, close IBM MQ Explorer before you close or log off from the VMware image.

- To correct the problem if IBM MQ Explorer is unresponsive:
 - Start the Windows task manager by right-clicking the Windows taskbar and selecting **Start Task Manager**.
 - From the Task Manager Applications tab, select the IBM MQ Explorer application.
 - Right-click over the IBM MQ Explorer application, and select **End Task**. If it does not end right away, repeat the End Task.

Keep this process handy for later exercises that make extensive use of IBM MQ Explorer.

End of exercise

Exercise summary

In this exercise, you

- Added a queue manager to the cluster that uses a separate transmission by changing the `DEFCLXQ` attribute of the queue manager before adding the queue manager to the cluster.
- Inhibited and restored use of a cluster queue manager to the cluster by using the `SUSPEND` and `RESUME` commands.
- Observed how a `SUSPEND` with `MODE(FORCE)` inhibited messages from the queue manager by stopping the channels. By contrast, a `SUSPEND` with default `MODE(QUIESCE)` allowed messages to reach the queue manager when the queue manager hosted the only instance of a cluster queue.
- Observed how some of the actions that you take to administer cluster message channels resemble the actions that you take to administer distributed message channels.
- Removed a queue manager from the cluster by following the process that uses the `RESET` command.
- Used the `REFRESH CLUSTER` command and reviewed the queue manager log entries that the command generated. These entries documented and warned about the consequences of using the `REFRESH CLUSTER` command.
- Took an initial look at the IBM MQ Explorer cluster administration capabilities.

Exercise 4. Working with IBM MQ security

Estimated time

02:00

Overview

In this exercise, you harden connection and channel authentication. You learn how to review the queue manager log to determine the correct security challenge to resolve. You then learn how to mitigate channel authentication challenges by setting channel authentication rules. You also learn how to adequately invoke commands in a queue manager with required connection authentication.

Objectives

After completing this exercise, you should be able to:

- Review the commands that are used to harden connection authentication and channel authentication
- Harden connection authentication and channel authentication in a queue manager
- Create a type ADDRESSMAP channel authentication record to allow a specific queue manager to connect to a queue manager with strict channel authentication
- Differentiate between connection authentication and channel authentication entries in the queue manager log
- Create a type QMGRMAP channel authentication record to allow the cluster member queue managers to interact with a cluster queue manager with strict channel authentication
- Invoke the runmqsc utility with credentials to access a queue manager with required connection authentication
- Use the dspmqaut and setmqaut control commands to display and set object authorities
- Use the MQSC DISPLAY AUTHREC and SET AUTHREC commands to display and set object authorities
- Use control commands in a queue manager with required connection authentication

Introduction

In this exercise, you change the IBM MQ security settings for cluster queue manager MQG1 to have strict connection authentication, and implement the “back-stop” channel authentication rule. After security is tightened, you learn to incrementally identify, and resolve security challenges. The causes of these challenges might be connection authentication, channel authentication, or object

authorizations in IBM MQ. However, regardless of the cause, these challenges normally surface as the same error code: 2035.

Rather than assuming that the last change you implemented did not work, in this exercise you learn the necessity of ***always*** looking in the IBM MQ queue manager logs, even for repeated 2035 codes.

In this exercise, ***you are expected to continue to find errors after you complete some of the commands***. The errors, how you investigate the cause, and how you correct a particular scenario, are intended as a learning experience.

After security is tightened in queue manager MQG1, you find cluster channel security errors. You also find security errors in the channel between MQE5 and MQG1. Your objective is to find the cause of the error, and implement the correct configuration adjustment.

When you complete your work with channel authentication and channel authorization, in the last section you learn to work with object authorizations.

Requirements

- Cluster WMADMCLS with queue managers MQ0A, MQ0B, MQG1, and MQ03
- External queue manager MQE5
- Two-way sender-receiver channels that are established between MQE5 and MQG1
- Remote queue TO.GATEWAY at queue manager MQE5 with target queue WM253.IN
- Cluster queues WM253.IN hosted by queue managers MQ0A and MQ03
- Unprivileged user fileusr1 for the object authorizations section of this exercise

Low IBM MQ privilege user	fileusr1	web1sphere	Must be in groups: <ul style="list-style-type: none"> • Users • Remote desktop users <i>Must NOT be</i> in any IBM MQ or Windows administrative group
---------------------------	----------	------------	--

4.1. Baseline your environment

Section 1: Confirm that the sending of messages from non-cluster queue manager MQE5 to queue WM253.IN via MQG1 is operational

- ___ 1. Open a command prompt window.
- ___ 2. Change to the `Unit5` lab directory by typing `cd C:\LabFiles\Unit5` and pressing the Enter key.
- ___ 3. Ensure that no messages are in cluster queues `WM253.IN` hosted by queue managers `MQ0A` and `MQ03` by using the `amqsget` utility as shown in the text box. You can use `CTL-C` to break out of the `amqsget` loop. It does not matter whether messages are present or not provided you empty the queue with the `amqsget` before you proceed.

```
C:\LabFiles\Unit5>amqsget WM253.IN MQ0A
Sample AMQSGETO start
^C
C:\LabFiles\Unit5>amqsget WM253.IN MQ03
Sample AMQSGETO start
^C
```

- ___ 4. Use sample program `amqspout` to send the messages from queue manager `E5` to the `G1` gateway to reach the cluster queues by typing the command exactly as shown. Be careful to use the correct queue and queue manager names.

```
amqspout TO.GATEWAY MQE5 < many_messages.txt
```

Expected results:

```
Sample AMQSPUTO start
target queue is TO.GATEWAY
Sample AMQSPUTO end
```

- ___ 5. Confirm that the messages reached the `WM253.IN` queues at queue managers `MQ0A` and `MQ03` by repeating the retrieval of messages with the `amqsget` sample program as depicted in the text box.



Note

The distribution of your messages between the two hosted queues might be different from the distribution shown in the text box.

```
C:\LabFiles\Unit5>amqsget WM253.IN MQ0A
Sample AMQSGETO start
```

```
message <one>
message <three>
message <five>
```

```
... ..
```

```
C:\LabFiles\Unit5>amqsget WM253.IN MQ03
```

```
Sample AMQSGETO start
```

```
message <two>
message <four>
message <six>
```

```
... ..
```

- ___ 6. If your messages arrived at the WM253.IN queues, proceed to the next section.
- ___ 7. If your messages did not arrive to the WM253.IN queues, check the MQE5 channel status for channel MQE5.MQG1. If necessary, review the MQG1 queue manager logs. Do not continue until you can successfully send messages as depicted.

Section 2: Check the cluster channels from the perspective of queue manager MQG1

- ___ 8. From the command prompt window, type `runmqsc MQG1` and press the Enter key to start a `runmqsc` session for queue manager MQG1.
- ___ 9. Display the status of the channels by typing the command as shown in the text box, and press the Enter key. Your results might not be identical to the results in the box.

```
dis clusqmgr(*) deftype status
```

Expected results:

```
1 : dis clusqmgr(*) deftype status
AMQ8441: Display Cluster Queue Manager details.
      CLUSQMGR(MQ03)                      CHANNEL(WMADMCLS.MQ03)
      CLUSTER(WMADMCLS)                   DEFTYPE(CLUSSDRA)
      STATUS(RUNNING)
AMQ8441: Display Cluster Queue Manager details.
      CLUSQMGR(MQ0A)                      CHANNEL(WMADMCLS.MQ0A)
      CLUSTER(WMADMCLS)                   DEFTYPE(CLUSSDRB)
      STATUS(RUNNING)
AMQ8441: Display Cluster Queue Manager details.
      CLUSQMGR(MQ0B)                      CHANNEL(WMADMCLS.MQ0B)
      CLUSTER(WMADMCLS)                   DEFTYPE(CLUSSDRA)
      STATUS(INACTIVE)
AMQ8441: Display Cluster Queue Manager details.
      CLUSQMGR(MQG1)                      CHANNEL(WMADMCLS.MQG1)
      CLUSTER(WMADMCLS)                   DEFTYPE(CLUSRCVR)
      STATUS(INACTIVE)
```



Note

It is expected that you might find some channels in INACTIVE status, which is okay. The arrival of the messages from MQE5 caused channels WMADMCLS.MQ03 and WMADMCLS.MQ0A to start. If you step away, and all channels are inactive when you run the display, **provided you do not see any channels with RETRYING status**, you can proceed with the rest of the exercise.

If you see any channel in retry, check the corresponding queue manager error log and rectify the problem channel before you proceed.

__ 10. Review the results.



Important

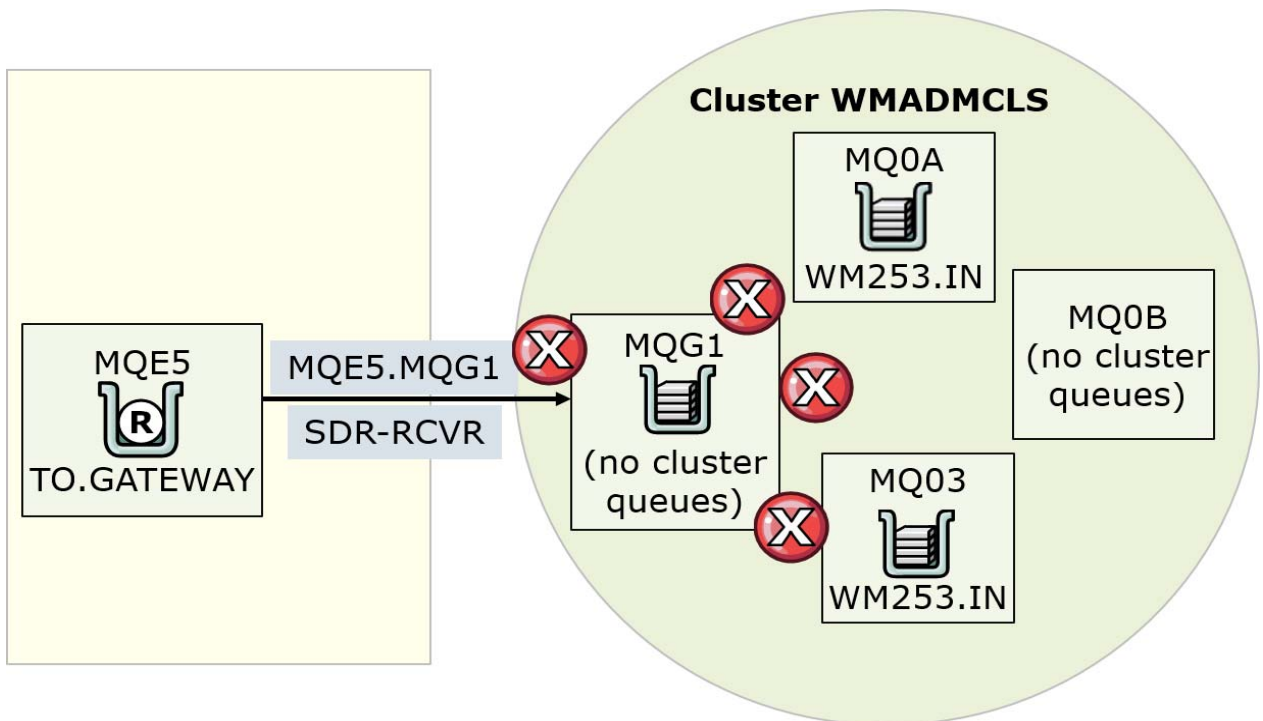
__ 11. **Ensure that you close the MQG1 runmqsc session** by typing `end` and pressing the Enter key.

4.2. Apply strict connection authentication and channel authentication to MQG1



Information

After you complete this section, you receive 2035 errors for connections to queue manager MQG1.



Section 1: Review the commands to tighten security that is found in the *lockSecurity.mqsc* script

- ___ 12. From your command prompt window, type `cd C:\LabFiles\Unit5` and press the Enter key to navigate to the lab files for this exercise.
- ___ 13. Display the `C:\LabFiles\Unit5` directory by typing `dir` and pressing the Enter key.
- ___ 14. Ensure that you find file `lockSecurity.mqsc`.
- ___ 15. The content of the `lockSecurity.mqsc` file is copied in the text box. Review the comments that follow the text box.

```

*****
* WM253 Exercise 4 lockSecurity script to harden security
* Queue manager name: MQG1
* The commands on this file harden connection authentication and
* channel authentication by taking the following actions:
* - Create the back-stop rule to block all connections
* - Creating a new authinfo record with required authentication
* - Changing the queue manager connauth to the new authinfo record
* - Refreshing queue manager security to activate the authinfo change
* - (channel rules changes do not require refresh)
* Queue manager platform: Windows
*****
*
* Create a new authinfo record with REQUIRED authentication
*
DEFINE AUTHINFO(WM253.G1.AUTHINFO) AUTHTYPE(IDPWOS) +
      CHCKCLNT(REQUIRED) CHCKLOCL(REQUIRED) REPLACE
*-----
*
* Set the queue manager CONNAUTH attribute to use the new authinfo
* and enable channel authentication
*
ALTER QMGR CONNAUTH(WM253.G1.AUTHINFO) CHLAUTH(ENABLED)
*-----
*
* Set the back-stop rule
SET CHLAUTH('*') TYPE(ADDRESSMAP) ADDRESS('*') USERSRC(NOACCESS) +
      DESCR('Back-stop rule') ACTION(REPLACE)
*-----
*
* Refresh the queue manager connection authentication security
* (channel authentication does not require a refresh)
*
REFRESH SECURITY TYPE(CONNAUTH)
*-----
*
* Display the changes in connection authentication and channel
* authentication
*
DIS QMGR chlauth connauth
DIS AUTHINFO(WM253.G1.AUTHINFO) CHCKCLNT CHCKLOCL
*
* End of lockSecurity.txt script
*
*-----

```

__ a. A new AUTHINFO record is created where connection authentication is required for

both local and client connections. Do not confuse connection authentication with channel authentication. Before this record is current in the queue manager, the queue manager CONNAUTH attribute must be changed to reflect this record. Connection authentication security must be refreshed in the queue manager (or the queue manager restarted). An alternative is to update the existing AUTHINFO CHCKCLNT and CHCKLOCL fields; however, it is preferable to leave the original system settings unchanged.

- ___ b. Next, queue manager CONNAUTH attribute is updated to use the new AUTHINFO record, and channel authentication is enabled.
- ___ c. The next command sets the back-stop rule. Observe how any generic items, such as the asterisk, are enclosed in single quotation marks. Any time an asterisk is any part of an attribute value, it must be enclosed in single quotation marks.
- ___ d. In the back-stop rule, pay particular attention to the action, `USERSRC(NOACCESS)`. ***You might need to remember this action to identify a channel authentication challenge when you look at the queue manager log later in this exercise.***
- ___ e. The next command refreshes the queue manager connection authentication setting so that it uses the new AUTHINFO record. An alternative to the `REFRESH` command is to restart the queue manager.
- ___ f. Next, the CHLAUTH and then the CONNAUTH attributes are displayed.
- ___ g. Last, the values of the CHCKCLNT and CHCKLOCL attributes for the newly activated AUTHINFO record are displayed.



Stop

The connection and channel authentication settings that are used in this exercise are selected for demonstrating the capabilities of the features and how to use them. ***The configuration in this exercise is not to be construed as a best practice.*** It is preferred that channel authentication and authorization are ***not*** disabled. However, the security configuration in your organization must adhere to the organization and application requirements and standards.

When working with IBM MQ clusters, security configuration is applied to the entire cluster, not just one queue manager. The objective of this exercise is not to secure all queue managers in the cluster, but to show you how to identify and adjust the configuration for stricter security settings in one queue manager. You can apply these skills to the rest of the queue managers in the cluster according to your organizational requirements.

In this exercise, one generic channel authentication rule is used to allow all cluster queue managers to connect. This rule might not be the standard in your organization. The work in this exercise is exclusively to illustrate how to recognize the need for, and set the channel authentication rules.

Section 2: Apply the security changes to MQG1 by running the lockSecurity.mqsc script

- ___ 16. From the command prompt, make the security changes coded in the script by typing the command as shown on the text box, and press the Enter key. Ensure that you capture the results in a file so that you can review that each command in the script completed successfully. **Ensure that you are in the C:\LabFiles\Unit5> directory before you type the command.** Do not be concerned about overwriting the MQG1.out file. You do not need the old results.

Optionally, you can copy the command from the Lab5_copyAndPaste.txt file.

```
runmqsc MQG1 < lockSecurity.mqsc > MQG1.out
```

All expected output goes to the MQG1.out file.

- ___ 17. From the same directory in the command prompt window, check the runmqsc utility results.
- ___ a. Open the MQG1.out file by typing notepad MQG1.out and pressing the Enter key.
- ___ b. Scroll to the end of the output. You are expected to see the results as displayed in the text box.
-

```
... .. (Partial results are displayed)
*-----
      : *
      : * Display the changes in connection authentication and channel
authentication
      : *
      5 : DIS QMGR chlauth connauth
AMQ8408: Display Queue Manager details.
      QMNAME(MQG1)                      CHLAUTH(ENABLED)
      CONNAUTH(WM253.G1.AUTHINFO)
      6 : DIS AUTHINFO(WM253.G1.AUTHINFO) CHCKCLNT CHCKLOCL
AMQ8566: Display authentication information details.
      AUTHINFO(WM253.G1.AUTHINFO)       AUTHTYPE(IDPWOS)
      CHCKCLNT(REQUIRED)                 CHCKLOCL(REQUIRED)
      : *
      : * End of lockSecurity.txt script
      : *
      : *****
6 MQSC commands read.
No commands have a syntax error.
All valid MQSC commands were processed.
```

- ___ c. In the queue manager display, CHLAUTH is expected to display ENABLED, and CONNAUTH is expected to reflect the WM253.G1.AUTHINFO AUTHINFO record.

- ___ d. The `WM253.G1.AUTHINFO` record display is expected to show the value in both `CHCKCLNT` and `CHCKLOCL` attributes as `REQUIRED`.
 - ___ e. If you see 6 MQSC commands read and No commands have a syntax error, continue to the next numbered step.
 - ___ 18. If the results were not as expected, review the output and correct any errors before you proceed.
 - ___ 19. Close file `MQG1.out`.
-

**Stop**

At the time this exercise was written, you must restart the queue manager to ensure that the channel authentication back-stop rule is enabled.

Ensure that you complete the next two steps to stop and start queue manager MQG1 before you proceed.

- ___ 20. End queue manager MQG1 by typing `endmqm MQG1` and pressing the Enter key.
- ___ 21. As soon as MQG1 ends, start the queue manager by typing `strmqm MQG1` and pressing the Enter key.
- ___ 22. Keep the command prompt window open.

