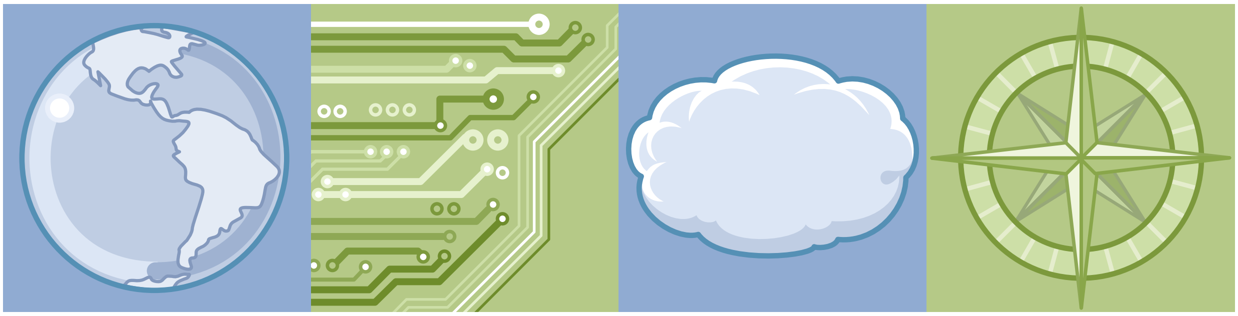
IBM Training

Student Exercises

IBM MQ V8 Advanced System Administration for z/OS

Course code WM312 ERC 1.1



IBM Systems
Middleware

Trademarks

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:

- DB2®
- HACMP™
- System i™
- System p™
- System x™
- System z™

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

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

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

July 2016 edition

The information contained in this document has not been submitted to any formal IBM test and is distributed on an “as is” basis without any warranty either express or implied. The use of this information or the implementation of any of these techniques is a customer responsibility and depends on the customer’s ability to evaluate and integrate them into the customer’s operational environment. While each item may have been reviewed by IBM for accuracy in a specific situation, there is no guarantee that the same or similar results will result elsewhere. Customers attempting to adapt these techniques to their own environments do so at their own risk.

Contents

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:

DB2®	HACMP™	System i™
System p™	System x™	System z™

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

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

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

WM312 Lab environment

Important

To log on to the lab virtual machine image, use ID `Administrator` and password `web1sphere` and then follow the instructions.

Refer to the readme.txt on the lab image desktop for possible additional information.

Online course material updates might exist for this course. To check for updates, go to the Instructor wiki at: `http://ibm.biz/CloudEduCourses`

The lab environment for course WM312 is configured to allow full autonomy to each student. Students are able to progress through each exercise without dependencies on another team.

The WM312 lab accommodates 12 students. Each student is assigned two TSO IDs, two queue managers that use the same local port, one queue-sharing group name, four coupling facility structures, and one DVIPA address and dynamic port.

Your Windows VMware image is configured with four IBM Personal Communications icons. The instructor confirms which two to use throughout the course. Disregard the unused Personal Communications icons.

Table 1:

Student numbers	LPAR1 Primary TSO ID and queue manager name	LPAR2 Secondary TSO ID and queue manager name	Local port Lab 5: generic port	Queue-sharing group name	Coupling facility structures
Primary: 01 Secondary: 21	TSM0001 MQ01	TSM0021 MQ21	3101 Lab 5: 3131	SG01	SG01CSQ_ADMIN CSQSYSAPPL SG02APPL1 SG01APPL2
Primary: 02 Secondary: 22	TSM0002 MQ02	TSM0022 MQ22	3102 Lab 5: 3132	SG02	SG02CSQ_ADMIN CSQSYSAPPL SG02APPL1 SG02APPL2
Primary: 03 Secondary: 23	TSM0003 MQ03	TSM0023 MQ23	3103 Lab 5: 3133	SG03	SG03CSQ_ADMIN CSQSYSAPPL SG03APPL1 SG03APPL2
Primary: 04 Secondary: 24	TSM0004 MQ04	TSM0024 MQ24	3104 Lab 5: 3134	SG04	SG04CSQ_ADMIN CSQSYSAPPL SG04APPL1 SG04APPL2

Table 1:

Student numbers	LPAR1 Primary TSO ID and queue manager name	LPAR2 Secondary TSO ID and queue manager name	Local port Lab 5: generic port	Queue-sharing group name	Coupling facility structures
Primary: 05 Secondary: 25	TSM0005 MQ05	TSM0025 MQ25	3105 Lab 5: 3135	SG05	SG05CSQ_ADMIN CSQSYSAPPL SG05APPL1 SG05APPL2
Primary: 06 Secondary: 26b	TSM0006 MQ06	TSM0026 MQ26	3106 Lab 5: 3136	SG06	SG06CSQ_ADMIN CSQSYSAPPL SG06APPL1 SG06APPL2
Primary: 07 Secondary: 27	TSM0007 MQ07	TSM0027 MQ27	3107 Lab 5: 3137	SG07	SG07CSQ_ADMIN CSQSYSAPPL SG07APPL1 SG07APPL2
Primary: 08 Secondary: 28	TSM0008 MQ08	TSM0028 MQ28	3108 Lab 5: 3138	SG08	SG08CSQ_ADMIN CSQSYSAPPL SG08APPL1 SG088APPL2
Primary: 09 Secondary: 29	TSM0009 MQ09	TSM0029 MQ29	3109 Lab 5: 3139	SG09	SG09CSQ_ADMIN CSQSYSAPPL SG09APPL1 SG09APPL2
Primary: 10 Secondary: 30	TSM0010 MQ10	TSM0030 MQ30	3110 Lab 5: 3140	SG10	SG10CSQ_ADMIN CSQSYSAPPL SG10APPL1 SG10APPL2
Primary: 11 Secondary: 31	TSM0011 MQ11	TSM0031 MQ31	3111 Lab 5: 3141	SG11	SG01CSQ_ADMIN CSQSYSAPPL SG11APPL1 SG11APPL2
Primary: 12 Secondary: 32	TSM0012 MQ12	TSM0012 MQ32	3112 Lab 5: 3142	SG12	SG12CSQ_ADMIN CSQSYSAPPL SG12APPL1 SG12APPL2

Terminology baseline

Each student is assigned a primary student number, and owns two student numbers throughout the course. The primary student number is in the range 1 – 12. The secondary student number is obtained by adding 20 to the primary student number. For example, if you are assigned student number 05, you “own” student number 05 and student number 25.

TSO IDs and queue manager names have an alphabetic and a numeric portion. To promote autonomy throughout the lab exercises, each student works with two queue managers and two corresponding TSO IDs. The numeric portion of the two student TSO IDs and the numeric portions of the two student queue manager names match the corresponding primary and secondary student numbers.

- Throughout the exercises, the placeholder for the primary queue manager and other objects that require a number equal to your primary student number is a double “pound” symbol, or `##`. **You replace the `##` with your primary student number**.
- Throughout the exercises, the placeholder for the secondary queue manager and other objects that require a number equal to your secondary student number is a double “at” symbol, or `@@`. **You replace the `@@` with your secondary student number.**

Warning

Always enter the primary student number as two digits: 1 as 01, 8 as 08, 12 as 12.

Use of single digits results in undesirable outcomes in the lab exercises.

The term QM might be used to denote “queue manager”.

Important

To indicate which TSO ID to use for the different sections of an exercise, a box with the “Important” label similar to this text box precedes the instructions when it is time to change TSO IDs. Be on the lookout for a box with these instructions throughout the exercises.

If you need to confirm what ID you should be using, you can always scroll back in the exercise to the previous “Important” box with TSO ID instructions.

Exercises description

This course includes the following exercises:

- Exercise 1: Configure IBM MQ lab environment. In this exercise, you become familiar with the lab environment and establish the channels that are needed for the SSL lab. Do not worry if you need to temporarily disable channel authentication. You obtain ample practice with channel authentication rules in Exercises 3 and 5. Ensure that your channels are running before you start Exercise 2.
- Exercise 2: Configure SSL/TLS for queue managers and channels. The first part of this exercise reinforces implementation of SSL across sender-receiver channels in two z/OS queue managers. In the second part of the exercise, you add a second signer certificate. You use the multiple certificate feature of IBM MQ V8 while you set up mutual

authentication from the Windows client to the primary z/OS queue manager. You use the outcome of the SSL work in this exercise to set up channel authentication SSLPEERMAP rules in Exercise 3.

- Exercise 3: Configure channel authentication rules. In this exercise, you first learn how to work with the built-in channel authentication rules without removing them. Next, you learn to set the back-stop rule to block all connections. You then set rules to allow the channels to connect by using the rules. You use the material learned in this lab for a portion of Exercise 5 after you incorporate the queue-sharing group into the cluster.
- Exercise 4. Configure queue-sharing groups and shared queues.
- Exercise 5. Intra-group and distributed queuing with queue-sharing groups.
- Exercise 6. Implementing 8-byte record byte address and buffers above the 2-GB line.
- Exercise 7. Getting started with IBM MQ statistics and accounting records.

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

Exercise 1. Configuring the IBM MQ lab environment

What this exercise is about

In this exercise, you initialize your lab environment and create and test the IBM MQ definitions that are used in the SSL/TLS exercise.

What you should be able to do

After completing this exercise, you should be able to:

- Assemble the queue manager parameter modules
- Start the two student queue managers and channel initiators
- Configure sender receiver channels across the queue managers
- Exchange messages across queue managers
- Set up an IBM MQ client channel definition table connection to a z/OS server connection channel
- Exchange messages from the client to the queue manager
- Add queue managers to IBM MQ Explorer

Introduction

This exercise lays the foundation for the lab exercises in this course. It is critical that the queue manager and client connectivity work successfully before you move to the SSL/TLS work in Exercise 2, or to any other exercise.

Requirements

- Two TSO IDs in separate LPARs
- Two partially preconfigured queue managers
- Corresponding `PDS SCSQPROC` libraries

Exercise instructions

In this exercise, you initialize your lab environment and create and test the IBM MQ definitions that are used in the SSL/TLS exercise.