4.3. Check the environment after the security changes, identify the cause of the 2035, and implement correct adjustments



Hint

You are expected to encounter different security challenges that are caused by either connection authentication, or channel authentication. You must carefully review the most recent queue manager log entries for the reason, as you might have multiple challenges.

- Since MQG1 is the queue manager that hardened the security, you start your investigation with the MQG1 queue manager log.
- Channel authentication challenges normally include the word “blocked” in the log entries. After you resolved any connection authentication issues, when you find “blocked” in the queue manager log entry, you have a channel authentication challenge.

Section 1: Repeat the sending of messages from MQE5 to the WM253.IN queues via queue remote TO.GATEWAY at MQE5

- ___ 23. Use sample program `amqspout` to send the messages from queue manager E5 to the G1 gateway to reach the cluster queues by typing the command exactly as shown. Be careful to use the correct queue and queue manager names.

```
amqspout TO.GATEWAY MQE5 < many_messages.txt
```

Expected results:

```
Sample AMQSPUT0 start
target queue is TO.GATEWAY
Sample AMQSPUT0 end
```

Section 2: Check whether the messages reached the WM253.IN queues

- ___ 24. From your command prompt window, use the `amqsget` sample program as shown in the text box. **No messages are expected to have reached the queue.** You can either wait until `amqsget` ends, or use CTL-C to end the wait.

```
amqsget WM253.IN MQ0A
Sample AMQSGET0 start
no more messages
Sample AMQSGET0 end
```

```
amqsget WM253.IN MQ03
Sample AMQSGET0 start
^C
```

Section 3: Check the status of sender channel MQE5.MQG1

- __ 25. From your command prompt window, start a runmqsc session for queue manager MQE5 by typing `runmqsc MQE5` and pressing the Enter key.
 - __ 26. Display the channel status by typing the command as shown in the window, and press the Enter key.
-

```
dis chs(MQE5*)
```

Expected results:

```
1 : dis chs(MQE5*)
AMQ8417: Display Channel Status details.
CHANNEL(MQE5.MQG1)                CHLTYPE(SDR)
CONNAME(localhost(1651))           CURRENT
RQMNAME( )                         STATUS(RETRYING)
SUBSTATE( )                        XMITQ(MQG1)
```



Troubleshooting

Since the channel to queue manager MQG1 is in retry status, start by looking at the MQG1 queue manager log for clues on the problem.

- __ 27. Type `end` and press the Enter key to close the runmqsc session.
- __ 28. Use Windows Explorer to navigate to the MQG1 queue manager log at `C:\ProgramData\IBM\MQ\qmgrs\MQG1\errors`.
- __ 29. Use your preferred editor to open queue manager log file `AMQERR01.LOG` and scroll to the end of the file.

- ___ 30. Look for an entry towards the end of the log with message AMQ9999 and then find the immediately preceding entry. Your results are expected to resemble the text box display.
Check the comments that follow the text box.

```
----- amqrmrsa.c : 938 -----
5/29/2017 10:29:53 - Process(5216.1004) User(MUSR_MQADMIN) Program(amqrmppa.exe)
                        Host(WS2008R2X64) Installation(IBM MQV9)
                        VRMF(9.0.0.0) QMgr(MQG1)
```

AMQ9777: Channel was blocked

EXPLANATION:

The inbound channel 'MQE5.MQG1' was blocked from address 'ws2008r2x64 (127.0.0.1)' because the active values of the channel matched a record **configured with USERSRC(NOACCESS). The active values of the channel were 'QMNAME(MQE5) ADDRESS(ws2008r2x64)'**.

ACTION:

Contact the systems administrator, who should examine the channel authentication records to ensure that the correct settings have been configured. The ALTER QMGR CHLAUTH switch is used to control whether channel authentication records are used. The command DISPLAY CHLAUTH can be used to query the channel authentication records.

```
----- cmqxrmrsa.c : 1476 -----
5/29/2017 10:29:53 - Process(5216.1004) User(MUSR_MQADMIN) Program(amqrmppa.exe)
                        Host(WS2008R2X64) Installation(IBM MQV9)
                        VRMF(9.0.0.0) QMgr(MQG1)
```

AMQ9999: Channel 'MQE5.MQG1' to host 'ws2008r2x64 (127.0.0.1)' ended abnormally.

EXPLANATION:

The channel program running under process ID 5216(2220) for channel 'MQE5.MQG1' ended abnormally. The host name is 'ws2008r2x64 (127.0.0.1)'; in some cases the host name cannot be determined and so is shown as '????'.

ACTION:

Look at previous error messages for the channel program in the error logs to determine the cause of the failure. Note that this message can be excluded completely or suppressed by tuning the "ExcludeMessage" or "SuppressMessage" attributes under the "QMErrorLog" stanza in qm.ini. Further information can be found in the System Administration Guide.

```
----- amqrmrsa.c : 938 -----
```

-
- ___ a. The AMQ9999 message informs you that channel MQE5.MQG1 ended abnormally, but you need extra details. Check the message that immediately precedes this message. The date and time stamps of the message should be the same, or the time should be within tenths of a second.
- ___ b. You see the entry "channel was blocked", which signals a channel authentication challenge.
- ___ c. The next piece of information, in bold, is that the channel authentication record that caused the problem contained a USERSRC(NOACCESS) rule. The back-stop rule contained a USERSRC(NOACCESS).
-

- ___ d. Read the rest of the message, and the key information is “inbound channel”. Later in the log entry you see that the queue manager that was blocked was MQE5. Since the channel named MQE5.MQG1 at the MQG1 queue manager is a receiver channel, you know that the channel blocked was the MQE5 sender channel, MQE5.MQG1.
- ___ 31. Close the MQG1 log file.

Section 4: Attempt a runmqsc session for queue manager MQG1 after the security commands were applied



Important

If you have any open MQG1 `runmqsc` sessions, you must close them before you start this section. ***If you do not start a new MQG1 runmqsc session, the steps in this section do not work as expected.***

- ___ 32. You need to create a channel authentication rule on queue manager MQG1 to allow the connection from cluster queue manager MQ0B. From a command prompt window, start the `runmqsc` session by typing `runmqsc MQG1` and pressing the Enter key. **Your attempt to use `runmqsc` is expected to fail**, as illustrated in the text box.

5724-H72 (C) Copyright IBM Corp. 1994, 2016.

Starting MQSC for queue manager MQG1.

AMQ8135: Not authorized.

No MQSC commands read.

No commands have a syntax error

- ___ 33. Return to the MQG1 queue manager log to determine the cause of the `runmqsc` failure. Use Windows Explorer to navigate to the MQG1 queue manager log at `C:\ProgramData\IBM\MQ\qmgrs\MQG1\errors`.
- ___ 34. Use your preferred error to open file `AMQERR01.LOG` and scroll to the end of the file.

- ___ 35. Look for an entry at or near the end of the log file that might offer a clue as to the reason for the AMQ8135: Not authorized response to the runmqsc command.

```
----- amqrmrsa.c : 938 -----
5/29/2017 10:57:00 - Process(2244.1082) User(MUSR_MQADMIN) Program(amqzlaa0.exe)
                        Host(WS2008R2X64) Installation(IBM MQV9)
                        VRMF(9.0.0.0) QMgr(MQG1)

AMQ5540: Application 'les\IBM\MQ\bin64\runmqsc.exe' did not supply a user ID
and password
EXPLANATION:
The queue manager is configured to require a user ID and password, but none was
supplied.
ACTION:
Ensure that the application provides a valid user ID and password, or change
the queue manager configuration to OPTIONAL to allow applications to connect
which have not supplied a user ID and password.
----- amqzfuca.c : 4751 -----
```



Note

It is expected that you find an entry at the end of the MQG1 log with the AMQ5540 message. You see that the entry identifies `runmqsc`, and states that it did not supply an ID and password. It also explains that the queue manager is configured to require credentials.

The `runmqsc` command has a `-u` parameter where you can provide an ID. When you use `-u` to provide a user ID to `runmqsc`, you are prompted for the password.

- ___ 36. Retry the `runmqsc` session by including the `-u` parameter as detailed in these steps:
- ___ a. On your command prompt window, start the `runmqsc` session by typing `runmqsc -u administrator MQG1` and pressing the Enter key. You are prompted for the password.
 - ___ b. At the password prompt, type `web1sphere` and press the Enter key. (The password is obfuscated in the display.)
 - ___ c. The start of the `runmqsc` session is expected to resemble the text box. **Do not** close the `runmqsc` session.

```
runmqsc -u administrator MQG1
```

Expected results:

```
5724-H72 (C) Copyright IBM Corp. 1994, 2016.
```

```
Enter password:
```

```
*****
```

```
Starting MQSC for queue manager MQG1.
```


Section 5: Create a channel authentication rule at MQG1 to allow MQE5 to use channel MQE5.MQG1

- ___ 37. Count how many channel authentication rules you have at queue manager MQG1 before you attempt to add a rule by typing the command as shown in the text box, and press the Enter key. Keep the `runmqsc` session open.

```
dis chlauth(*) descr
```

Expected results:

```
2 : dis chlauth(*) descr
AMQ8878: Display channel authentication record details.
      CHLAUTH(SYSTEM.ADMIN.SVRCONN)          TYPE(ADDRESSMAP)
      DESCR(Default rule to allow MQ Explorer access)
      ADDRESS(*)                             USERSRC(CHANNEL)
AMQ8878: Display channel authentication record details.
      CHLAUTH(SYSTEM.*)                      TYPE(ADDRESSMAP)
      DESCR(Default rule to disable all SYSTEM channels)
      ADDRESS(*)                             USERSRC(NOACCESS)
AMQ8878: Display channel authentication record details.
      CHLAUTH(*)                             TYPE(ADDRESSMAP)
      DESCR(Back-stop rule)                  ADDRESS(*)
      USERSRC(NOACCESS)
AMQ8878: Display channel authentication record details.
      CHLAUTH(*)                             TYPE(BLOCKUSER)
      DESCR(Default rule to disallow privileged users)
      USERLIST(*MQADMIN)
```



Note

When you are new, it is a good habit to count the rules before and after you add a rule. This practice is especially important if you change an existing rule. Sometimes a change request, if not adequately qualified, results in the addition of a new record.

-
- ___ 38. Review your results. Expect to find four rules as shown in the text box: the three original default rules, plus the back stop rule that you created with the script. If your results are different, request some assistance to remedy the situation before you continue.
- ___ 39. From your MQG1 `runmqsc` session, type the channel authentication rule as shown in the text box, and press the Enter key. Optionally, you can copy and paste the command from the `Lab4_copyAndPaste.txt` file at the `C:\LabFiles\Unit5` directory. **Do not copy and paste from this lab guide as you might introduce special characters that introduce errors.** If you type the channel authentication rule, type it all in one line, and do not press the Enter key until the end of the command. **Keep the `runmqsc` session open.**

**Note**

The rule is typed in a contiguous line, without breaks. If copying the command, ensure that you copy the entire line, without line breaks. Keep the runmqsc session open.

If you use copy-and-paste, copy the rule as shown in the text box exclusively. Do not copy any other rules now.

```
SET CHLAUTH('MQE5.MQG1') TYPE(ADDRESSMAP) ADDRESS('127.0.0.1')
MCAUSER('administrator') ACTION(REPLACE)
DESCR('Allow administrator to connect to MQG1')
```

Expected response:

```
1 : SET CHLAUTH('MQE5.MQG1') TYPE(ADDRESSMAP) ADDRESS('127.0.0.1') MCAUSER(
'administrator') ACTION(REPLACE) DESCR('Allow administrator to connect to
MQG1')
```

AMQ8877: IBM MQ channel authentication record set.

-
- ___ 40. If you see the AMQ8877: IBM MQ channel authentication record set response, continue to the next step. If your command resulted in an error, check your typing and correct the problem before you continue.
 - ___ 41. While still in the MQG1 runmqsc session, display the status for channel MQE5.MQG1 by typing the command as shown in the text box to determine whether the rule resolved the retrying blocked channel.
-

```
dis chs(MQE5*)
```

Expected response:

```
2: dis chs(MQE5*)
```

AMQ8417: Display Channel Status details.

CHANNEL(MQE5.MQG1)	CHLTYPE(RCVR)
CONNAME(127.0.0.1)	CURRENT
RQMNAME(MQE5)	STATUS(RUNNING)
SUBSTATE(RECEIVE)	

- ___ 42. Close the MQG1 runmqsc session by typing `end` and pressing the Enter key.
- ___ 43. If the receiver MQE5.MQG1 channel did not return to running status, take the action that is shown in the Troubleshooting box.



Troubleshooting

If receiver channel MQE5.MQG1 in queue manager MQG1 did not return to running status, take these steps. **If the channel was in running status in the previous step, skip this Troubleshooting box.**

Troubleshooting box.

- Open a runmqsc session for queue manager MQE5 by typing `runmqsc MQE5` and pressing the Enter key.
- Stop sender channel MQE5.MQG1 by typing `stop chl(MQE5.MQG1)` and pressing the Enter key.
- Start sender channel MQE5.MQG1 by typing `start chl(MQE5.MQG1)` and pressing the Enter key.
- Confirm that the channel started by typing `dis chs(MQE*)` and pressing the Enter key. The status is expected to be running.

__ 44. Repeat the `amqsget WM253.IN MQ0A` and `amqsget WM253.IN MQ03`. You should now see the messages in queues.



Note

The messages from MQE5 are going from MQE5 to MQG1, for which the new rule resolved the channel authentication challenge. Messages that are outbound from MQG1 to the other cluster member queue managers continue to flow. However, you see issues with channels inbound to queue manager MQG1.

Section 6: Determine the behavior of the cluster after MQG1 has hardened security



Stop

Depending on the sequence of actions that are taken in the queue manager, ***your results from this point might, or might not, be exactly as shown in the next few sections.***

The key point is that when you have hardened connection and channel authentication, you must ensure that the cluster is communicating.

- So far you resolved the problem with the sender-receiver channel from the MQE5 queue manager outside the cluster, to MQG1.
- An extra channel authentication rule is required to allow the other cluster queue managers to connect to MQG1. The rule involves all channels to MQG1, which means WMADMCLS.MQG1 for each member queue manager.

If your results are not the same, assuming that channel MQE5.MQG1 resumed correctly after you added the first rule, you can proceed to the following section. In this section, you add the second rule to resolve MQG1 security issues with the rest of the cluster queue managers.

-
- ___ 45. Open a `runmqsc` session to queue manager MQ03 by typing `runmqsc MQ03` and pressing the Enter key.
 - ___ 46. Display the cluster channels from queue manager MQ03 by typing the `DIS CLUSQMGR` command as shown in the text box, and press the Enter key. **Read the notes that follow the display.**
-

```
dis clusqmgr(*) deftype status
```

Expected reply:

```
2 : dis clusqmgr(*) deftype status
AMQ8441: Display Cluster Queue Manager details.
      CLUSQMGR(MQ03)                CHANNEL(WMADMCLS.MQ03)
      CLUSTER(WMADMCLS)             DEFTYPE(CLUSRCVR)
      STATUS(RUNNING)
AMQ8441: Display Cluster Queue Manager details.
      CLUSQMGR(MQ0A)                CHANNEL(WMADMCLS.MQ0A)
      CLUSTER(WMADMCLS)             DEFTYPE(CLUSSDRA)
      STATUS(RUNNING)
AMQ8441: Display Cluster Queue Manager details.
      CLUSQMGR(MQ0B)                CHANNEL(WMADMCLS.MQ0B)
      CLUSTER(WMADMCLS)             DEFTYPE(CLUSSDRB)
      STATUS(RUNNING)
```

- ___ 47. Review the output. You see the CLUSRCVR channel for MQ03 running, and you see dynamic cluster sender channels to MQ0A and MQ0B also RUNNING, so **at first it appears as a good display. However, there is no WMQDMCLS.MQG1 CLUSSDRA dynamic cluster sender channel to MQG1.**
- ___ 48. Type `end` to exit the `runmqsc` session for MQ03 and press the Enter key.
- ___ 49. Open a `runmqsc` session to full repository queue manager MQ0B by typing `runmqsc MQ0B` and pressing the Enter key.
- ___ 50. Display the cluster channels from queue manager MQ0B by typing the `DIS CLUSQMGR` command as shown in the text box, and press the Enter key. Read the notes that follow the display.

```
dis clusqmgr(*) deftype status
```

Expected response (might vary):

```
1 : dis clusqmgr(*) deftype status
AMQ8441: Display Cluster Queue Manager details.
      CLUSQMGR(MQ03)                      CHANNEL(WMADMCLS.MQ03)
      CLUSTER(WMADMCLS)                   DEFTYPE(CLUSSDRA)
      STATUS(RUNNING)
AMQ8441: Display Cluster Queue Manager details.
      CLUSQMGR(MQ0A)                      CHANNEL(WMADMCLS.MQ0A)
      CLUSTER(WMADMCLS)                   DEFTYPE(CLUSSDRB)
      STATUS(RUNNING)
AMQ8441: Display Cluster Queue Manager details.
      CLUSQMGR(MQ0B)                      CHANNEL(WMADMCLS.MQ0B)
      CLUSTER(WMADMCLS)                   DEFTYPE(CLUSRCVR)
      STATUS(RUNNING)
AMQ8441: Display Cluster Queue Manager details.
      CLUSQMGR(MQG1)                      CHANNEL(WMADMCLS.MQG1)
      CLUSTER(WMADMCLS)                   DEFTYPE(CLUSSDRA)
      STATUS(RETRYING)
```

- ___ 51. Review the display. You looked at the list from a full repository queue manager. You see that channels are running among the queue managers in the cluster **except** to queue manager MQG1. Cluster sender CLUSSDRA channel WMADMCLS.MQG1 to the MQG1 queue manager is in retry. Although the rule mitigated the external MQE5 channel to MQG1, the cluster channels from the other queue managers to MQG1 are challenged.
- ___ 52. Type `end` and press the Enter key to close the runmqsc session.

Section 7: Confirm the reason that the WMADMCLS.MQG1 channel is in retry by checking the MQG1 queue manager error log

- ___ 53. Use Windows Explorer to navigate to the MQG1 queue manager log at
C:\ProgramData\IBM\MQ\qmgrs\MQG1\errors.
 - ___ 54. Use your preferred editor to open queue manager log file `AMQERR01.LOG` and scroll to the end of the file.
 - ___ 55. Look for an entry towards the end of the log with message `AMQ9999` and then find the immediately preceding entry. Your results are expected to resemble the text box display.
Check the comments that follow the text box. Review the notes that follow the display.
-



Note

You might need to repeat the search to get to the correct entry, which should display a date and time stamp closer to when the connection was attempted, after the security was hardened.

```
5/29/2017 12:43:12 - Process(1816.15) User(MUSR_MQADMIN) Program(amqrmppa.exe)
                        Host(WS2008R2X64) Installation(IBM MQV9)
                        VRMF(9.0.0.0) QMgr(MQG1)
```

AMQ9777: Channel was blocked

EXPLANATION:

The inbound channel 'WMADMCLS.MQG1' was blocked from address 'ws2008r2x64 (127.0.0.1)' because the active values of the channel matched a record configured with USERSRC(NOACCESS). The active values of the channel were 'QMNAME(MQ0B) ADDRESS(ws2008r2x64)'.

ACTION:

Contact the systems administrator, who should examine the channel authentication records to ensure that the correct settings have been configured. The ALTER QMGR CHLAUTH switch is used to control whether channel authentication records are used. The command DISPLAY CHLAUTH can be used to query the channel authentication records.

- __ 56. Review the message. You see several pieces of information:
 - __ a. You see confirmation of channel WMADMCLS.MQG1.
 - __ b. You see that the reason the channel is blocked is due to a channel authentication record with USERSRC(NOACCESS). This record is the back-stop rule. The words rule and record are used interchangeably.
 - __ c. You then see confirmation that the queue manager that attempted this connection was MQ0B, from which you saw the RETRYING status to MQG1.
- __ 57. Close the MQG1 log file.

Section 8: Create a channel authorization record to allow connectivity with the cluster queue managers

- __ 58. Open a runmqsc session for queue manager MQG1 by typing the command as shown in the text box, and press the Enter key. You are prompted for the password.
-

```
runmqsc -u administrator MQG1
```

- __ 59. When prompted, type the password `webSphere` and press the Enter key.
- __ 60. Determine how many channel authentication rules you have at queue manager MQG1 before you attempt to add a rule by typing `DIS CHLAUTH(*)` and pressing the Enter key. **Keep the runmqsc session open.**
- __ 61. Count the channel authentication records. You should have five, composed of the three default rules, the back-stop rule, and the MQE5.MQG1 rule to allow MQE5 to connect to MQG1.

- ___ 62. Copy and paste the command to add the channel authentication record from the Lab4_copyAndPaste.txt file at the C:\LabFiles\Unit5 directory. **Do not copy and paste from this lab guide as you might introduce special characters that introduce errors.** If you type the channel authentication rule, type it all in one line, and do not press the Enter key until the end of the command. **Keep the runmqsc session open.**



Note

The rule is typed in a contiguous line, without breaks. If copying the command, ensure that you copy the entire line, without line breaks. Keep the MQG1 runmqsc session open.

```
SET CHLAUTH('WMADMCLS.*') TYPE(QMGRMAP) ADDRESS('127.0.*')
DESCR('Access for WMADMCLS cluster channels') MCAUSER('administrator')
QMNAME('MQ*')
```

Expected response:

```
1 : SET CHLAUTH('WMADMCLS.*') TYPE(QMGRMAP) ADDRESS('127.0.*')
DESCR('Access for WMADMCLS cluster channels') MCAUSER('administrator')
QMNAME('MQ*')
AMQ8877: IBM MQ channel authentication record set.
```

- ___ 63. Ensure that you see message AMQ8877 to confirm that the rule was set.
- ___ 64. Determine how many channel authentication rules you have at queue manager MQG1 before you attempt to add a rule by typing `DIS CHLAUTH(*)` and pressing the Enter key. **Keep the runmqsc session open.**
- ___ 65. Count the channel authentication records. You should have six records now, including the rule that you created.
- ___ 66. Type `end` and press the Enter key to close the `runmqsc` session.

Section 9: Check that the new rule resolved connectivity issues in the other cluster member queue managers



Attention

Your results might not be exactly the same as all results in this exercise. What you need to do is check each cluster queue manager and ensure that no CLUSSDRA or CLUSSDRB type channels are in retry. If you find a CLUSSDRA or CLUSSDRB type channel with RETRYING status, stop and restart the channel that is in retrying status.

The example that follows shows how to process one of the cluster member queue managers. You repeat a similar process as needed in the other cluster member queue managers.

Do not use this process in a CLUSRCVR type channel. As with regular channels, you normally start and stop sender type channels exclusively. With a cluster, these channels are CLUSSDRA and CLUSSDRB type channels.

-
- ___ 67. Start a `runmqsc` session for queue manager MQ0A by typing `runmqsc MQ0A` and pressing the Enter key.
 - ___ 68. Display the cluster queue managers by typing the command as shown in the text box, and press the Enter key. Keep the `runmqsc` session open.
-

```
dis clusqmgr(*) deftype status
```

Expected response:

```
1 : dis clusqmgr(*) deftype status
AMQ8441: Display Cluster Queue Manager details.
    CLUSQMGR(MQ03)                      CHANNEL(WMADMCLS.MQ03)
    CLUSTER(WMADMCLS)                   DEFTYPE(CLUSSDRA)
    STATUS(INACTIVE)
AMQ8441: Display Cluster Queue Manager details.
    CLUSQMGR(MQ0A)                      CHANNEL(WMADMCLS.MQ0A)
    CLUSTER(WMADMCLS)                   DEFTYPE(CLUSRCVR)
    STATUS(RUNNING)
AMQ8441: Display Cluster Queue Manager details.
    CLUSQMGR(MQ0B)                      CHANNEL(WMADMCLS.MQ0B)
    CLUSTER(WMADMCLS)                   DEFTYPE(CLUSSDRB)
    STATUS(INACTIVE)
AMQ8441: Display Cluster Queue Manager details.
    CLUSQMGR(MQG1)                      CHANNEL(WMADMCLS.MQG1)
    CLUSTER(WMADMCLS)                   DEFTYPE(CLUSSDRA)
    STATUS(RETRYING)
```

- ___ 69. Stop CLUSSDRA channel WMADMCLS.MQG1 by typing `stop chl(WMADMCLS.MQG1)` and pressing the Enter key. Keep the `runmqsc` session open.
- ___ 70. Start CLUSSDRA channel WMADMCLS.MQG1 by typing `start chl(WMADMCLS.MQG1)` and pressing the Enter key. Keep the `runmqsc` session open.
- ___ 71. Repeat the display by typing `dis clusqmgr(*) deftype status` and pressing the Enter key.
- ___ 72. Check the status for CLUSSDRA channel WMADMCLS.MQG1. It is expected that it is now in running status.
- ___ 73. Exit the MQ0A `runmqsc` session by typing `end` and pressing the Enter key.
- ___ 74. Repeat the process that is used for queue manager MQ0A in queue manager MQ0B.
- ___ 75. Repeat the process that is used for queue manager MQ0A in queue manager MQ03.

**Information**

When you have the channel authentication back-stop rule set, you need to add rules for other connections. In this exercise, you added rules relevant to cluster gateway and cluster connections exclusively.

You completed the connection authentication and channel authentication parts of this exercise.

4.4. Part 3: Work with basic object authorizations

In this part of the lab exercise, you use control commands `dspmqaaut` and `setmqaut` to review and set basic object authorizations. You also use MQSC commands `DISPLAY AUTHREC` and `SET AUTHREC` to accomplish the same objectives as with the control commands. You also observe the effects of required connection authentication settings in a queue manager when you use the control commands (`dspmqaaut` and `setmqaut`).

Section 1: Use `dspmqaaut` to display the authorizations of a user for a queue in a queue manager with optional connection authentication

- ___ 76. Close any open command prompt windows.
- ___ 77. Open a new command prompt window.
- ___ 78. Use the `dspmqaaut` control command to display the authorizations of the `administrator` user for cluster queue `MQ03.ONLY`. Type the command as shown in the text box. **Do not copy the command from the user guide as it might contain extraneous characters and fail.**

```
dspmqaaut -m MQ03 -n MQ03.ONLY -t q -p administrator
```

Expected results:

Entity administrator has the following authorizations for object MQ03.ONLY:

```
get
browse
put
inq
set
crt
dlt
chg
dsp
passid
passall
setid
setall
clr
```

Section 2: Use `DIS AUTHREC` to display the authorizations of an unprivileged user for a queue in a queue manager with optional connection authentication

- ___ 79. Open a `runmqsc` session for queue manager `MQ03` by typing `runmqsc MQ03` and pressing the Enter key.

-
- __ 80. Use the `DISPLAY AUTHREC MQSC` command to display the authorizations of the `fileusr1` user for cluster queue `MQ03.ONLY`. Type the command as shown in the text box. Do not copy the command from the user guide as it might contain extraneous characters. You must spell out `queue` for the `objtype` parameter. **Leave the `runmqsc` session open.**
-

```
dis authrec profile(MQ03.ONLY) objtype(queue) principal(fileusr1)
```

Expected results:

```
1 : dis authrec profile(MQ03.ONLY) objtype(queue) principal(fileusr1)
AMQ8459: Not found.
```



Note

When you use the `DIS AUTHREC` command, a “not found” indicates that a profile for user `fileusr1` was not found. The “not found” message is not referring to the existence of queue `MQ03`.

After you grant `put` authorization to queue `MQ03` for user `fileusr1`, the profile is displayed.

- __ 81. Grant user `fileusr1` `put` authority to queue `MQG1` by typing the command as shown in the text box. Do not copy the command from the user guide as it might contain extraneous characters. You must spell out `queue` for the `objtype` parameter. **Leave the `runmqsc` session open. Type the command in a continuous line, and do not press Enter until the end of the command.**
-

```
set authrec profile(MQ03.ONLY) objtype(queue) principal(fileusr1) authadd(put)
```

Expected results:

```
2 : set authrec profile(MQG1) objtype(queue) principal(fileusr1) authadd(put)
AMQ8862: IBM MQ authority record set.
```

- __ 82. Repeat the `DIS AUTHREC MQSC` command by typing the command as shown in the text box. Do not copy the command from the user guide as it might contain extraneous characters. You must spell out `queue` for the `objtype` parameter. **Leave the `runmqsc` session open.**
-

```
dis authrec profile(MQ03.ONLY) objtype(queue) principal(fileusr1)
```

Expected response:

```
3 : dis authrec profile(MQ03.ONLY) objtype(queue) principal(fileusr1)
AMQ8864: Display authority record details.
PROFILE(MQG1)                                ENTITY(fileusr1@WS2008R2X64)
ENTITYTYPE(PRINCIPAL)                        OBJTYPE(QUEUE)
AUTHLIST(PUT)
```

- __ 83. While you are in the `runmqsc` session, refresh the object authority cache by typing the command as shown in the text box. Do not copy the command from the user guide as it might contain extraneous characters.

```
refresh security type(authserv)
```

Expected response:

```
4 : refresh security type(authserv)
AMQ8560: IBM MQ security cache refreshed.
```

- __ 84. Exit the `runmqsc` session by typing `end` and pressing the Enter key.

Section 3: Use `dspmqaut` to display the authorization of user `fileusr1` to connect to queue manager `MQG1`, which has the required connection authentication

- __ 85. Use the `dspmqaut` control command to display the authorizations of the `fileusr1` user to connect to queue manager `MQG1`. Type the command as shown in the text box. Do not copy the command from the user guide as it might contain extraneous characters.

```
dspmqaut -m MQG1 -t qmgr -p fileusr1
```

Expected results:

```
AMQ7077: You are not authorized to perform the requested operation.
```

- __ 86. Use Windows Explorer to navigate to the `MQG1` queue manager logs at `C:\ProgramData\IBM\MQ\qmgrs\MQG1\errors`.
- __ 87. Double-click file **`AMQERR01.LOG`** to open it.
- __ 88. Scroll to the end of the file, and then look for an entry that mentions use of the `dspmqaut` command. The expected entry is displayed in the text box.

```
5/31/2017 09:04:58 - Process(4888.39) User(MUSR_MQADMIN) Program(amqzlaa0.exe)
                        Host(WS2008R2X64) Installation(IBMMQV9)
                        VRMF(9.0.0.0) QMgr(MQG1)
```

AMQ5540: Application 'es\IBM\MQ\bin64\dspmqaut.exe' did not supply a user ID and password

EXPLANATION:

The queue manager is configured to require a user ID and password, but none was supplied.

ACTION:

Ensure that the application provides a valid user ID and password, or change the queue manager configuration to `OPTIONAL` to allow applications to connect which have not supplied a user ID and password.

**Note**

The actions that are suggested in the error message are misleading when connection authentication is required in a queue manager. **This text box is exclusively informational. Numbered steps follow.**

When connection authentication is required, object authorizations can be displayed and set by using either of the following options:

- Restarting the queue manager with the `-ns` parameter
- Using the MQSC commands `DIS AUTHREC` or `SET AUTHREC`

First, you use the queue manager that restarted with the `-ns` parameter. However, you first must stop the processes that are active in the queue manager.

-
- __ 89. Stop queue manager MQG1 by typing `endmqm -i MQG1` and pressing the Enter key.
 - __ 90. Restart queue manager MQG1 with the `-ns` parameter by typing `strmqm -ns MQG1` and pressing the Enter key.
 - __ 91. After you restart the queue manager, repeat the `dspmqaut` control command to display the authorizations of the `fileusr1` user to connect to queue manager MQG1. Type the command as shown in the text box. Do not copy the command from the lab exercise guide as it might contain extraneous characters.
-

```
dspmqaut -m MQG1 -t qmgr -p fileusr1
```

Expected response:

Entity `fileusr1` has the following authorizations for object MQG1:

-
- __ 92. The previous step showed that user `fileusr1` did not have any authorizations for the queue manager type object in queue manager MQG1. You now add connect authorization.

Section 4: Grant user `fileusr1` connect authority to queue manager MQG1

-
- __ 93. From your command prompt window, type the command as shown in the text box to grant `fileusr1` authority to connect to queue manager MQG1.
-

```
setmqaut -m MQG1 -t qmgr -p fileusr1 +connect
```

Expected results:

The `setmqaut` command completed successfully.

Section 5: Check the results of the `setmqaut` by using `dis authrec` from a `runmqsc` session

-
- __ 94. Stop queue manager MQG1 by typing `endmqm -i MQG1` and pressing the Enter key.

- ___ 95. Restart queue manager MQG1 by typing `strmqm MQG1` and pressing the Enter key.
- ___ 96. Open a `runmqsc` session by typing `runmqsc -u administrator MQG1` and pressing the Enter key.
- ___ 97. When prompted, provide the password as `webSphere` and press the Enter key.
- ___ 98. Display the authorizations for queue manager object MQG1 for user fileusr1 by typing the command as shown in the text box, and press the Enter key.

```
dis authrec objtype(qmgr) principal(fileusr1)
```

Expected response:

```
1 : dis authrec objtype(qmgr) principal(fileusr1)
AMQ8864: Display authority record details.
  PROFILE(SELF)                      ENTITY(fileusr1@WS2008R2X64)
  ENTTYPE(PRINCIPAL)                 OBJTYPE(QMGR)
  AUTHLIST(CONNECT)
AMQ8864: Display authority record details.
  PROFILE(@CLASS)                    ENTITY(fileusr1@WS2008R2X64)
  ENTTYPE(PRINCIPAL)                 OBJTYPE(QMGR)
  AUTHLIST(NONE)
```



Note

As you might observe, it is more labor intensive to use the control commands when connection authentication is required. You might prefer to use the MQSC commands, by using `runmqsc` with the `dis authrec` and `set authrec` commands. Use of MQSC commands is also best if you need to refresh the object authorities cache because `REFRESH SECURITY` is available as an MQSC command, and with IBM MQ Explorer, but not as a control command.

End of exercise

Exercise review and wrap-up

In this lab exercise, you:

- Reviewed the commands that are used to harden connection authentication and channel authentication
- Hardened connection authentication and channel authentication in a queue manager
- Created a type ADDRESSMAP channel authentication record to allow a specific queue manager to connect to a queue manager with strict channel authentication
- Differentiated between connection authentication and channel authentication entries in the queue manager log
- Created a type QMGRMAP channel authentication record to allow the cluster member queue managers to interact with a cluster queue manager with strict channel authentication
- Invoked the runmqsc utility with credentials to access a queue manager with required connection authentication
- Used the dspmqaut and setmqaut control commands to display and set object authorities
- Used the MQSC DISPLAY AUTHREC and SET AUTHREC commands to display and set object authorities
- Used control commands in a queue manager with required connection authentication

Exercise 5. Working with workload balancing options

Estimated time

01:30

Overview

This exercise starts by setting up and testing a round robin message distribution of messages in a cluster. You then use queue, channel, and queue manager options to influence the distribution of messages.