Important

Ensure that you read the terminology baseline section before you proceed. If you have any questions, consult with your instructor.

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

- `C:\labfiles` for the Windows platform. The exercises point you to the lab files when needed.
- The full contents of the lab files also appear in each corresponding exercise. However, you are requested to use the lab files to prevent copying unexpected extraneous characters that might result in errors that can take time to resolve.
- The student z/OS JCL library for this course is found as follows:
  - For the primary LPAR, `TSM00##.WM31.SCSQPROC` where the `##` placeholder stands for the primary assigned student number.
  - For the secondary LPAR, `TSM00@@.WM31.SCSQPROC` where the `@@` placeholder stands for the secondary student number.

Day 1 environment

The two queue managers that you use in these exercises are partially configured. The z/OS work that is required from IBM MQ on z/OS is complete. Logs and bootstrap data sets are created, and queue manager and channel initiator procedures are moved to the designated libraries. For your primary and secondary queue managers, you assemble the corresponding parameter modules and start the queue managers. The corresponding channel initiator tasks are configured to start automatically.

After your queue managers start, you create and test the message and client channel definitions in preparation for the SSL/TLS exercise.

Figure 1-1 illustrates the configuration that you create for the SSL/TLS work after the queue managers are started in Exercise 1.

An extra server connection channel is used with IBM MQ Explorer; it is `TSM00##.XPL`, which is not shown in the figure. The definition of TSM00##.XPL is included in the PDS members that contain definitions for each queue manager.

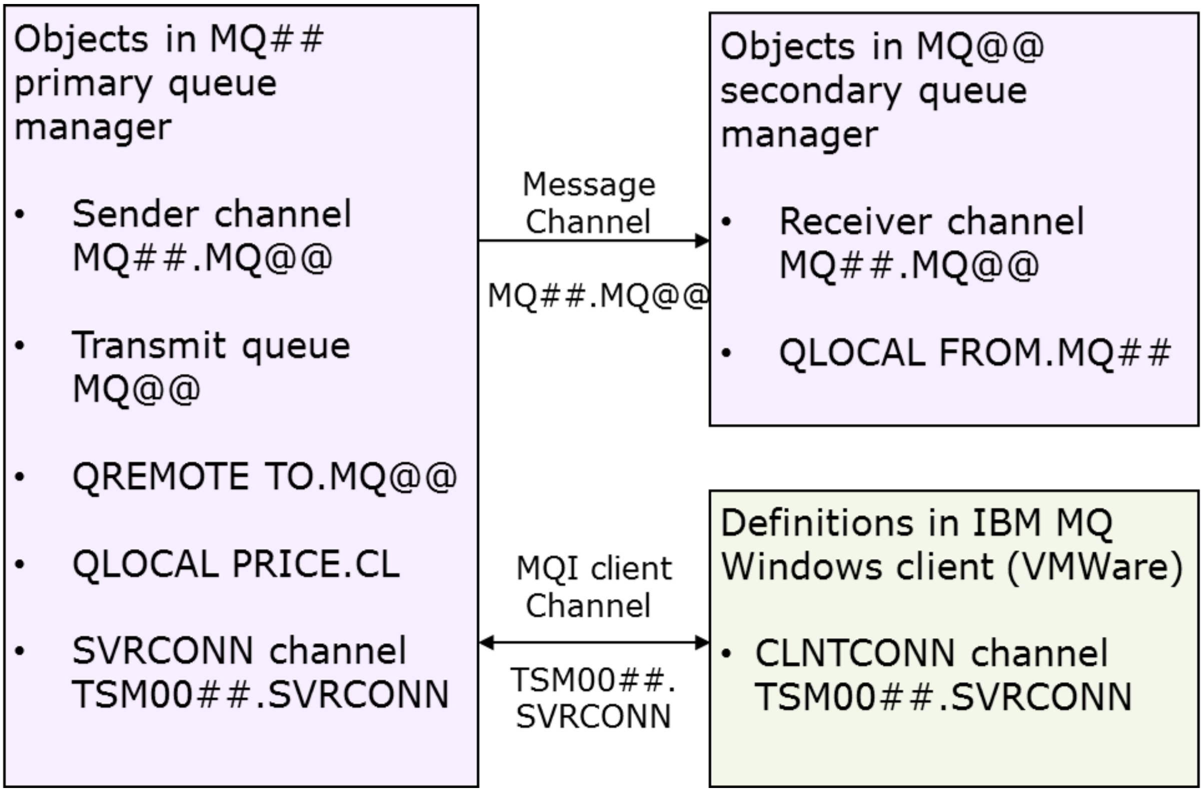

Objects in MQ## primary queue manager
• Sender channel MQ##.MQ@@
• Transmit queue MQ@@
• QREMOTE TO.MQ@@
• QLOCAL PRICE.CL
• SVRCONN channel TSM00##.SVRCONN
Message Channel
MQ##.MQ@@
Objects in MQ@@ secondary queue manager
• Receiver channel MQ##.MQ@@
• QLOCAL FROM.MQ##
MQI client Channel
TSM00##. SVRCONN
Definitions in IBM MQ Windows client (VMWare)
• CLNTCONN channel TSM00##.SVRCONN

Figure 1-1. Lab environment configuration after Exercise 1 WM3121.1

Section 1: Record details about your lab environment

You obtain part of the information from the instructor. You determine part of the information from the primary student number. Details to log on to the course and lab environments are provided separately by the instructor.