Objectives

After completing this exercise, you should be able to:

- Send messages to cluster queues by using round robin distribution
- Use channel priority to influence the cluster workload
- Use channel weights to influence the cluster workload
- Use the queue manager to influence the cluster workload

Introduction

In this exercise, you use IBM MQ Explorer to create a basic cluster that is named CLUS1 with four queue managers:

- Full repository queue manager QMC1 on localhost(9001)
- Full repository queue manager QMC2 on localhost(9002)
- QMC3 on localhost(9003) that uses QMC1 for its full repository
- QMC4 on localhost(9004) that uses QMC2 for its full repository

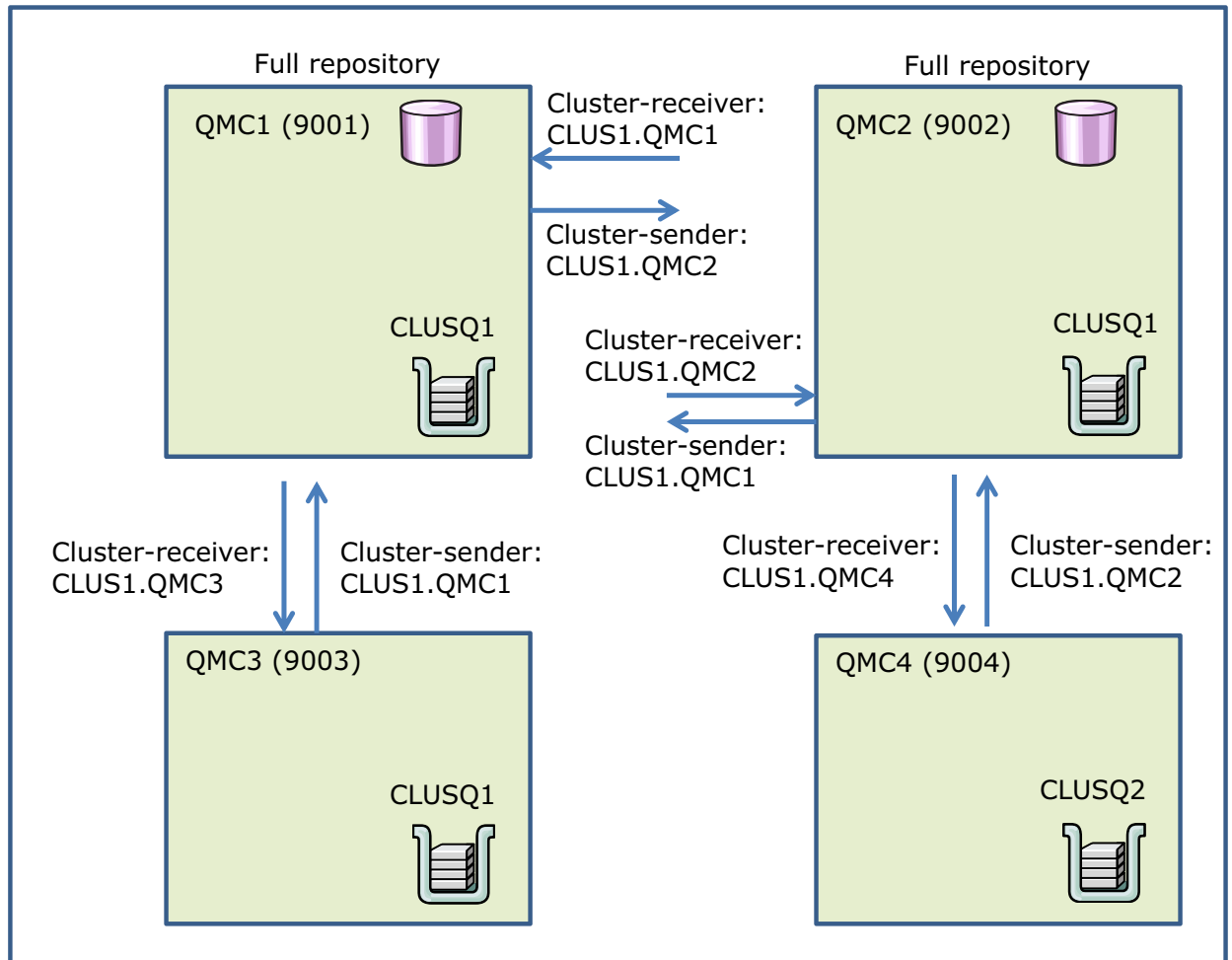
The IBM MQ Explorer **Create Cluster** wizard defines the following channels:

- One cluster-receiver (CLUSRCVR) channel to each queue manager in the cluster
- One cluster-sender (CLUSSDR) channel from each full repository queue manager to the other full repository queue manager (QMC1 to QMC2 and QMC2 to QMC1)
- One cluster-sender (CLUSSDR) channel from QMC3 to QMC1
- One cluster-sender (CLUSSDR) channel from QMC4 to QMC2

After you create the queue managers and the cluster, you define local queues that are used for cluster queues as follows:

- CLUSQ1 on QMC1, QMC2, QMC3
- CLUSQ2 on QMC4

Cluster = **CLUS1**



Requirements

- IBM MQ V9 and IBM MQ Explorer
- data.txt file in the C:\labfiles\Unit6 directory

5.1. End any active queue managers



Note

In earlier exercises, you used different queue managers. To prevent any possible resource issues, ensure that any queue managers that remain running are stopped before proceeding.

- ___ 1. Open a command prompt window by selecting the Command Prompt icon in the Windows taskbar.
- ___ 2. Display any active queue managers by typing `dspmqr` and pressing the Enter key. If you completed the first four exercises first, the expected response is as displayed in the text box.

`dspmqr`

<code>QMNAME (MQG1)</code>	<code>STATUS(Running)</code>
<code>QMNAME (MQ03)</code>	<code>STATUS(Running)</code>
<code>QMNAME (MQ0A)</code>	<code>STATUS(Running)</code>
<code>QMNAME (MQ0B)</code>	<code>STATUS(Running)</code>
<code>QMNAME (MQE5)</code>	<code>STATUS(Running)</code>
<code>QMNAME (MQ0X)</code>	<code>STATUS(Running)</code>

- ___ 3. For each queue manager that is running, stop it by typing the `endmqm` command. For the queue managers in the display, type the commands as shown in the text box:

```
endmqm -i MQG1
endmqm -i MQ03
endmqm -i MQ0A
endmqm -i MQ0B
endmqm -i MQE5
endmqm -i MQ0X
```

- ___ 4. Ensure that all queue managers are ended by typing `dspmqr` again, and press the Enter key. If the status is “quiescing”, they are in the process of stopping. If they say “running”, type `endmqm` again for any remaining queue managers that are running.

5.2. Defining the cluster queue managers, channels, and queues

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

- ___ 5. Start IBM MQ Explorer unless it is already running.
- ___ 6. Using IBM MQ Explorer, create the following queue managers as described in the table. Accept the default values for any properties that are not listed in the table.

Queue manager name	TCP/IP Listener port	Dead-letter queue
QMC1	9001	QMC1.DLQ
QMC2	9002	QMC2.DLQ
QMC3	9003	QMC3.DLQ
QMC4	9004	QMC4.DLQ

- ___ 7. Create the dead-letter queue for each queue manager.
- ___ 8. Create the queue manager cluster.
 - ___ a. In the IBM MQ Explorer Navigator, right-click **Queue Manager Clusters** and then click **New > Queue manager cluster**.
 - ___ b. For the cluster name, enter `CLUS1` and then click **Next**.
 - ___ c. For the first full repository queue manager, select **QMC1** and then click **Next**.



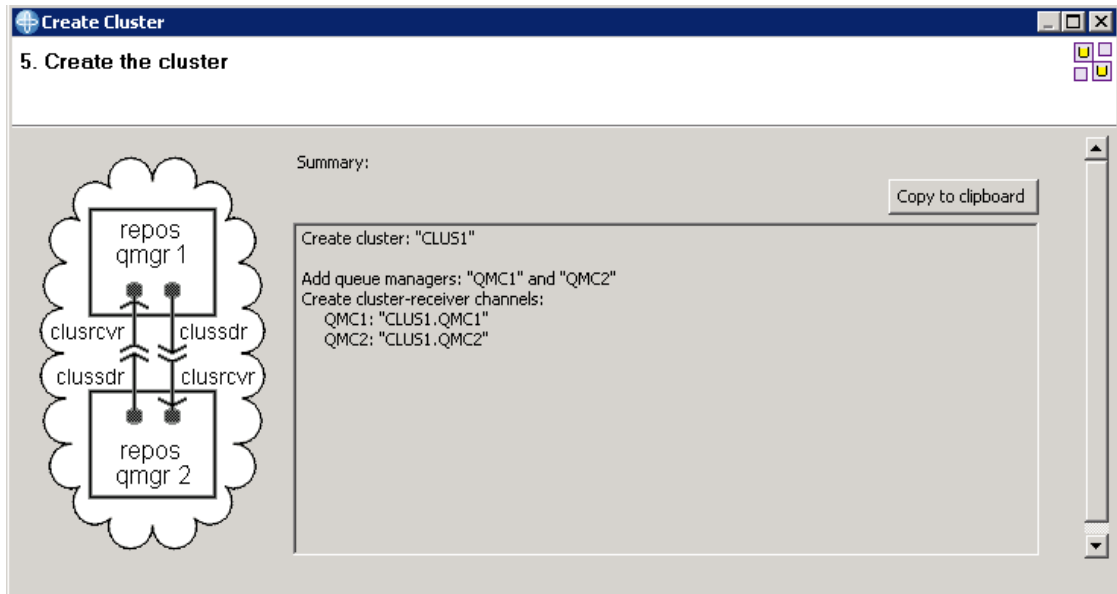
Information

The MQSC command to define QMC1 as a full repository queue manager is:

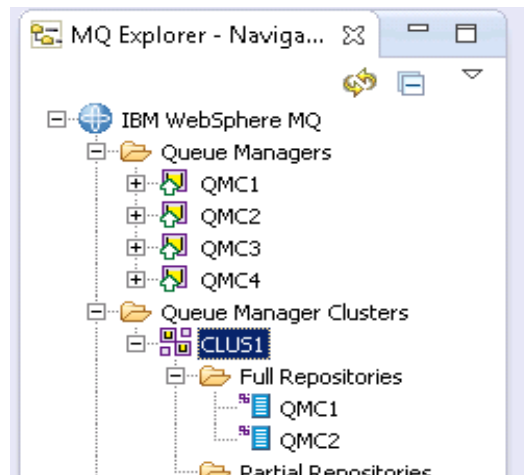
```
ALTER QMGR REPOS(CCLUS1)
```

- ___ d. For the second full repository queue manager, select **QMC2** from the list and then click **Next**.
- ___ e. Click **Next** to define the cluster channels.
- ___ f. Change the name of cluster-receiver channel for QMC1 to `CLUS1.QMC1` and the connection name to `localhost(9001)` and then click **Next**.
- ___ g. Change the name of cluster-receiver channel for QMC2 to `CLUS1.QMC2` and the connection name to `localhost(9002)` and then click **Next**.

- ___ h. Review the cluster summary and then click **Finish**.

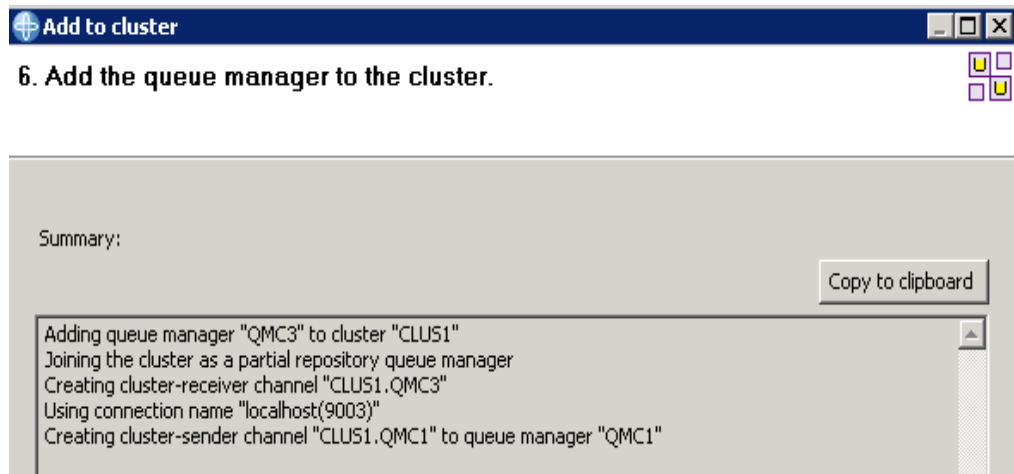


- ___ 9. Expand the **Queue Managers Clusters** folder in the IBM MQ Explorer and verify that you have a cluster that is named `CLUS1` with `QMC1` and `QMC2` listed as full repositories.



- ___ 10. Select the full repository queue manager `QMC1` under the `Full Repositories` folder.
- ___ 11. Verify that `QMC1` has one cluster-receiver channel that is named `CLUS1.QMC1` and one cluster-sender channel that is named `CLUS1.QMC2`.
- ___ 12. Add `QMC3` to the cluster as partial repository.
- ___ a. Right-click **CLUS1** in IBM MQ Explorer Navigator view and then click **Add Queue Manager to Cluster**.
 - ___ b. Select **QMC3** and then click **Next**.
 - ___ c. Select **Partial repository** and then click **Next**.
 - ___ d. Change the name of the cluster-receiver channel to `CLUS1.QMC3` and the connection name to `localhost(9003)`. Click **Next**.
 - ___ e. Select **QMC1** as the full repository queue manager and then click **Next**.

- ___ f. Accept the default to use the cluster-receiver channel **CLUS1.QMC1**.
- ___ g. Review the summary and then click **Finish**.



Information

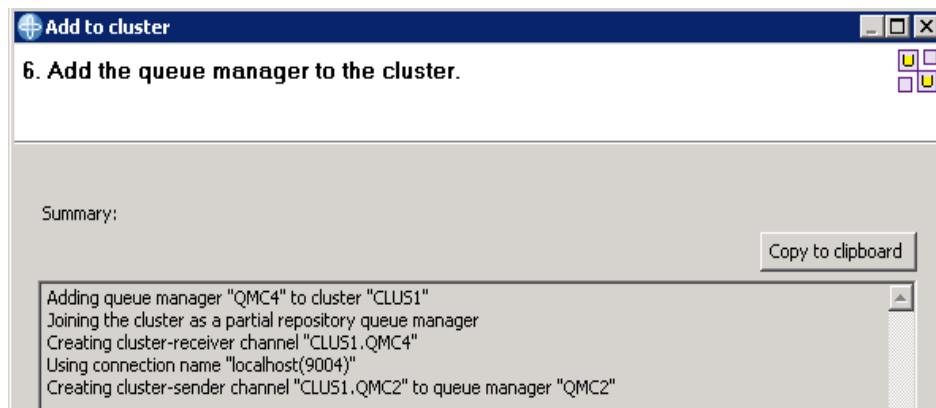
To use MQSC to add a partial repository to a cluster, define a cluster-receiver channel for QMC3 and define a cluster-sender channel that points to one of the full repository queue managers.

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

```
DEFINE CHANNEL(CCLUS1.QMC3) CHLTYPE(CCLUSRCVR) TRPTYPE(TCP) +
CONNAME('localhost(9003)') CLUSTER(CCLUS1)
```

```
DEFINE CHANNEL(CCLUS1.QMC1) CHLTYPE(CCLUSSDR) TRPTYPE(TCP) +
CONNAME('localhost(9001)') CLUSTER(CCLUS1)
```

- ___ 13. Using the same techniques as step 9, add QMC4 to the cluster as a partial repository that uses QMC2 as a full repository.
 - Change the name of the cluster-receiver channel to `CLUS1.QMC4` and the connection name to `localhost(9004)`.
 - Select **QMC2** as the full repository queue manager.
 - Accept the default to use the cluster-receiver channel **CLUS1.QMC2**.



- ___ 14. Select QMC1 under the `Full Repositories` folder and verify that cluster-sender channels were automatically defined to the partial repository queue managers QMC3 and QMC4.
- ___ 15. Select QMC3 under the **Partial Repositories** folder and verify that QMC3 has cluster-sender channels to both full repositories (QMC1 and QMC2).
- ___ 16. Notice that the channel to the second full repository (QMC2) was automatically defined.
- ___ 17. Create a cluster queue that is named `CLUSQ1` on QMC1.
 - ___ a. Expand QMC1 under the **Queue Managers** folder in the Navigator.
 - ___ b. Right-click **Queues** and then click **New > Local Queue**.
 - ___ c. Enter `CLUSQ1` for the queue name and then click **Next**.
 - ___ d. On the **Cluster** properties page, click **Shared in cluster** and enter `CLUS1` for the cluster name.
 - ___ e. Click **Finish**.

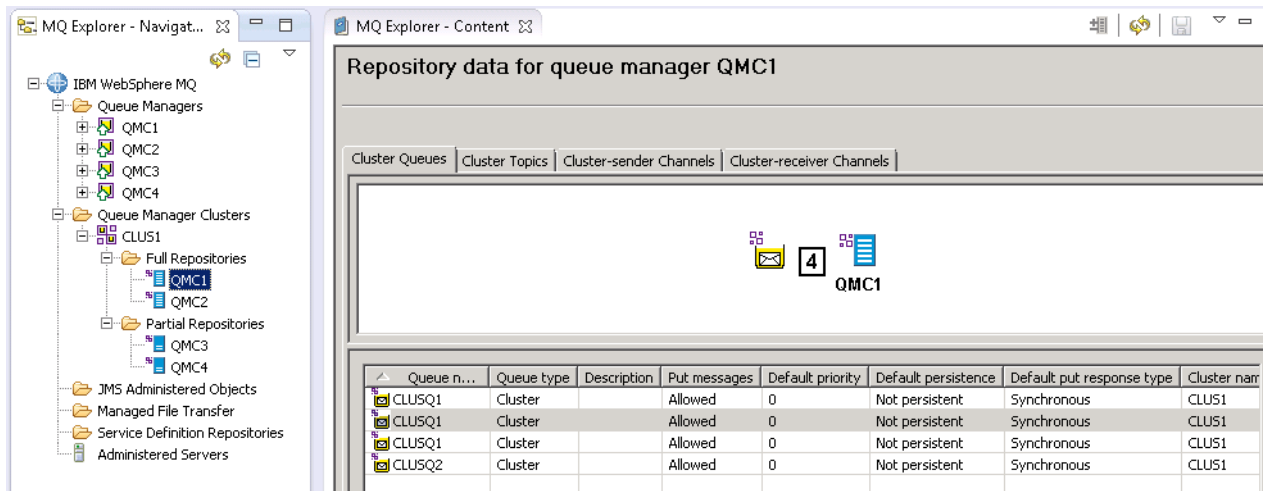


Information

The MQSC command for creating the cluster queue is: `DEF QL(CLUSQ1) CLUSTER(CLUS1)`

- ___ f. Click **Finish**.
- ___ 18. Follow the procedure in step 13 and create the following queues on QMC2, QMC3, QMC4:
 - ___ a. On QMC2, create a cluster queue that is named `CLUSQ1` that is shared in cluster `CLUS1`.
 - ___ b. On QMC3, create a cluster queue that is named `CLUSQ1` that is shared in cluster `CLUS1`.
 - ___ c. On QMC4, create a cluster queue that is named `CLUSQ2` that is shared in cluster `CLUS1`.
- ___ 19. Select **QMC1** under the `Full Repositories` folder and then click the **Cluster Queues** tab to verify that you have a cluster queue on each queue manager.

QMC1 should show a cluster queue that is named `CLUSQ1` on QMC1, QMC2, and QMC3 and a cluster queue that is named `CLUSQ2` on QMC4.



The screenshot displays the IBM MQ Explorer interface. The left pane, titled 'MQ Explorer - Navigat...', shows a tree view of the system. Under 'Queue Managers', QMC1, QMC2, QMC3, and QMC4 are listed. Under 'Queue Manager Clusters', CLUS1 is shown, which contains 'Full Repositories' (QMC1, QMC2, QMC3) and 'Partial Repositories' (QMC4). The right pane, titled 'MQ Explorer - Content', shows the 'Repository data for queue manager QMC1' tab. This tab includes a table of cluster queues.

Queue n...	Queue type	Description	Put messages	Default priority	Default persistence	Default put response type	Cluster nam
CLUSQ1	Cluster		Allowed	0	Not persistent	Synchronous	CLUS1
CLUSQ1	Cluster		Allowed	0	Not persistent	Synchronous	CLUS1
CLUSQ2	Cluster		Allowed	0	Not persistent	Synchronous	CLUS1

5.3. Using a round-robin scheme for workload balancing

In the most basic scenario, if a number of messages are put to the cluster queue CLUSQ1, messages are equally distributed between servers in the cluster in which a local definition of the queue exists.

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

- ___ 20. Use the sample program `amqspout` and the supplied text file `data.txt` in the `C:\labfiles\Unit6` directory to write 15 messages to the cluster queue CLUSQ1.