Table 2: Lab information

Item	Value

z/OS image host name for LPAR1:
`+++++++.ilsvpn.ibm.com`
`+++++++` value is **either** `mvsmb11` **or** `mvsmb13`
	____________`.ilsvpn.ibm.com`

z/OS image host name for LPAR2:
`+++++++.ilsvpn.ibm.com`
`+++++++` value is **either** `mvsmb12` **or** `mvsmb14`
	____________`.ilsvpn.ibm.com`

Primary student number	
Secondary student number (primary student number + 20)	

Table 2: Lab information

Item	Value
Primary TSO ID `TSM00##` (replace `##` with your primary student number)	
Secondary TSO ID `TSM00@@` (replace `@@` with your secondary student number)	
Primary queue manager `MQ##` at port `16##`	`MQ`____ at port 31____ Both primary and secondary QMs must use the same port number.
Secondary queue manager `MQ@@` at port `16@@`	`MQ`____ at port 31____ Both primary and secondary QMs must use the same port number.
TSO passwords	

Section 2: .Configure and start the primary queue manager

Important

Log on with your primary TSO ID.

Throughout all exercises in this course, each time you log on to a TSO session, ensure that you log on to the correct LPAR.

To confirm the system that you are logged on to, from the ISPF primary option menu, scroll to the lower right of the screen and check the System ID field:

- For your primary ID, `TSM00##`, System ID should be `S101`
- For your secondary ID, `TSM00@@`, System ID should be `S102`

__ 1. As soon as you reach ISPF, place the cursor on the uppermost line and press the PF2 key to split the screen, so that you can switch between two ISPF panels by using the PF9 key.

__ 2. Type `=3.4` in the ISPF `Option ==>` prompt and press the Enter (Ctrl) key.

__ 3. If your primary TSO ID is not already present in the `Dsname Level` input line, type your primary TSO ID in the `Dsname Level` line and press the Enter (Ctrl) key.

__ 4. You should see the data sets required for your primary queue manager listed. Scroll down until you find your `TSM00##.WM31.SCSQPROC` library and open it for edit by typing an `E` to the left of the data set name and pressing the Enter (Ctrl) key.

Note

In the interest of time, most parameters are configured ahead of time. You change the OPMODE parameter of the CSQ6SYSP macro. Care must be taken to preserve the position of the “X” continuation character to ensure that the macros assemble and link successfully.

Instructions are provided to start your primary queue manager, `MQ##`. You repeat these instructions for your secondary queue manager.

__ 5. Open `MQ##ZPRM` where `##` is your primary student number. `MQ##ZPRM` is a partially preconfigured copy of `CSQ4ZPRM`.

__ 6. Find the start of the `CSQ6SYSP` macro and find the `OPMODE` parameter. Use the edit command input line to change the value of `OPMODE` from `COMPAT` to `NEWFUNC` as illustrated in the display. ***Ensure that you use uppercase***:

```
C COMPAT NEWFUNC
```

__ 7. Check that the change is in uppercase, and the continuation character, `X`, is still lined up at the right column. If it moved, adjust it to column 72.

__ 8. Submit your `MQ##ZPRM` job and check results. Make any corrections if needed. If you have a problem, check that all the continuation characters in the `MQ##ZPRM` JCL member line up. A misplaced continuation character is a common cause of assembly issues.

__ 9. After your parameter module assembles and successfully, the linked module is placed on the correct library. You are ready to start your queue manager.

__ 10. The command prefix character for IBM MQ in the lab environment for this course is null. Use only a forward slash (`/`) when you enter IBM MQ commands in the z/OS console view.

__ 11. Proceed to the z/OS log from ISPF. In the ISPF command input prompt, type `=s;log` and press the Enter (Ctrl) key.

__ 12. Start your primary queue manager. **Always include the parameter module option when starting the queue managers for this course**:

```
/MQ## START QMGR PARM(MQ##ZPRM)
```

__ 13. After you enter the command, follow the progress in the z/OS log.

__ 14. Check the started tasks for your `MQ##MSTR` and `MQ##CHIN` jobs by selecting them from the z/OS job list. Type `=s;da` or `=s;st` in the ISPF command input prompt.

__ 15. Carefully review both started tasks to ensure that they are without errors. If you encounter any error messages, consult with your instructor before proceeding.

Section 3: Configure and start the secondary queue manager

Important

This section is a repetition of the steps that are detailed for the primary queue manager. You work with your secondary TSO ID, and all substitutions to the `@@` placeholders use your secondary student number. Your secondary student number is the sum of your primary student number and the number 20.

Log on to a new TSO session with your secondary TSO ID.