In a command window, type:

```
amqspout CLUSQ1 QMC1 < C:\labfiles\Unit6\data.txt
```

- ___ 21. Use IBM MQ Explorer to view the queue depth of CLUSQ1 on QMC1, QMC2, and QMC3.

What do you notice? You should see that all 15 messages were put to one queue manager, QMC1.

- ___ a. Examine the **Cluster** properties for the queue manager QMC1. The **Cluster workload use queue** (CLWLUSEQ) attribute is set to **Local**.
- ___ b. Examine the **Cluster** properties for the local queue CLUSQ1 on QMC1. The default queue definition has the **Default bind type** (DEFBIND) set to **Open** and the **Cluster workload use queue** (CLWLUSEQ) attribute set to **Queue manager**.

The queue and queue manager cluster attributes explain why all the messages were put onto the local cluster queue on QMC1.

- The queue attribute CLWLUSEQ(QMGR) means that the CLWLUSEQ attribute of the queue manager definition specifies the behavior. By default, the queue manager CLWLUSEQ attribute is set to LOCAL so the target of an MQPUT is the local queue instance, if one exists.
 - DEFBIND(OPEN) binds the queue handle to a specific instance of the cluster queue when the queue is opened so all messages are put to the same queue.
- ___ 22. Change the queue definition for CLUSQ1 queue on QMC1, QMC2, and QMC3 so that **Default bind type** is set to **Not fixed** and **Cluster workload use queue** is set to **Any**.
- ___ a. Right-click **CLUSQ1** on the **Queue** content view and then click **Properties**.
 - ___ b. Click **Cluster** to display the **Cluster** properties.
 - ___ c. Change **Default bind type** set to **Not fixed** so that the queue handle is not bound to any one instance of the cluster queue.
 - ___ d. Change **Cluster workload use queue** to **Any** so that any queue can be used.

- __ e. Click **Apply** and then click **OK**.



Information

The MQSC command for changing the cluster queue properties is:

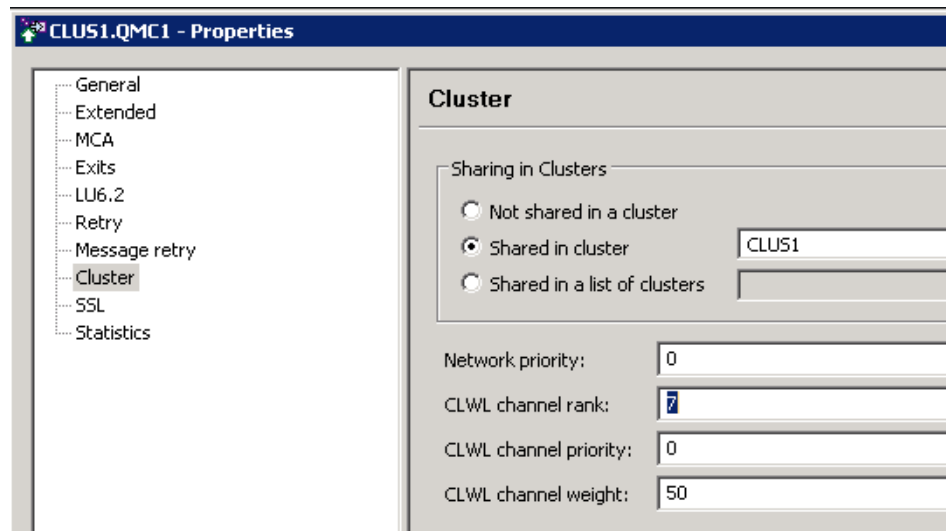
```
ALTER QL(CLUSQ1) DEFBIND(NOTFIXED) CLWLUSEQ(ANY)
```

- __ 23. Clear the messages on CLUSQ1 on QMC1.
- __ 24. Use the `amqspout` program to put the 15 messages to the cluster queue CLUSQ1 from QMC1.
- __ a. Run the `amqspout` program with the data file that is named `data.txt`.
- ```
amqspout CLUSQ1 QMC1 < C:\labfiles\Unit6\data.txt
```
- \_\_ b. View the queue depth of CLUSQ1 on the three queue managers.
- What do you notice now? The three instances of CLUSQ1 should have five messages each.
- \_\_ c. The messages in the `data.txt` file include a sequence number. Browse the messages on the queues and examine the message data to determine the distribution of the messages.

## 5.4. Using channel and queue rank to control workload

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

- \_\_\_ 25. Use IBM MQ Explorer to change the cluster-receiver channel definitions of QMC1 to have a cluster workload channel rank of 7.
  - \_\_\_ a. In the IBM MQ Explorer Navigator view, click the **Channels** folder under the queue manager to display the **Channels** content view.
  - \_\_\_ b. Right-click the cluster-receiver channel and then click **Properties**.
  - \_\_\_ c. On the **Cluster** page, set the **CLWL channel rank** property to 7.



- \_\_\_ d. Click **Apply** and then click **OK**.



### Information

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

```
ALTER CHANNEL(CCLUS1.QMC1) CHLTYPE(CCLUSRCVR) CLWLRRANK(7)
```

- \_\_\_ 26. Follow the same procedure as step 1 and change the cluster-receiver channel definition of QMC3 to have a cluster workload channel rank of 7.
- \_\_\_ 27. Clear the CLUSQ1 of all messages on each queue manager so that queue depth is zero.

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. The queue can be cleared by right-clicking the queue in the **Queue** contents view, clicking **Clear Messages**, and then selecting **Queue will be cleared using MQGET API calls**. This option gets the messages and closes the queue.

- \_\_\_ 28. Run the `amqspout` program on QMC1 with `data.txt` to put 15 messages to the cluster queue CLUSQ1.
- ```
amqspout CLUSQ1 QMC1 < C:\labfiles\Unit6\data.txt
```
- ___ 29. What do you notice about the distribution of messages?
- ___ 30. You should see that CLUSQ1 on QMC1 has eight messages, CLUSQ1 on QMC2 has zero messages, and CLUSQ1 on QMC3 has seven messages.
- ___ 31. Change the cluster workload channel rank for the QMC2 cluster receiver channel to a value of 9, and rerun the `amqspout` program.
- ___ 32. All the messages should now be directed to QMC2 because it has the highest rank. CLUSQ1 on QMC2 should have a queue depth of 15.
- ___ 33. Change the cluster workload queue rank (CLWLRANK) property for CLUSQ1 on QMC3 to 6.
- ___ a. Right-click **CLUSQ1** on the **Queue** content view and then 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)
```

- ___ 34. Use `amqspout` again to put messages to the cluster queue CLUSQ1 and notice where the messages were put this time.

```
amqspout CLUSQ1 QMC1 < C:\labfiles\Unit6\data.txt
```

All the messages were put on CLUSQ1 on QMC2 because the channel rank is evaluated before the queue rank. QMC2 has the highest channel rank.

5.5. Using channel priority to control workload

- ___ 35. Use IBM MQ Explorer to reset the cluster workload channel rank (CLWLRANK) for the cluster-receiver channels on QMC1, QMC2, and QMC3 to a value of zero.
- ___ 36. Clear the cluster queue CLUSQ1 on QMC1, QMC2, and QMC3.
- ___ 37. Using IBM MQ Explorer, set the cluster-receiver channel **CLWL channel priority** (CLWPRTY) of QMC2 to 3 and QMC3 to 1.
- ___ 38. Put the messages again by using the `amqspout` program.
- ___ 39. Where did all the messages end up? Why?
- ___ 40. All 15 messages are on CLUSQ1 on QMC3. Even though QMC2 has a higher CLWL channel priority, the CLWL queue rank on CLUSQ1 on QMC3 is still set to 6 from part 3, step 6. The cluster workload algorithm evaluates the channel and queue rank before the CLWL channel priority.
- ___ 41. Modify the cluster so that all the messages are directed to the highest priority queue manager, QMC2.
 - ___ a. Alter the CLUSQ1 queue definition on QMC3 so that the **CLWL queue rank** (CLWLRANK) is zero.
 - ___ b. Run the `amqspout` program and verify that all 15 messages went to CLUSQ1 on QMC2.
- ___ 42. Stop the cluster-receiver (CLUSRCVR) channel CLUS1.QMC2.
- ___ 43. After giving the cluster repository a few seconds to update, run the `amqspout` test again. You should see that all the messages go to QMC3, which is the highest priority queue manager.
- ___ 44. Restart the cluster-receiver channel (CLUSRCVR) on channel on QMC2.
- ___ 45. Clear all messages from the CLUSQ1 on QMC1, QMC2, and QMC3.

5.6. Using channel WEIGHT to control workload

Suppose that QMC3 has greater processing power than any of the other queue manager servers in this cluster. You can use channel weight to direct workload to the most powerful queue manager.

- ___ 46. Reset the cluster workload cluster priority (CLWLPRTY) of the QMC2 and QMC3 cluster-receiver channels to zero.
- ___ 47. Assuming that QMC3 has twice the processing power of QMC1 and QMC2, set the following channel weights (CLWLWGHT) on the cluster-receiver channels:
 - QMC1: CLWL channel weight = 25
 - QMC2: CLWL channel weight = 25
 - QMC3: CLWL channel weight = 50
- ___ 48. Run four iterations of the `amqsput` test and observe the queue depths of CLUSQ1 on QMC1, QMC2, and QMC3.
- ___ 49. You should see that 50% of the messages went to QMC3, 25% of the messages went to QMC1, and the other 25% of the messages went to QMC2.
- ___ 50. For example, if you put 60 messages, you would see 15 messages on QMC1, 15 messages on QMC2, and 30 messages on QMC3.
- ___ 51. Clear all messages from the CLUSQ1 from QMC1, QMC2, and QMC3.
- ___ 52. Reset the cluster workload channel weight (CLWLWGHT) attribute on the cluster-receiver channel on QMC1 and QMC2 to the default value of 50.

5.7. Restricting the number of outbound cluster channels

The cluster workload most recently used attribute (CLWLMRUC) is specified on a queue manager definition. It controls the number of channels that can be used as the destination of a message that is put into a large network where potentially hundreds or thousands of destination choices are possible.

In this step, QMC1 is used as a gateway queue manager. It no longer contains local instances of cluster queues.

- ___ 53. Delete the local queue CLUSQ1 from QMC1.
- ___ 54. Define the local queue CLUSQ1 on QMC4 so that it is shared in CLUS1, the **Default bind type** is set to **Not fixed**, and **Cluster workload use queue** is set to **Any**.



Information

The MQSC command to create the cluster queue on QMC4 is:

```
DEF QL(CLUSQ1) CLUSTER(CLUS1) DEFBIND(NOTFIXED) CLWLUSEQ(ANY)
```

- ___ 55. Limit the number of queue manager destinations from the gateway queue manager (QMC1) to two.

In queue manager **Cluster** properties for QMC1, set the **Max outbound cluster channels** property (CLWLMRUC) to 2.

QMC1 - Properties

Cluster

Cluster Membership
This queue manager is a member of these clusters:

Cluster name:	Full repository for a cluster:
CLUS1	Yes

Cluster workload exit:

Cluster workload data:

Cluster workload length:

Max outbound cluster channels:

Cluster workload use queue:

Default cluster transmission queue:

Cluster workload mode:



Information

The MQSC command to restrict the maximum number of outbound cluster channels to 2 is:

```
ALTER QMGR CLWLMRUC(2)
```

- __ 56. Run the `amqspout` test.

What do you notice when you look at the CLUSQ1 queue depth across the cluster? Remember that CLUSQ1 is now defined on QMC4.

You should see that the workload algorithm limited the messages to two queue managers even though three queue managers contain the cluster queue.

- __ 57. Now run the `amqspout` program to put messages to cluster queue CLUSQ2 on QMC4.

```
amqspout CLUSQ2 QMC4 < C:\labfiles\Unit6\data.txt
```

- __ 58. Check the queue depth on QMC4 for CLUSQ2. It should show 15 messages.

- __ 59. Run the `amqspout` to put messages to CLUSQ1 on QMC1:

```
amqspout CLUSQ1 QMC1 < C:\labfiles\Unit6\data.txt
```

What was the distribution of messages this time?

The messages are now spread across QMC2 and QMC3 or QMC3 and QMC4. One channel was eliminated as a destination this time, even though it was being used for a different queue.

- __ 60. Clear the cluster queues on QM2, QMC3, and QM4.

- __ 61. Reset the number of queue manager destinations from the gateway queue manager (QMC1) to the default value of 999999999.

In queue manager **Cluster** properties for QMC1, set the **Max outbound cluster channels** property (CLWLMRUC) to 999999999.

- __ 62. Stop all queue managers that are running. To determine the active queue managers, type the command `dspmq` in a command prompt window and press the Enter key.

- __ 63. Use `endmqm` followed by the queue manager name and press the Enter key to stop any queue managers that displayed with a Running status.

End of exercise

Exercise review and wrap-up

In the first part of the exercise, you created a cluster with two full repository queue managers and two partial repository queue managers with cluster queues.

You then used the queue manager, queue, and channel cluster workload properties to control the workload.

End of exercise

Exercise 6. Configuring a publish/subscribe cluster

Estimated time

01:00

Overview

In this exercise, you work with publish/subscribe clusters and learn how to verify the path of the messages.

Objectives

After completing this exercise, you should be able to:

- Create a cluster topic and review its status
- Create subscriptions and publications to a topic
- Use the dspmqrt tool to identify the path of a message in the publish/subscribe cluster
- Configure and verify topic host routing

Exercise instructions

Preface

In part 1 of the exercise, you set up a publish/subscribe cluster and use MQSC commands and IBM MQ Explorer to verify the cluster properties. To attach a sample publish/subscribe application and start to send messages through the publish/subscribe cluster, you use the `amqssub` and `amqspu` commands and IBM MQ Explorer.

In part 2 of the exercise, you use the IBM MQ display route command (`dspmqrt`) to verify the route that the message takes through the cluster. You also use IBM MQ Explorer to view the proxy subscriptions.

In part 3 of the exercise, you modify the publish/subscribe cluster to use topic host routing. You use the IBM MQ display route command (`dspmqrt`) to verify the route that the message is routed through the topic host.

Requirements

- IBM MQ V9 and IBM MQ V9 Explorer
- The IBM MQ `amqssub` and `amqspu` sample programs

6.1. Stop any active queue managers

- ___ 1. Open a command prompt window.
- ___ 2. List the queue managers by typing `dspmq` and pressing the Enter key.
- ___ 3. For any queue manager that shows Running status in the output, type `endmqm` followed by the queue manager name, and press the Enter key to stop the active queue managers.

6.2. Clustered publish/subscribe with direct routing

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 IBM MQ Explorer to create a cluster that is named PSCLUS with five queue managers (QMPS1 to QMPS5).

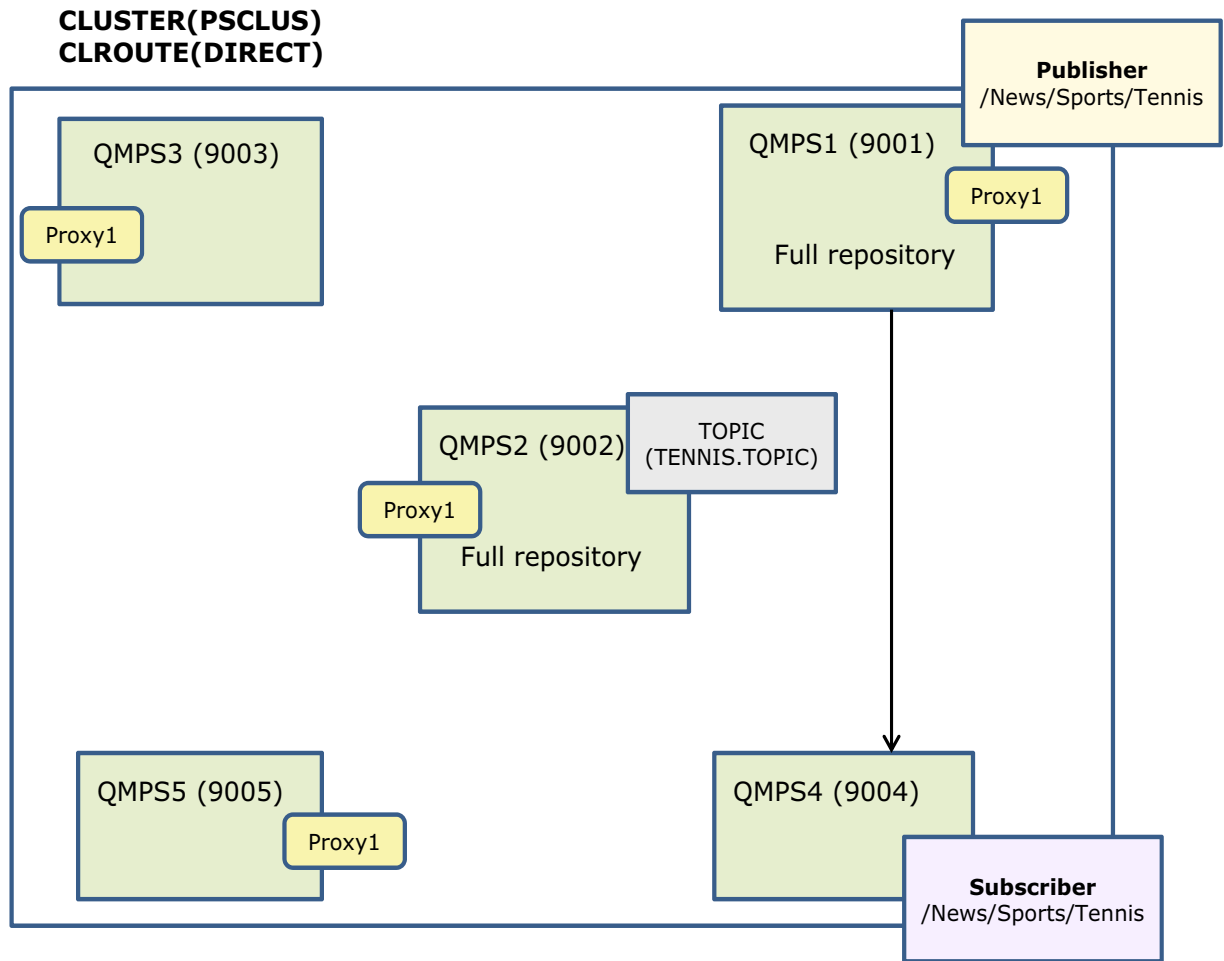
- Full repository queue manager QMPS1 on localhost(9001) uses QMPS2 for its full repository.
- Full repository queue manager QMPS2 on localhost(9002) uses QMPS1 for its full repository.
- QMPS3 on localhost(9003) uses QMPS1 for its full repository.
- QMPS4 on localhost(9004) uses QMPS1 for its full repository.
- QMPS5 on localhost(9005) uses QMPS1 for its full repository.

The IBM MQ Explorer **Create Cluster** wizard defines the following channels:

- One cluster-receiver (CLUSRCVR) channel to each queue manager in the cluster
- One cluster-sender (CLUSSDR) channel from each full repository queue manager to the other full repository queue manager (QMPS1 to QMPS2 and QMPS2 to QMPS1)
- One cluster-sender (CLUSSDR) channel for each partial repository (QMPS3, QMPS4, QMPS5) queue to a full repository queue manager

An administered topic object that is named TENNIS.TOPIC, which has CLROUTE(DIRECT), is declared on the QMPS2 queue manager. When the subscriber application runs and attaches to the QMPS4 queue manager through the topic string `/News/Sports/Tennis`, proxy subscriptions are propagated to every member of the cluster.

After you create the cluster and topic, a publication that is published on QMPS1 is routed directly to the subscriber on QMPS4 by using the cluster channels between QMPS1 and QMPS4. The arrows in the figure show the path that the message follows when the publish/subscribe cluster uses direct routing.

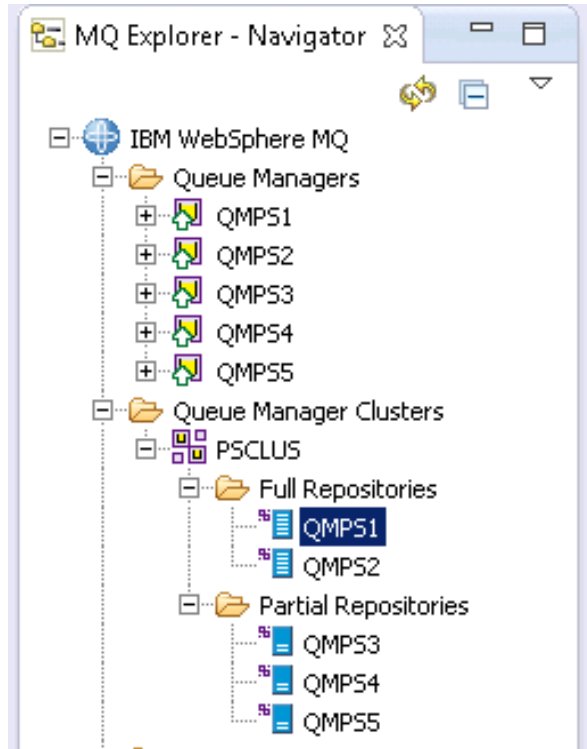


To prevent port conflicts, ensure that you stop any active queue managers as instructed in the first part of this exercise before you proceed.

Section 1: Create and verify the queue managers and the cluster that is used in this exercise

- ___ 4. Using IBM MQ Explorer, create the cluster queue managers.
 - QMPS1 on TCP listener port 9001
 - QMPS2 on TCP listener port 9002
 - QMPS3 on TCP listener port 9003
 - QMPS4 on TCP listener port 9004
 - QMPS5 on TCP listener port 9005
- ___ 5. Using IBM MQ Explorer, create a dead-letter queue for each queue manager.

- ___ 6. Using IBM MQ Explorer, create the cluster PSCLUS. Use the syntax of *ClusterName.QueueManager* for the cluster channels. For example, name the cluster-receiver channel for QMPS1: PSCLUS.QMPS1
- QMPS1 is a full repository queue manager on localhost(9001).
 - QMPS2 is a full repository queue manager on localhost(9002).
 - QMPS3 is a partial repository on localhost(9003) that uses QMPS1 for its full repository.
 - QMPS4 is a partial repository on localhost(9004) that uses QMPS1 for its full repository.
 - QMPS5 is a partial repository on localhost(9005) that uses QMPS2 for its full repository.



- ___ 7. Start a `runmqsc` session for queue manager QMPS1 by typing `runmqsc QMPS1` and pressing the Enter key.
- ___ 8. Verify the correct creation of the cluster by typing the command as shown in the text box, and press the Enter key.