__ 16. Proceed to your list of data sets by typing `=3.4` in the command input prompt and pressing the Enter (Ctrl) key.

__ 17. You should see the data sets for your secondary queue manager listed. Scroll down until you find your `TSM00@@.WM31. SCSQPROC` library and open it for edit by typing an `E` to the left of the data set name and pressing the Enter (Ctrl) key.

__ 18. Select member **`MQ@@ZPRM`** where `@@` is your secondary student number`.`

__ 19. Find the start of the `CSQ6SYSP` macro and find the **`OPMODE`** parameter. Use the edit panel command input line to change the value of `OPMODE` from `COMPAT` to `NEWFUNC` as illustrated in the display:

```
C COMPAT NEWFUNC
```

__ 20. Check that the continuation character, `X`, is still lined up at the right column. If it moved, adjust it to column 72.

__ 21. Submit your `MQ@@ZPRM` job and check results. Make any corrections if needed. If you have a problem, check that all the continuation characters in the `MQ@@ZPRM` JCL member line up. A misplaced continuation character is a common cause of assembly issues.

__ 22. Proceed to the z/OS log from ISPF. In the ISPF command input prompt, type `=s;log` and press the Enter (Ctrl) key.

__ 23. Start your primary queue manager. Always include the parameter module option when starting the queue managers for this course:

```
/MQ@@ START QMGR PARM(MQ@@ZPRM)
```

__ 24. After you enter the command, follow the progress in the z/OS log.

__ 25. Check the started tasks for your `MQ@@MSTR` and `MQ@@CHIN` jobs by selecting them from the z/OS job list. Type `=s;da` or `=s;st` in the ISPF command input prompt.

__ 26. Carefully review both started tasks to ensure that they are without errors. If you encounter any error messages, consult with your instructor before proceeding.

Section 4: Temporarily disable channel authentication

Important

Log on with your primary TSO ID.

Note

After you complete the channel authentication exercise and have practice with CHLAUTH rules, you are encouraged to leave channel authentication enabled throughout the course. You have extra channel authentication work in the queue-sharing group distributed queuing exercise. However, first you must complete the environment setup and SSL labs. ***To save time now, disable channel authentication.*** You learn to make all necessary CHLAUTH adjustments for your SSL and other channels in the SSL exercise, and later in the queue-sharing distributed queuing exercise.

__ 27. Disable channel authentication in the queue manager by typing the command that is shown from one of the SDSF channels. Replace the `##` placeholder with your primary student ID:

```
/MQ## ALTER QMGR CHLAUTH(DISABLED)
```

Important

If you restart the queue manager, you must type the `ALTER QMGR CHLAUTH(DISABLED)` command again. ***You should not need to restart the queue manager until Exercise 4***.

You need to permanently enable `CHLAUTH` for Exercise 3, and leave it enabled, so you do not make a permanent change. ***Do not alter `CSQ4IYNG` to disable `CHLAUTH`***.

Section 5: Create the required IBM MQ object definitions for the primary queue manager

__ 28. Scroll down until you find your `TSM00##.WM31.SCSQPROC` library and open it for edit by typing an `E` to the left of the data set name and pressing the Enter (Ctrl) key.

__ 29. Select member **`I111MQSC`**.

__ a. Change all instances of the `##` placeholder to your primary student number.

__ b. Change all instances of the `@@` placeholder to your secondary student number.

__ c. Replace the `+++++++` placeholder with the first node of the secondary LPAR host name, for example: `mvsmb12` or `mvsmb14`

These names change per class; ensure that you use the correct name as confirmed for your class.

__ d. Review your changes and the definitions that are made with this JCL.

__ e. Submit the job and check the results. Do not rely on the condition code alone; go to SDSF, and check the job output and definition messages.

__ f. If necessary, correct any errors. Your results should look like the example output:

```
CSQU057I 8 commands read
CSQU058I 8 commands issued and responses received, 0 failed
... ... ...
```

Section 6: Create the required IBM MQ object definitions for the secondary queue manager

Important

Log on to a second session with your secondary TSO ID to complete the work in this section.

You might want to leave your primary TSO session open, as you return to use it soon.

__ 30. Scroll down until you find your `TSM00##.WM31. SCSQPROC` library and open it for edit by typing an `E` to the left of the data set name and pressing the Enter (Ctrl) key.

__ 31. Select member **`I121MQSC`**.

__ a. Change all instances of the `##` placeholder to your primary student number.

__ b. Change all instances of the `@@` placeholder to your secondary student number.

__ c. Replace the `+++++++` placeholder with the first node of the primary LPAR host name, for example: `mvsmb11` or `mvsmb13`

These names change per class; ensure that you use the correct name as confirmed for your class.

__ d. Review your changes and the definitions that are made with this JCL.

__ e. Submit the job and check the results. Do not rely on the condition code alone; go to SDSF, and check the job output and definition messages.

__ f. If necessary, correct any errors.

Section 7: Start the message channel and send messages

Important

Log on, or switch to, the primary TSO ID session. Keep your secondary TSO ID session open.

__ 32. Proceed to the SDSF log panel by typing `=s;log` at the command input prompt and pressing the Enter (Ctrl) key.

__ 33. Type the command to start the channel from your primary queue manager. Replace the placeholders with your primary and secondary student numbers:

```
/MQ## START CHL(MQ##.MQ@@)
```

__ 34. Review the results to ensure that the channel started. If there are any problems, correct them before you proceed.

__ 35. Send some messages across the channel by using the remote queue that is defined in your primary queue manager.

__ a. Open member `JCLPUT` from your `TSM00##.WM31.SCSQPROC` PDS.

__ b. Change all instances of the `##` placeholder to your primary student number.

__ c. Change the queue name parameter to **`TO.MQ@@`** where `@@` is your secondary student number. An example of the parameter line of your JCL, which uses MQ01 for illustrative purposes, is provided:

```
// PARM=('MQ01 TO.MQ21 0003 1 0100 P')
```

__ d. Submit the job and check results.

__ 36. Change to your secondary TSO ID logon session.

__ 37. Check your messages by using the message handler utility. Type `=H` in your command input prompt and press the Enter (Ctrl) key.

__ a. Type `MQ@@` for the queue manager name.

__ b. Type `FROM.MQ##` for the queue name.

__ c. The message payload consists of 100 occurrences of the number 1.

__ 38. Log off your secondary TSO session.

Section 8: Configure the IBM MQ client environment

Important

In this section, you work with your Windows VMware image.

__ 39. Proceed to your IBM MQ client in your Windows VMware session.

__ 40. Open a Windows command input prompt. Move to the `C:\LabFiles` directory.

__ 41. From the `C:\LabFiles` directory, open file `1defCltChl.mqsc` for edit. You can use Notepad.

__ 42. Review the contest of the `1defCltChl.mqsc` file:

```
DEFINE CHL(TSM00##.SVRCONN) CHLTYPE(CLNTCONN) TRPTYPE(TCP) +
CONNAME('+++++++.ilsvpn.ibm.com(31##)')
```

__ 43. Change both the `##` placeholder and the ?? placeholder to your primary student number.

__ 44. Change the `+++++++` placeholder to the first node of the z/OS **LPAR1** host name. This node is either `mvsmb11` ***or*** `mvsmb13` as provided by your instructor.

__ 45. If you are not in the `LabFiles` directory, change to that directory in your command input prompt window.

__ 46. Enter the command `runmqsc -n < 1defCltChl.mqsc` and press the Enter key.

__ 47. Ensure that there are no error messages.

__ 48. Confirm that the channel and connection parameters were defined correctly by displaying the information in the runmqsc session.

__ 49. Invoke the MQSC script by typing the command as shown in this step and pressing the Enter key:

```
runmqsc -n
```

__ 50. Type the sample command as shown in this step. The output example uses MQ05 for illustrative purposes:

```
dis chl(*) conname
     3 : dis chl(*) connameAMQ8414: Display Channel details.
   CHANNEL(TSM0005.SVRCONN)                CHLTYPE(CLNTCONN)
   CONNAME(mvsmb11.ilsvpn.ibm.com(3105))
end
```

__ 51. Type `exit` to end the command script shell, and press the Enter key.

Section 9: Test the IBM MQ client connection

Important

In this section, you work with the Windows VMware image and with the primary TSO ID session.

__ 52. Use the `amqsputc` sample program to test the IBM MQ client connectivity. The `amqsputc` program writes one message per line of input, and ends when it receives a blank line; that is, the Enter key is pressed with no input. From your Windows command prompt, type:

```
amqsputc PRICE.CL
```

__ 53. If your IBM MQ MQI channels are configured correctly, `amqsputc` should return the response in the display. ***Do not press the Enter key until you are explicitly asked***.

```
Sample AMQSPUT0 start
target queue is PRICE.CL
```

__ 54. You do not press the Enter key until you complete the next few steps.

__ 55. If your results are not the same, make any necessary corrections before you proceed. If you do not see any error messages in your Windows session, you can switch to the z/OS system log to see whether any error messages are associated with your `TSM00##.SVRCONN` channel.

__ 56. After you receive the expected results from the “`amqsputc PRICE.CL`” test, your channel is now started, although you did not send any messages yet. Go to a z/OS system log panel in your primary TSO ID session. Type `=s;log` in the command input prompt and press the Enter (Ctrl) key.

__ 57. Look or do a find for the name of your MQI channel (`TSM00##.SVRCONN`) in the z/OS log. There should be a message to confirm that your channel started:

```
CSQX511I MQ21 CSQXRESP Channel TSM00##.SVRCONN started ...1
connection 19.4.117.156
```

__ 58. While the channel for `amqsputc` is running, display the channel status on the z/OS session. Type the command that is shown in the display and press the Enter (Ctrl) key:

```
/MQ## DIS CHS(TSM00##.SVRCONN)
```

__ 59. Your results should resemble the display:

```
CSQM201I MQ## CSQMDRTC  DIS CHSTATUS DETAILS 088
CHSTATUS(TSM00##.SVRCONN)
CHLDISP(PRIVATE)
CONNAME(19.4.117.156)
CURRENT
CHLTYPE(SVRCONN)
STATUS(RUNNING)
SUBSTATE(RECEIVE)
STOPREQ(NO)
RAPPLTAG(C: mqv8 bin64 amqsputc.exe)
 END CHSTATUS DETAILS
```

__ 60. Return to your Windows VMware `command window`.

__ 61. Type one or two messages in the active `amqsputc` session.

__ 62. Enter a blank line to end the program.

__ 63. Check your messages on your primary TSO ID session. Type `=H` in a command input prompt and press the Enter (Ctrl) key to access the message handler utility.