```
dis clusqmgr(*) deftype status qmtype
```

Expected results:

```
1 : dis clusqmgr(*) deftype status qmtype
AMQ8441: Display Cluster Queue Manager details.
    CLUSQMGR(QMPS1)                CHANNEL(PSCLUS.QMPS1)
    CLUSTER(PSCLUS)                DEFTYPE(CLUSRCVR)
    QMTYPE(REPOS)                  STATUS(RUNNING)
AMQ8441: Display Cluster Queue Manager details.
    CLUSQMGR(QMPS2)                CHANNEL(PSCLUS.QMPS2)
    CLUSTER(PSCLUS)                DEFTYPE(CLUSSDR)
    QMTYPE(NORMAL)                 STATUS(RUNNING)
AMQ8441: Display Cluster Queue Manager details.
    CLUSQMGR(QMPS3)                CHANNEL(PSCLUS.QMPS3)
    CLUSTER(PSCLUS)                DEFTYPE(CLUSSDRA)
    QMTYPE(NORMAL)                 STATUS(RUNNING)
AMQ8441: Display Cluster Queue Manager details.
    CLUSQMGR(QMPS4)                CHANNEL(PSCLUS.QMPS4)
    CLUSTER(PSCLUS)                DEFTYPE(CLUSSDRA)
    QMTYPE(NORMAL)                 STATUS(RUNNING)
AMQ8441: Display Cluster Queue Manager details.
    CLUSQMGR(QMPS5)                CHANNEL(PSCLUS.QMPS5)
    CLUSTER(PSCLUS)                DEFTYPE(CLUSSDRA)
    QMTYPE(NORMAL)                 STATUS(RUNNING)
```

- ___ 9. Review the results:
 - ___ a. The values on the channel definition type, DEFTYPE, must be either CLUSRCVR, CLUSSDRA, or CLUSSDRB. If you see any CLUSSDR, it has an error.
 - ___ b. If you complete this step right after you create the cluster, the status of all channels is expected to be running. If you took a break before you typed the `DIS CLUSQMGR` command, status might be INACTIVE, which is good provided the previous step did not show any CLUSSDR type channels.
 - ___ c. You might also check that the two expected queue repository queue managers, QMPS1 and QMPS2, have QMTYPE(REPOS). The other member queue managers show QMTYPE(NORMAL).
- ___ 10. End the `runmqsc` session for queue manager QMPS1 by typing `end` and pressing the Enter key.
- ___ 11. Start a `runmqsc` session for queue manager QMPS2 by typing `runmqsc QMPS2` and pressing the Enter key.

Section 2: Create and review a cluster topic

- ___ 12. Create a cluster topic on QMPS2 that is called TENNIS.TOPIC with a topic string of `/News/Sports/Tennis`. In MQSC for QMPS2, by typing the command as shown in the text box, and press the Enter key. Keep the `runmqsc` session open.

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

Expected results:

```
1 : DEFINE TOPIC(TENNIS.TOPIC) TOPICSTR('/News/Sports/Tennis')
CLUSTER(PSCLUS)
AMQ8690: IBM MQ topic created.
```

- ___ 13. From the QMPS2 runmqsc session, verify the cluster topic definition by typing the display command as shown in the text box, and press the Enter key. Keep the runmqsc session open.
-

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

Expected response:

```
2 : DISPLAY TCLUSTER(TENNIS.TOPIC) TOPICSTR CLUSTER CLROUTE CLSTATE
AMQ8633: Display topic details.
TOPIC(TENNIS.TOPIC)                                TYPE(CLUSTER)
TOPICSTR(/News/Sports/Tennis)                       CLUSTER(PSCLUS)
CLROUTE(DIRECT)                                       CLSTATE(ACTIVE)
```

- ___ 14. From the QMPS2 runmqsc session, review and verify the topic tree by typing the display command as shown in the text box, and press the Enter key:
-

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

Expected response:

```
3 : DISPLAY TPSTATUS('/#') TYPE(TOPIC) CLUSTER CLROUTE SUBCOUNT ADMIN
AMQ8754: Display topic status details.
TOPICSTR()                                           ADMIN(SYSTEM.BASE.TOPIC)
CLUSTER( )                                           CLROUTE(NONE)
SUBCOUNT(0)
AMQ8754: Display topic status details.
TOPICSTR(/News)                                     ADMIN( )
CLUSTER( )                                           CLROUTE(NONE)
SUBCOUNT(0)
AMQ8754: Display topic status details.
TOPICSTR(/News/Sports/Tennis)                     ADMIN(TENNIS.TOPIC)
CLUSTER(PSCLUS)                                     CLROUTE(DIRECT)
SUBCOUNT(0)
AMQ8754: Display topic status details.
TOPICSTR(/News/Sports)                             ADMIN( )
CLUSTER( )                                           CLROUTE(NONE)
SUBCOUNT(0)
```

- ___ 15. Review the results of the display. In the `DIS TPSTATUS` output:
- ___ a. `TOPICSTR` is the topic string for each tree node.
 - ___ b. `ADMIN` contains the name of the administrative topic object.
 - ___ c. `CLUSTER` is the name of the cluster and appears on the clustered topic only.
 - ___ d. `CLROUTE` is set to `NONE` except for the `TENNIS.TOPIC` where it is set to `DIRECT`, which indicates direct routing.
 - ___ e. `SUBCOUNT` is the number of subscriptions currently aware of the topic.



I

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

You now end the repeat the steps to verify the cluster topic definition and status for queue managers QMPS1, QMPS3, QMPS4, and QMPS5.

- ___ 16. End the `runmqsc` session for queue manager QMPS2 by typing `end` and pressing the Enter key.



N

Instructions to use IBM MQ Explorer to review and verify the cluster topic are provided at the end of the MQSC steps.



E

For each queue manager – QMPS1, QMPS3, QMPS4, and QMPS5 – you type the command to display the cluster topic and status as shown in this text box. In the steps that follow, type the two commands in this example:

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

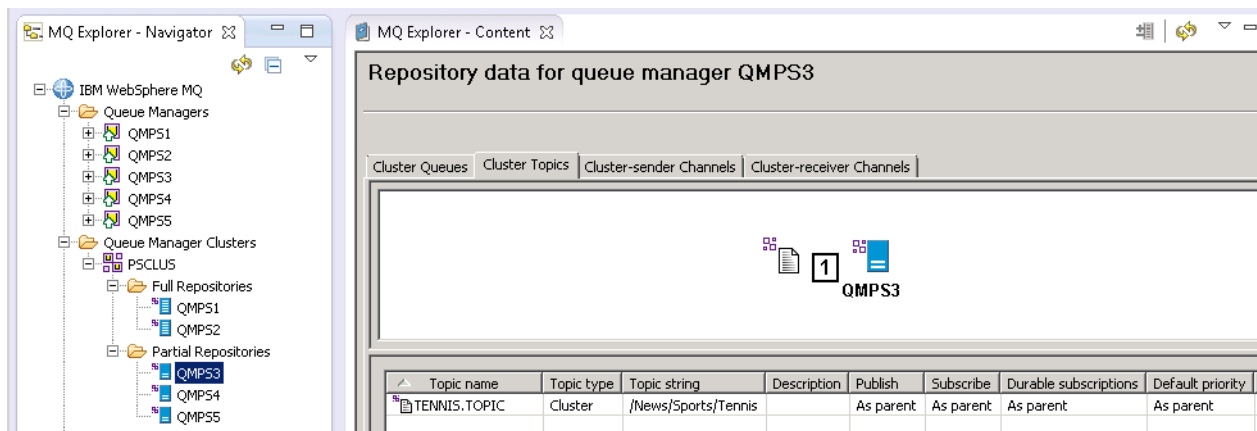
```
DISPLAY TPSTATUS('/#') TYPE(TOPIC) CLUSTER CLROUTE SUBCOUNT ADMIN
```

- ___ 17. Start a `runmqsc` session for queue manager QMPS1 by typing `runmqsc QMPS1` and pressing the Enter key.
- ___ 18. Type the commands as shown in the example box, and press the Enter key. Results are expected to resemble the results that are shown for queue manager QMPS2.
- ___ 19. End the `runmqsc` session for queue manager QMPS1 by typing `end` and pressing the Enter key.
- ___ 20. Start a `runmqsc` session for queue manager QMPS3 by typing `runmqsc QMPS3` and pressing the Enter key.

- ___ 21. Type the commands as shown in the example box, and press the Enter key. Results are expected to resemble the results that are shown for queue manager QMPS2.
- ___ 22. End the `runmqsc` session for queue manager QMPS3 by typing `end` and pressing the Enter key.
- ___ 23. Start a `runmqsc` session for queue manager QMPS4 by typing `runmqsc QMPS4` and pressing the Enter key.
- ___ 24. Type the commands as shown in the example box, and press the Enter key. Results are expected to resemble the results that are shown for queue manager QMPS2.
- ___ 25. End the `runmqsc` session for queue manager QMPS4 by typing `end` and pressing the Enter key.
- ___ 26. Start a `runmqsc` session for queue manager QMPS5 by typing `runmqsc QMPS5` and pressing the Enter key.
- ___ 27. Type the commands as shown in the example box, and press the Enter key. Results are expected to resemble the results that are shown for queue manager QMPS2.
- ___ 28. End the `runmqsc` session for queue manager QMPS5 by typing `end` and pressing the Enter key.

Section 3: Use IBM MQ Explorer as an extra way to verify the cluster topic

- ___ 29. You can also view the cluster topic in IBM MQ Explorer by clicking the queue manager under the cluster in the IBM MQ Explorer Navigator view.
- ___ 30. Click the **Cluster Topics** tab in the Content view.
- ___ 31. Scroll the Content view to the right to see the **Cluster queue manager** and **Cluster route** properties.



Section 4: Test the publish/subscribe cluster



I

You must work expeditiously through this step. This subscription has a timeout of 30 seconds before it automatically disconnects.

For the instructions that follow, if the automatic timeout is causing you problems, type the commands in each command window but do not press Enter until all the commands are ready. Then, go back to each command window, starting with the subscriptions, and press Enter.

- ___ 32. Open three command windows and arrange them so that you can see all three windows at the same time.
- ___ 33. In the first command window, set up a publisher on QMPS1. Type:

```
amqspub /News/Sports/Tennis QMPS1
```
- ___ 34. In the second command window, set up a subscriber on QMPS3. Type:

```
amqssub /News/Sports/Tennis QMPS3
```
- ___ 35. In the third command window, set up a subscriber on QMPS4. Type:

```
amqssub /News/Sports/Tennis QMPS4
```
- ___ 36. In the publisher command window (QMPS1), enter a message. You should see the message in both of the subscriber command windows.

The image shows three overlapping Windows Command Prompt windows, each titled "Administrator: Command Prompt".

The top window (QMPS1) shows the publisher setup and execution:

```
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.

C:\Users\Administrator.WS2008R2X64>amqspub /News/Sports/Tennis QMPS1
Sample AMQSPUBA start
target topic is /News/Sports/Tennis
French open today
Sample AMQSPUBA end
C:\Users\Administrator.WS2008R2X64>
```

The middle window (QMPS3) shows the subscriber setup and execution:

```
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.

C:\Users\Administrator.WS2008R2X64>amqssub /News/Sports/Tennis QMPS3
Sample AMQSSUBA start
Calling MQGET : 30 seconds wait time
message <French open today>
Calling MQGET : 30 seconds wait time
no more messages
Sample AMQSSUBA end
```

The bottom window (QMPS4) shows the subscriber setup and execution:

```
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.

C:\Users\Administrator.WS2008R2X64>amqssub /News/Sports/Tennis QMPS4
Sample AMQSSUBA start
Calling MQGET : 30 seconds wait time
message <French open today>
Calling MQGET : 30 seconds wait time
no more messages
Sample AMQSSUBA end
C:\Users\Administrator.WS2008R2X64>
```

- ___ 37. Close the subscriber command windows but leave the publisher (amqspub) command window open.
- ___ 38. In the publisher window, press the Enter key one time to end the program, but leave the window open.

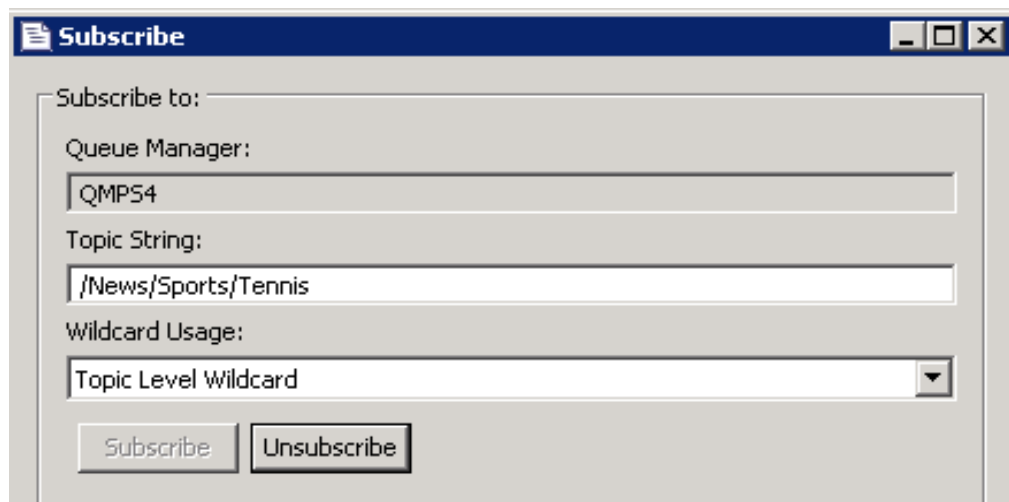
6.3. Testing cluster publication routing

In the first part of this exercise, the test showed that messages are delivered from publisher to subscriber, but no evidence exists about how the messages are routed from the publisher to the subscriber.

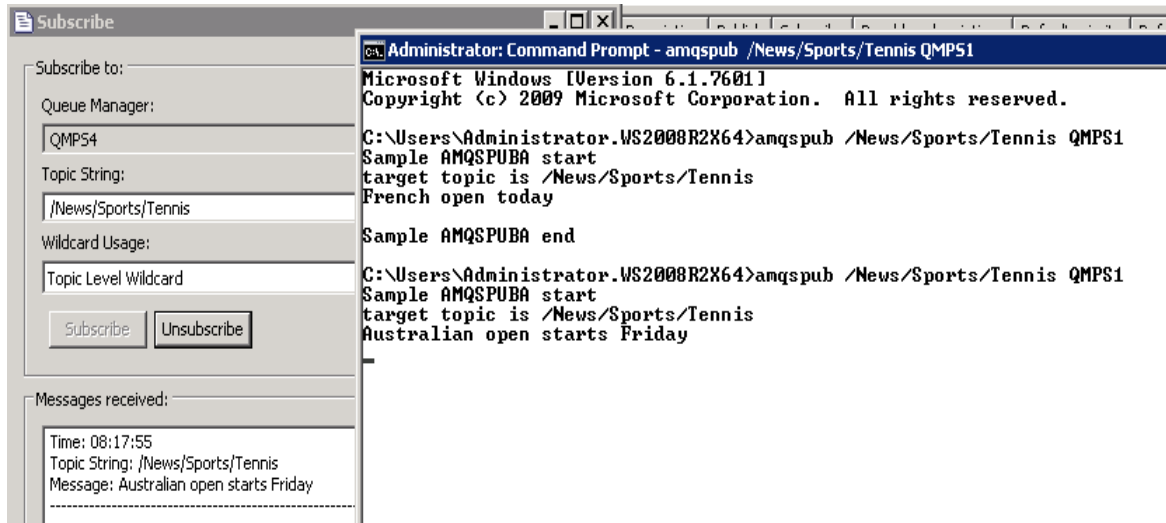
In this part of the exercise, you use IBM MQ Explorer to attach subscribers and the IBM MQ display route command (`dspmqrte`) to display the route of the data through the cluster.

Section 1: Create a subscription and publication

- ___ 39. In IBM MQ Explorer, create a test subscription on QMPS4.
 - ___ a. Expand the queue manager QMPS4 under the **Queue Managers** folder in the IBM MQ Explorer Navigator view.
 - ___ 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.



- ___ 40. In the command window, run the `amqspub` sample program and enter a message. You should see the message in the **Subscriber** window in IBM MQ Explorer.



Section 2: View the message route

- ___ 41. In a new command window, enter the display route command by typing the command as shown in the text box, and press the Enter key.