__ 64. Type `MQ##` for the queue manager name, and `PRICE.CL` for the queue name. Your results should be similar to the display:

```
Queue Manager     : MQ21                                              :
 Queue             : PRICE.CL                                           :

 Message number   01 of 03                           Total Msgs : 000

 Msg  Put Date    Put Time  Format     User       Put Application
 No   MM/DD/YYYY  HH:MM:SS   Name    Identifier   Type      Name
 01   05/06/2015  16:39:41 MQSTR    ADMINISTRATO 00000011 Sphere MQ\bin64\amqsp
 02   05/06/2015  16:39:41 MQSTR    ADMINISTRATO 00000011 Sphere MQ\bin64\amqsp
 03   05/06/2015  16:39:42 MQSTR    ADMINISTRATO 00000011 Sphere MQ\bin64\amqsp
```

Information

Notice the User Identifier field in the display, ADMINISTRATO, truncated from the Administrator user ID from the Windows VMware session.

When you assembled your CSQ4ZPRM module, you changed the OPMODE to NEWFUNC, which allows the 12-character ID to be used in the z/OS environment.

__ 65. Look at one of the messages. Data conversion is not done by the message handler program, so the data is not converted to EBCDIC; it is still in ASCII format. For the purposes of this course, it is not necessary to convert this data

__ 66. You might want to leave your primary TSO ID session open in case you need it in the next section.

Section 10:Add your queue managers to IBM MQ Explorer

__ 67. Open IBM MQ Explorer by double-clicking the IBM MQ Explorer icon on your Windows VMware image.

__ 68. In the “IBM MQ Explorer - Navigator” pane, right-click the **Queue Managers** menu.

__ 69. Select **Add Remote Queue Manager**. The “Add Queue Manager” panel opens. Type your primary queue manager name, `MQ##`, where `##` is your primary student number. Click **Next**.

__ 70. Complete the required details in this panel:

__ a. Host name or IP address: `+++++++.ilsvpn.ibm.com` where `+++++++` is the first node of the server name, which your instructor confirms.

__ b. Port number: `31##` where `##` is your primary student number.

__ c. Server-connection channel: `TSM00##.XPL` where `##` is your primary student number.

__ d. Click **Next** two times to reach the panel titled *Specify user identification details*.

__ e. Check the box that is named *Enable user identification.*

__ f. Type your primary user ID in the box labeled `Userid`.

__ ***g. Leave the password field blank.***

__ h. Click **Finish**.

__ 71. After a brief moment, the icon to the left of the queue manager name under the Queue Managers menu in IBM MQ Explorer should turn yellow. You should see an arrow to the left of this icon, indicating that you can click to show the needed details in the `MQ##` queue manager.

__ 72. If your queue manager was not successfully added to IBM MQ Explorer, resolve the problem before you continue. Some of the steps you can take are:

__ a. Check all the connection information, host name, port, and channel name to ensure that there are no errors.

__ b. Go to your z/OS system log and look for any error messages, specifically for your TSM00##.XPL channel. You might do a find for the channel name in the SDSF log panel.

__ 73. If your primary queue manager is successfully added to IBM MQ Explorer, repeat the same process to add your secondary queue manager to IBM MQ Explorer.

Important

Exit IBM MQ Explorer; select **File > Exit**.

If your message and client channels started successfully, you completed Exercise 1.

End of exercise

Exercise review and wrap-up

In this exercise, you:

- Became familiar with the lab environment
- Completed configuration and started the two IBM MQ z/OS queue managers
- Created and started message channels
- Created and started the client, or MQI channels
- Sent messages across all channels
- Added the queue managers to IBM MQ Explorer

Exercise 2. Configuring SSL/TLS for queue managers and channels

What this exercise is about

This exercise provides hands-on experience with the RACF RACDCERT utility to implement SSL/TLS in a z/OS queue manager. The exercise then details how to implement SSL/TLS in selected queue manager and client channels.

What you should be able to do

After completing this exercise, you should be able to:

- Generate a certificate authority (CA) signer certificate
- Create a default queue manager certificate that is signed by the local CA
- Display contents of a certificate
- Create a queue manager key ring
- Connect certificates to the queue manager key ring
- Display certificates in a key ring
- Configure the queue manager to enable use of SSL/TLS
- Connect a sender-receiver channel pair that uses SSL/TLS
- Resolve date range conflicts between signer and personal certificates
- Configure use of multiple CA certificates for the same queue manager
- Configure an SSL/TLS IBM MQ client to queue manager connection

Introduction

In this exercise, you learn to implement SSL/TLS for two common scenarios:

- Sender-receiver channel across two queue managers
- IBM MQ client that uses client channel definition table (CCDT) to a server connection channel in a queue manager

You use the `RADCERT` utility to create a certificate authority (CA) signer. Then, this CA signer is going to sign certificates for you to use. To illustrate the ability to have different CA signers in the same queue manager, in the second part of this exercise you create a different CA signer for the IBM MQ client connection. Your primary queue manager is configured with two different CAs and two different certificates, each signed by a different CA.