```
dspmqrte -ts /News/Sports/Tennis -ac -d yes -v outline activity -w 3 -m QMPS1
```

Expected results:

AMQ8694: DSPMQRTE command successfully put a message to topic string
'/News/Sports/Tennis', queue manager 'QMPS1'.

AMQ8657: DSPMQRTE command used CorrelId

0x414D5120514D4331202020202020205BB5F45320007D04.

AMQ8674: DSPMQRTE command is now waiting for information to display.

After a brief pause, the results of the command are expected to resemble the results that are shown.

Activity:

ApplName: 'QMPS1 '

ApplType: QmgrPublish

ActivityDesc: 'Message publication '

Operation:

OperationType: Put

QMgrName: 'QMPS1'

TopicString: '/News/Sports/Tennis'

Operation:

OperationType: Publish

SubId: X'414D5120514D4331202020202020205BB5F45320000D0F'

SubLevel: 1

QMgrName: 'QMPS1'

Activity:

ApplName: 'Sphere MQ\bin64\amqrmppa.exe'

ApplType: WindowsNT

Activity Desc: 'Sending Message Channel Agent'

Operation:

OperationType: Get

QMgrName: 'QMPS1'

QName: 'SYSTEM.CLUSTER.TRANSMIT.QUEUE'

ResolvedQName: 'SYSTEM.CLUSTER.TRANSMIT.QUEUE'

Operation:

OperationType: Send

QMgrName: 'QMPS1'

RemoteQMgrName: 'QMPS4'

ChannelName: 'PSCLUS.QMPS4'

ChannelType: ClusSdr

XmitQName: 'SYSTEM.CLUSTER.TRANSMIT.QUEUE'

```
Activity:
  ApplName: 'Sphere MQ\bin64\amqrmppa.exe'
  ApplType: WindowsNT
  Activity Desc: 'Sending Message Channel Agent'
```

```
Operation:
  OperationType: Receive
  QMgrName: 'QMPS4'
  RemoteQMGrName: 'QMPS1'
  ChannelName: 'PSCLUS.QMPS4'
  ChannelType: ClusRcvr
```

```
Operation:
  OperationType: Put
  QMgrName: 'QMPS4'
  QName: 'SYSTEM.INTER.QMGR.PUBS'
  ResolvedQName: 'SYSTEM.INTER.QMGR.PUBS'
```

```
-----
Activity:
  ApplName: 'QMPS4'
  ApplType: QmgrPublish
  Activity Desc: 'Message publication'
```

```
Operation:
  OperationType: Put
  QMgrName: 'QMPS4'
  TopicString: '/News/Sports/Tennis'
```

```
Operation:
  OperationType: Publish
  SubId: X'414D5120514D433420202020202020202084B5F45320006404'
  SubLevel: 1
  QMgrName: 'QMPS4'
```

```
-----
AMQ8652: DSPMQORTE command has finished.
```



The report shows that the message was sent directly from the publisher (QMPS1) to the subscriber (QMPS4) by using the cluster channels.

__ 42. Close the **Subscribe** window.

Section 3: Validate proxy subscriptions

- ___ 43. Create a test subscription on QMP55 by using IBM MQ Explorer:
- ___ a. Expand the queue manager QMP55 under the **Queue Managers** folder in the IBM MQ Explorer Navigator view.
 - ___ b. Right-click **Topics** and then click **Test Subscription**.
 - ___ c. In the **Topic String** field, type: /News/Sports/Tennis
 - ___ d. Click **Subscribe**. **Do not close**.
- ___ 44. Validate the proxy subscriptions by using IBM MQ Explorer:
- ___ a. Expand the queue manager QMP55 under the **Queue Managers** folder in the IBM MQ Explorer Navigator view.
 - ___ b. Right-click **Topics** and then click **Status**.
 - ___ c. Expand the topic tree to view the details of the **Tennis** node, starting from the **[Empty]** node. The proxy subscription count is provided in the **Sub count** column.

QMP55 - Topic Status

Queue Manager: QMP55

Topic status:

Topic string	Admin topic name	Sub count	Pub count	Retained publication	NPM delivery	PM delivery	Publication scope	Subscription scope	Clust
[Empty]	SYSTEM.BASE.TOPIC	0	0	No	To all available subscribers	To all durable subscribers	All	All	
News		0	0	No	To all available subscribers	To all durable subscribers	All	All	
Sports		0	0	No	To all available subscribers	To all durable subscribers	All	All	
Tennis	TENNIS.TOPIC	1	0	No	To all available subscribers	To all durable subscribers	All	All	PSCL

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

/News/Sports/Tennis - Status

Queue Manager: QMP55

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

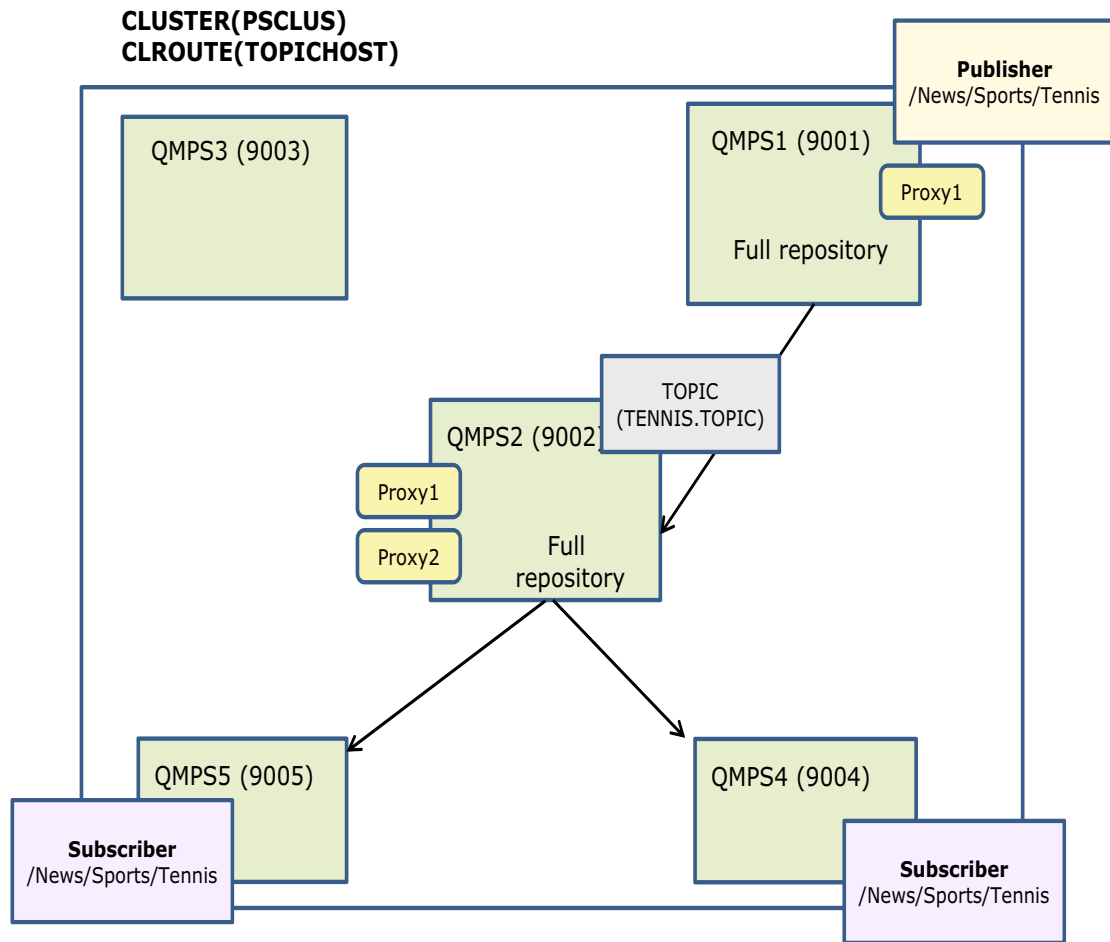
Filter: [Not Available]

Topic string	Subscription ID	User	Durable	Type	Connection ID
/News/Sports/Tennis	414D5120514D5053352...	Adminis...	No	API	414D5143514D5...

- ___ e. Close the topic status windows.
- ___ f. Close the **Subscribe** window.

6.4. Clustered publish/subscribe with topic host routing

In this part of the exercise, you modify the topic on QMPS2 to change it from direct routing to topic host routing. In this part of the exercise, QMPS2 is the topic host, as shown in the figure. When you modify the topic, create a subscription, and trace the route, you see that the message is now routed through the topic host queue manager.



Topic host routing is available with IBM MQ V8 and later. In a cluster, the full repositories and partial repositories that participate in publish/subscribe messaging communication must run IBM MQ V8 or later.

In IBM MQ V8 and later, the attribute **VERSION** was added to the `DISPLAY CLUSQMGR` command so that you can see when IBM MQ V8 queue managers coexist with queue managers from previous versions. The IBM MQ version is also displayed when you view the cluster queue manager details in IBM MQ Explorer.

Section 1: Replace the existing *DIRECT* routed topic on QMPS2 with a *TOPICHOST* routed topic

- ___ 45. From your command prompt window, start a runmqsc session for queue manager QMPS2 by typing `runmqsc QMPS2` and pressing the Enter key.
- ___ 46. Remove the existing TENNIS.TOPIC topic that has routing set to DIRECT from QMPS2 by typing the command as shown in the text box, and press the Enter key. Keep the `runmqsc` session open.

```
alter topic(TENNIS.TOPIC) CLUSTER(' ')
```

Expected response:

```
1 : alter topic(TENNIS.TOPIC) CLUSTER(' ')
AMQ8691: IBM MQ topic changed.
```



I

The topic object must be removed from the cluster by clearing the CLUSTER attribute before the CLROUTE parameter 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.

- ___ 47. Add the TENNIS.TOPIC to QMPS2 (the topic host) with routing set to TOPICHOST by typing the command as shown on the text box, and press the Enter key. Keep the `runmqsc` session open.

```
ALTER TOPIC(TENNIS.TOPIC) CLROUTE(TOPICHOST) CLUSTER(PSCLUS)
```

Expected response:

```
2 : ALTER TOPIC(TENNIS.TOPIC) CLROUTE(TOPICHOST) CLUSTER(PSCLUS)
AMQ8691: IBM MQ topic changed.
```

- ___ 48. From the QMPS2 runmqsc session, display the topic status to verify the topic definition by typing the command as shown in the text box, and press the Enter key. Keep the `runmqsc` session open.

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

Expected response:

```
3 : DISPLAY TCLUSTER(TENNIS.TOPIC) TOPICSTR CLUSTER CLROUTE CLSTATE
AMQ8633: Display topic details.
TOPIC(TENNIS.TOPIC)                                TYPE(CLUSTER)
TOPICSTR(/News/Sports/Tennis)                       CLUSTER(PSCLUS)
CLROUTE(TOPICHOST)                                  CLSTATE(ACTIVE)
```

- ___ 49. Type `end` and press the Enter key to end the `runmqsc` session for QMPS2.
 - ___ 50. For each of the other cluster queue managers – QMPS1, QMPS3, QMPS4, and QMPS5 – open a `runmqsc` session and verify the topic definition by typing the command as shown in the text box. Press the Enter key.
-

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

- ___ 51. Review the results. The topic status for each other cluster queue manager is expected to match the display for the QMPS2 status. However, it is possible that if a partial repository queue manager did not have the “need to know” about the topic, you might not see the status in its output.
-



I

You can also verify the topic definition in IBM MQ Explorer by examining the contents on the Cluster Topics tab for a cluster queue manager.

Section 2: Test the topic host routing by creating a test subscription on QMPS4 in IBM MQ Explorer

- ___ 52. Expand the queue manager QMPS4 under the **Queue Managers** folder in the IBM MQ Explorer Navigator view.
- ___ 53. Right-click **Topics** and then click **Test Subscription**.
- ___ 54. In the **Topic String** field, type: `/News/Sports/Tennis`
- ___ 55. Click **Subscribe**. Do not click **Close**.
- ___ 56. Publish a message on QMPS1 by using the `amqspub` sample program. If you closed the command window that was running the `amqspub` sample program:
 - ___ a. Open a new command window and type: `amqspub /News/Sports/Tennis QMPS1`
 - ___ b. Enter a message.
 - ___ c. Verify that the publication message appears in the **Subscribe** window.

57. Enter the display route command to determine whether the message was routed through the topic host by typing the command as shown in the text box, and press the Enter key.

```
dspmqrte -ts /News/Sports/Tennis -ac -d yes -v outline activity -w 3 -m QMPS1
```

Expected results:

```
AMQ8653: DSPMQRTE command started with options '-ts /News/Sports/Tennis -ac -d
yes -v outline activity -w 3 -m QMPS1'.
AMQ8694: DSPMQRTE command successfully put a message to topic string
'/News/Sports/Tennis', queue manager 'QMPS1'.
AMQ8657: DSPMQRTE command used CorrelId
0x414D5120514D5053312020202020207A4EE75420007804.
AMQ8674: DSPMQRTE command is now waiting for information to display.
```

After a brief pause, the results of the command are expected to resemble the results shown in the example that follows, which shows that the message is now routed through QMPS2 (the topic host).

Activity:

```
ApplName: 'QMPS1'
ApplType: QMgrPublish
ActivityDesc: 'Message publication'
```

Operation:

```
OperationType: Put
QMgrName: 'QMPS1'
TopicString: '/News/Sports/Tennis'
```

Activity:

```
ApplName: 'Sphere MQ\bin64\amqrmppa.exe'
ApplType: WindowsNT
ActivityDesc: 'Sending Message Channel Agent'
```

Operation:

```
OperationType: Get
QMgrName: 'QMPS1'
QName: 'SYSTEM.CLUSTER.TRANSMIT.QUEUE'
ResolvedQName: 'SYSTEM.CLUSTER.TRANSMIT.QUEUE'
```

Operation:

```
OperationType: Send
QMgrName: 'QMPS1'
RemoteQMgrName: 'QMPS2'
ChannelName: 'PSCLUS.QMPS2'
ChannelType: ClusSdr
XmitQName: 'SYSTEM.CLUSTER.TRANSMIT.QUEUE'
```

Activity:

ApplName: 'Sphere MQ\bin64\amqrmppa.exe'
 ApplType: WindowsNT
 ActivityDesc: 'Receiving Message Channel Agent'

Operation:

OperationType: Receive
 QMgrName: 'QMPS2'
 RemoteQMGrName: 'QMPS1'
 ChannelName: 'PSCLUS.QMPS2'
 ChannelType: ClusRcvr

Operation:

OperationType: Put
 QMgrName: 'QMPS2'
 QName: 'SYSTEM.INTER.QMGR.PUBS'
 ResolvedQName: 'SYSTEM.INTER.QMGR.PUBS'

Activity:

ApplName: 'QMPS2'
 ApplType: QMgrPublish
 ActivityDesc: 'Message publication'

Operation:

OperationType: Put
 QMgrName: 'QMPS2'
 TopicString: '/News/Sports/Tennis'

Operation:

OperationType: Publish
 SubId: 'X'414D5120514D505332202020202020B34EE75420000D14'
 SubLevel: 1
 QMgrName: 'QMPS2'

Activity:

ApplName: 'Sphere MQ\bin64\amqrmppa.exe'
 ApplType: WindowsNT
 ActivityDesc: 'Sending Message Channel Agent'

Operation:

OperationType: Get
 QMgrName: 'QMPS2'
 QName: 'SYSTEM.CLUSTER.TRANSMIT.QUEUE'
 ResolvedQName: 'SYSTEM.CLUSTER.TRANSMIT.QUEUE'

Operation:

OperationType: Send
 QMgrName: 'QMPS2'

```

RemoteQMgrName: 'QMPS4
ChannelName: 'PCLUS.QMPS4
ChannelType: ClusSdr
XmitQName: 'SYSTEM.CLUSTER.TRANSMIT.QUEUE
-----
Activity:
  ApplName: 'Sphere MQ\bin64\amqrmppa.exe'
  ApplType: WindowsNT
  ActivityDesc: 'Receiving Message Channel Agent '

Operation:
  OperationType: Receive
  QMgrName: 'QMPS4
  RemoteQMgrName: 'QMPS2
  ChannelName: 'PCLUS.QMPS4
  ChannelType: ClusRcvr
Operation:
  OperationType: Put
  QMgrName: 'QMPS4
  QName: 'SYSTEM.INTER.QMGR.PUBS
  ResolvedQName: 'SYSTEM.INTER.QMGR.PUBS
-----
Activity:
  ApplName: 'QMPS4
  ApplType: QMgrPublish
  ActivityDesc: 'Message publication

Operation:
  OperationType: Put
  QMgrName: 'QMPS4
  TopicString: '/News/Sports/Tennis'
Operation:
  OperationType: Publish
  SubId: X'414D5120514D5053342020202020202020924FE75420017A04'
  SubLevel: 1
  QMgrName: 'QMPS4
-----
AMQ8652: DSPMQRTE command has finished.

```

___ 58. In the amqspub window, press Enter one time to end the amqspub sample program.

6.5. Exercise clean-up

- ___ 1. Stop the cluster queue managers: QMPS1, QMPS2, QMPS3, QMPS4, and QMPS5.
- ___ 2. Close any open command windows.

End of exercise

Exercise review and wrap-up

In this exercise, you:

- Created a cluster topic and reviewed its status
- Created subscriptions and publications to a topic
- Used the dspmqrte tool to identify the path of a message in the publish/subscribe cluster
- Configured and verified topic host routing

Exercise 7. Configuring an overlapping cluster

Estimated time

01:00

Overview

In this exercise, you work with the implementation and verification of overlapping clusters.

Objectives

After completing this exercise, you should be able to:

- Implement overlapping clusters
- Verify and test the overlapping clusters

Requirements

- IBM MQ V9 and IBM MQ Explorer V9

7.1. Creating an overlapping cluster

In this exercise, you create a new cluster that is named CLUS2. As shown in Figure 6-2, CLUS2 uses the full repository queue managers from CLUS1 (QMC1 and QMC2) and two new partial repository queue managers that are named QMCA and QMCB.

The `amqsput` sample program is used to send messages to a queue that is called ACCOUNTING, which is on QMCA and QMCB.

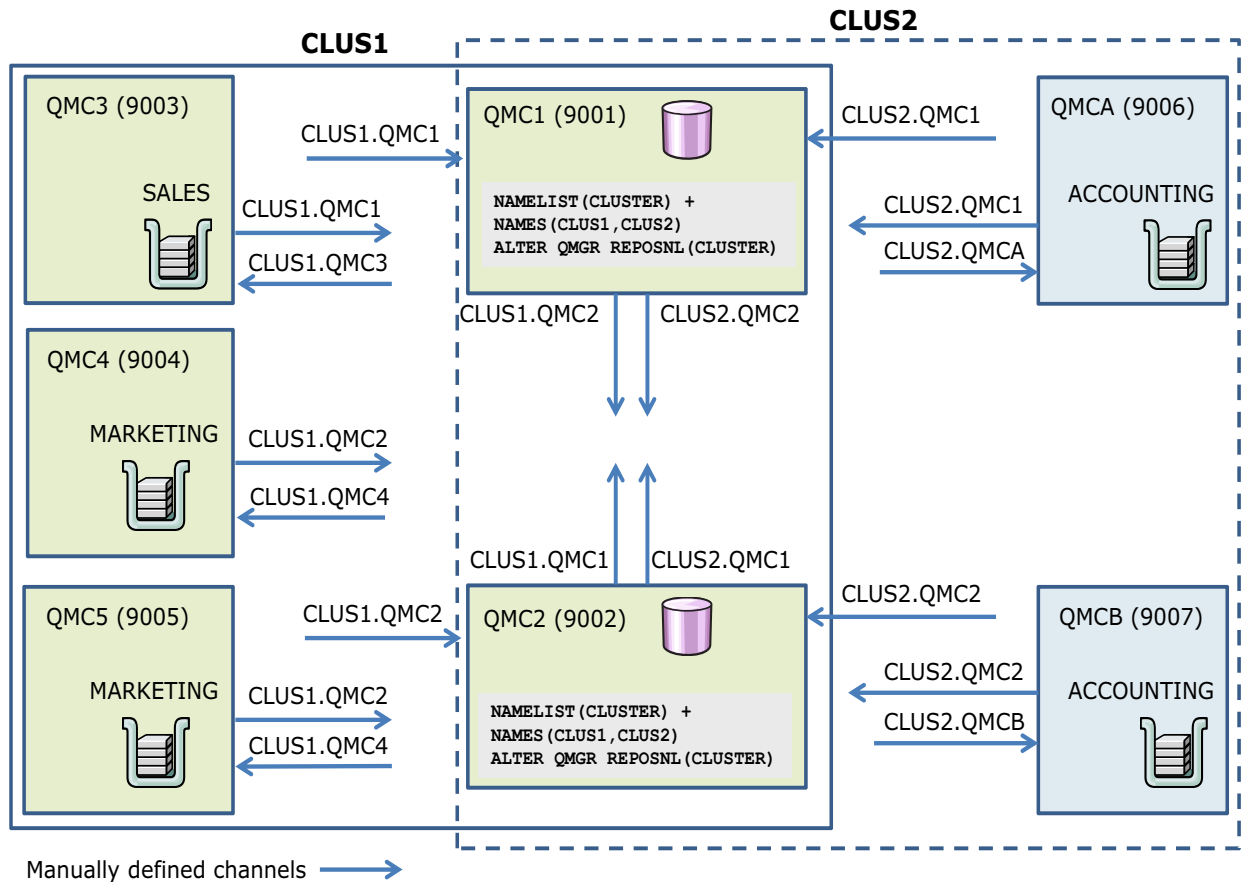


Figure 7-1. Overlapping cluster

In this part of the exercise, you revise the configuration to add the following objects:

QMC1

- Cluster-receiver channel for CLUS2
- Cluster-sender channel to QMC2 for CLUS2
- Repository for CLUS1 and CLUS2

QMC2

- Cluster-receiver channel for CLUS2
- Cluster-sender channel to QMC1 for CLUS2
- Repository for CLUS1 and CLUS2

QMCA

- Listener on 9006
- Cluster-receiver channel for CLUS2
- Cluster-sender channel to QMC1 for CLUS2
- Local queue that is called ACCOUNTING for CLUS2

QMCB

- Listener on 9007
- Cluster-receiver channel for CLUS2
- Cluster-sender channel to QMC2 for CLUS2
- Local queue that is called ACCOUNTING for CLUS2

Section 1: Ensure that the correct queue managers are active for this exercise, and any other queue managers are stopped

- ___ 1. Open a command prompt window.
- ___ 2. Display the status of the queue managers in your VMware image by typing `dspsmq` and pressing the Enter key.
- ___ 3. If any of the queue managers that are used for the publish/subscribe exercise – MQG1, MQ03, MQ0A, MQ0B, MQE5, or MQ0X – are running, stop them by typing the `endmqm -i` command followed by the queue manager name. Press the Enter key.
- ___ 4. If any of the queue managers that are used for the publish/subscribe exercise – QMPS1, QMPS2, QMPS3, QMPS4, or QMPS5 – are running, stop them by typing the `endmqm -i` command followed by the queue manager name. Press the Enter key.
- ___ 5. Look at queue managers QMC1, QMC2, QMC3, QMC4, and QMC5.
 - ___ a. If the five queue managers that are identified in this step are running, no action is necessary.
 - ___ b. If the five queue managers that are identified in this step are stopped, start them by typing `strmqm` followed by the queue manager name, and press the Enter key.

Section 2: Create the CLUS2 queue managers QMCA and QMCB

- ___ 6. Using IBM MQ Explorer, create the queue managers QMCA listening on port 9006, and QMCB listening on port 9007. Name the dead-letter queues QMCA.DLQ and QMCB.DLQ. You define the dead-letter queues later in this exercise.
- ___ 7. After you create the new queue managers, verify that the listeners are running.
- ___ 8. Configure the queue managers to be full repositories for CLUS1 and CLUS2.
 - ___ a. In MQSC for QMC1, type:


```
DEFINE NAMELIST(CLUSTER) NAMES(CLUS1, CLUS2)
ALTER QMGR REPOS(' ') REPOSNL(CLUSTER)
```
 - ___ b. In MQSC for QMC2, type:


```
DEFINE NAMELIST(CLUSTER) NAMES(CLUS1, CLUS2)
ALTER QMGR REPOS(' ') REPOSNL(CLUSTER)
```

- ___ 9. Configure the channels between the queue managers in cluster CLUS2. For the new queue managers QMCA and QMCB, define the queue that was identified as the dead-letter queue.

- ___ a. In MQSC for QMC1, type:

```
DEFINE CHANNEL(CLUS2.QMC1) CHLTYPE(CLUSRCVR) +
TRPTYPE(TCP) CONNAME('localhost(9001)') CLUSTER(CLUS2)
```

```
DEFINE CHANNEL(CLUS2.QMC2) CHLTYPE(CLUSSDR) +
TRPTYPE(TCP) CONNAME('localhost(9002)') CLUSTER(CLUS2)
```

- ___ b. In MQSC for QMC2, type:

```
DEFINE CHANNEL(CLUS2.QMC2) CHLTYPE(CLUSRCVR) +
TRPTYPE(TCP) CONNAME('localhost(9002)') CLUSTER(CLUS2)
```

```
DEFINE CHANNEL(CLUS2.QMC1) CHLTYPE(CLUSSDR) +
TRPTYPE(TCP) CONNAME('localhost(9001)') CLUSTER(CLUS2)
```

- ___ c. In MQSC for QMCA, type:

```
DEFINE CHANNEL(CLUS2.QMCA) CHLTYPE(CLUSRCVR) +
TRPTYPE(TCP) CONNAME('localhost(9006)') CLUSTER(CLUS2)
```

```
DEFINE CHANNEL(CLUS2.QMC1) CHLTYPE(CLUSSDR) +
TRPTYPE(TCP) CONNAME('localhost(9001)') CLUSTER(CLUS2)
```

```
DEF QL(QMCA.DLQ)
```

- ___ d. In MQSC for QMCB, type:

```
DEFINE CHANNEL(CLUS2.QMCB) CHLTYPE(CLUSRCVR) +
TRPTYPE(TCP) CONNAME('localhost(9007)') CLUSTER(CLUS2)
```

```
DEFINE CHANNEL(CLUS2.QMC2) CHLTYPE(CLUSSDR) +
TRPTYPE(TCP) CONNAME('localhost(9002)') CLUSTER(CLUS2)
```

```
DEF QL(QMCB.DLQ)
```

Section 3: Verify the resulting CLUS1 and CLUS2 overlapping cluster from the perspective of full repository queue manager QMC1

- ___ 10. Verify the resulting overlapping cluster by typing the command as shown in the text box for queue manager QMC1, and press the Enter key. Review the results after the display.

```
dis clusqmgr(*) deftype status conname
```

Expected results:

```
1 : dis clusqmgr(*) deftype status conname
AMQ8441: Display Cluster Queue Manager details.
    CLUSQMGR(QMC1)                      CHANNEL(CCLUS2.QMC1)
    CLUSTER(CLUS2)                      CONNAME(localhost(9001))
    DEFTYPE(CLUSRCVR)                   STATUS(RUNNING)
AMQ8441: Display Cluster Queue Manager details.
    CLUSQMGR(QMC1)                      CHANNEL(CCLUS1.QMC1)
    CLUSTER(CLUS1)                      CONNAME(localhost(9001))
    DEFTYPE(CLUSRCVR)                   STATUS(RUNNING)
AMQ8441: Display Cluster Queue Manager details.
    CLUSQMGR(QMC2)                      CHANNEL(CCLUS2.QMC2)
    CLUSTER(CLUS2)                      CONNAME(localhost(9002))
    DEFTYPE(CLUSSDRA)                   STATUS(RUNNING)
AMQ8441: Display Cluster Queue Manager details.
    CLUSQMGR(QMC2)                      CHANNEL(CCLUS1.QMC2)
    CLUSTER(CLUS1)                      CONNAME(localhost(9002))
    DEFTYPE(CLUSSDRB)                   STATUS(RUNNING)
AMQ8441: Display Cluster Queue Manager details.
    CLUSQMGR(QMC3)                      CHANNEL(CCLUS1.QMC3)
    CLUSTER(CLUS1)                      CONNAME(localhost(9003))
    DEFTYPE(CLUSSDRA)                   STATUS(RUNNING)
AMQ8441: Display Cluster Queue Manager details.
    CLUSQMGR(QMC4)                      CHANNEL(CCLUS1.QMC4)
    CLUSTER(CLUS1)                      CONNAME(localhost(9004))
    DEFTYPE(CLUSSDRA)                   STATUS(RUNNING)
AMQ8441: Display Cluster Queue Manager details.
    CLUSQMGR(QMCA)                      CHANNEL(CCLUS2.QMCA)
    CLUSTER(CLUS2)                      CONNAME(localhost(9006))
    DEFTYPE(CLUSSDRA)                   STATUS(RUNNING)
```

___ 11. Review the results.

- ___ a. In this display, you added the `CONNAME` parameter so that in case any channel status was `RETRYING`, you would be able to check the information in the `CONNAME` parameter to the retrying channel.
- ___ b. All channels are in `RUNNING` status. However, if you step away before you type the `DIS CLUSQMGR` command, you might see channel status(`INACTIVE`) in some channels. A status of `INACTIVE` is okay provided you do not see any `DEFTYPE` of `CLUSSDR` by itself (not `CLUSSDRA` or `CLUSSDRB`).
- ___ c. You now see the two `CLUSRCVR` channels for `QMC1`. You also see the cluster names to which each channel belongs in the `CLUSTER` label of the display.
- ___ d. The rest of the display is as expected, except that you now see the identification of the two different clusters.

Section 4: Add the queue ACCOUNTING to the CLUS2 queue managers QMCA and QMCB

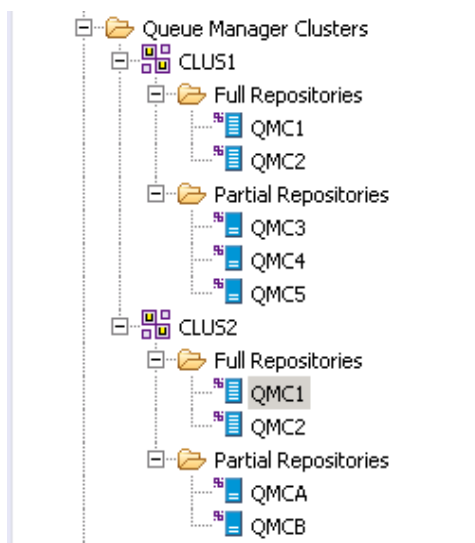
- ___ 12. In MQSC for QMCA and QMCB, type the following command:

```
DEFINE QL(ACCOUNTING) CLUSTER(CLUS2) DEFBIND(NOTFIXED)
```

- ___ a. In MQSC for QMC1, type the following command to verify that the ACCOUNTING cluster queue is defined correctly on QMCA and QMCB:

```
DIS QCLUSTER(ACCOUNTING)
```

- ___ b. View the cluster in IBM MQ Explorer. You should now see two clusters (CLUS1 and CLUS2). QMC1 and QMC2 should be listed as full repository queue managers for both clusters.



- ___ 13. Use the `amqspout` sample program on QMC1 and QMC2 to put messages to the ACCOUNTING cluster queue on QMCA and QMCB. You need to type one message for each queue manager, then press a blank line to end `amqspout`. If you need a refresher on how to use the `amqspout` command, refer to the earlier exercises.

Examples:

```
amqspout ACCOUNTING QMC1
amqspout ACCOUNTING QMC2
```

- ___ 14. Use the `amqsget` sample program as shown on the display to verify that the messages arrived on the ACCOUNTING queue that is hosted by QMCA and QMCB.

**CAUTION**

The results might vary according to how many messages you used. Concentrate on accounting for all messages that are typed, whether you manually typed messages, or whether you used the message text file.

When you use `amqsget`, the program runs for a few seconds. You can use Ctrl-C to end the program after all messages are retrieved. One possible outcome, with the command example, is below.

```
amqsget ACCOUNTING QMCA
```

Possible results:

```
Sample AMQSGET0 start  
message <this is a test to check things out>  
^C  
amqsget ACCOUNTING QMCB
```

Possible results:

```
Sample AMQSGET0 start  
message <this is another test to check things out>  
^C
```

7.2. Putting messages across the clusters

In this part of the exercise, configure the cluster so that queues can be referenced in both clusters.

First, you create a cluster queue on QMC1 that is shared by both CLUS1 and CLUS2.

Next, you create a queue alias so that you can put a message from QMCA on CLUS2 to the SALES queue on QMC3 in CLUS1.

Section 1: Create a queue that is shared by using the cluster namelist and test across clusters

- ___ 15. Create a cluster queue that is named SHAREQ on QMC1 that is shared in both clusters. In MQSC for QMC1, type:

```
DEF QL(SHAREQ) CLUSNL(CLUSTER)
```

- ___ 16. Verify that you can put messages to SHAREQ by using the `amqspout` sample program.

- ___ a. To verify that you can put to SHAREQ from a queue manager that is in CLUS1, type the `amqspout` command as shown. Type suggested message text “one message from qmc3” and press the Enter key one time to end `amqspout`.

```
amqspout SHAREQ QMC3
```

- ___ b. To verify that you can put to SHAREQ from a queue manager that is in CLUS2, type the `amqspout` command as shown. Type suggested message text “one message from qmca” and press the Enter key one time to end `amqspout`.

```
amqspout SHAREQ QMCA
```

- ___ 17. Ensure that both messages are at the SHAREQ queue at CLUS1 queue manager QMC1 by typing the command as shown in the text box, and press the Enter key. You can use Ctrl-C to break out of the `amqsget` program.

```
amqsget SHAREQ QMC1
```

Expected results:

```
Sample AMQSGETO start
message <one message from qmc3>
message <one message from qmca>
^C
```

Section 2: Create a queue alias to put messages across the two clusters and test the definitions

- ___ 18. On QMC3, define the SALES queue by typing the command as shown in the text box, and press the Enter key.

```
DEFINE QL(SALES) CLUSTER(CLUS1) DEFBIND(NOTFIXED)
```

```
1 : DEFINE QL(SALES) CLUSTER(CLUS1) DEFBIND(NOTFIXED)
AMQ8006: IBM MQ queue created.
```

- __ 19. On QMC1, which is considered the overlapping cluster bridge queue manager, create a queue alias MYSALES as shown in the text box, and press the Enter key.
-

```
DEF QA(MYSALES) TARGQ(SALES) CLUSTER(CLUS2) DEFBIND(NOTFIXED)
```

```
1 : DEF QA(MYSALES) TARGQ(SALES) CLUSTER(CLUS2) DEFBIND(NOTFIXED)
AMQ8006: IBM MQ queue created.
```



Note

When you open a queue, you need to set `DEFBIND` to either `NOTFIXED` or `QDEF`. If `DEFBIND` is left as the default `OPEN`, the queue manager resolves the alias definition to the bridge queue manager that hosts it. The bridge does not forward the message.

- __ 20. Verify that you can put a message from QMCA in CLUS2 to QMC3 in CLUS1 by using the alias queue MYSALES. Type:

```
amqsput MYSALES QMCA
```

- __ 21. Use the `amqsput` sample:
-

```
amqsget SALES QMC3
```

Expected results vary depending on messages you typed. In this example, the three messages you see were typed in the previous step.

Sample `AMQSGET0` start:

```
message <this is a couple of messages>
message <to test>
message <the last step in the exercise>
```

7.3. Exercise clean-up

Using IBM MQ Explorer, stop and delete all the queue managers in both CLUS1 and CLUS2.

End of exercise

Exercise review and wrap-up

In this exercise, you:

- Implemented overlapping clusters
- Verified and tested the overlapping clusters