Requirements

This exercise requires ***successful*** completion of Exercise 1.

-

Information common to Scenarios 1 and 2 of this exercise

The detailed instructions for this exercise reference a number of JCL members in your **TSM00##.WM31.SCSQPROC** or **TSM00@@.WM31.SCSQPROC** data sets. Both data sets have the same JCL members. **It is important to be logged on with the correct TSO ID when submitting each job.** The JCL members and the required TSO IDs are summarized in Table 1. The “QM” acronym is used to denote “queue manager”.

Use the JCL to organize and track the work that is done, and the steps to be completed. This course uses JCL for all RACDCERT work. Do not use the RACF ISPF panels.

Detailed instructions for each scenario are provided throughout the exercise.

Table 3: Exercise 1 TSL/SSL JCL member and TSO IDs.

TSO ID	JCL member name	Purpose
TSM00##	R111CRCA	Create signer certificate ST##CA1 for the primary QM
TSM00##	R112CRCR	Create the first primary QM certificate ibmWebSphereMQMQ##
TSM00##	R113CRRG	Create a certificate ring for the primary QM.
TSM00##	R114CNRG	Connect ST##CA1 and primary QM certificates to the primary QM ring.
TSM00##	R115DSRG	Display certificates in the primary QM ring
TSM00##	R116CHQM	Enable use of SSL/TLS in primary QM (CSQUTIL).
TSM00##	R117CHCL	Change the MQ##.MQ@@ sender channel to use SSL/TLS.
TSO ID changes		
TSM00@@	R121CRC2	Create a secondary QM certificate.
TSM00@@	R122CRR2	Create a certificate ring for the secondary QM.
TSM00@@	R123CNR2	Connect ST##CA1 and secondary QM certificates to the secondary QM ring.
TSM00@@	R124DSR2	Display certificates in the secondary QM ring
TSM00@@	R125CHQ2	Enable use of SSL/TLS in secondary QM (CSQUTIL).
TSM00@@	R126CHC2	Change the MQ##.MQ@@ receiver channel to use SSL/TLS.
TSO ID changes		

Table 3: Exercise 1 TSL/SSL JCL member and TSO IDs.

TSO ID	JCL member name	Purpose
TSM00##	R131MQPT	Send messages from primary to secondary QM over the SSL/TLS configured channel.
End of section 1		
TSM00##	R211CRCA	Create signer certificate ST##CA2 for the primary QM.
TSM00##	R212CRCS	Create a second primary QM certificate secondCAMQ## signed with ST##CA2.
TSM00##	R213CRCW	Create a personal certificate for a Windows client signed with ST##CA2.
TSM00##	R214CNC2	Connect *only* ST##CA2 and second primary QM certificates to primary QM ring.
TSM00##	R216EXCA	Export ST##CA2 certificate to a data set.
TSM00##	R217EXWC	Export Windows client personal certificate to a data set.
TSM00##	R218CHSV	Change the TSM00##.SVRCONN server connection channel to use SSL/TLS.
Optional JCL not part of instructions		
TSM00## or TSM00@@	R991DLCA	Delete a CA signer certificate from the RACF database.
TSM00## or TSM00@@	R992DLCR	Delete a personal certificate from the RACF database.
TSM00## or TSM00@@	R993RMFR	Remove a certificate from the QM ring.
TSM00## or TSM00@@	R994DSCA	List the contents of a specific CA signer certificate.

Hint

The *Security Server RACF Command Language Reference* document page is available at C:\WM312\Documents\ichza4c0.pdf. If you scroll down the bookmarks pane, you find the different RACDCERT commands. If you have problems or questions with RACDCERT commands, you might want to use this document as a reference. You might also want to make a note of this document for future use.

Caveats

The steps to configure SSL/TLS in this exercise are expected to complete without problems. It is critical to pay particular attention to the warning notes when you submit any of the `RACDCERT` jobs.

Warning

Keep these warnings present when working with all `RACDCERT` jobs throughout this exercise.

Always check the job output contents for the READY display confirmation.

- **Do not check only for return code 0; the return code is often misleading.**
- You must check under the `READY` response that the command ended successfully and no errors are present. The example shows a good completion. You see `READY` before and after the `RACDCERT` command, and no warning or error messages are present:

```
READY
RACDCERT CERTAUTH GENCERT SUBJECTSDN(CN('STUDENT 01 CA1') T('IBM MQ')
OU('TRAINIING') O('IBM') L('HURSLEY') SP('HAMPSHIRE') C('UK'))
WITHLABEL('ST01CA1') KEYUSAGE(CERTSIGN)
READY
```

Be careful with related certificate valid dates.

- If the queue manager certificates are created with a higher valid date than the signer certificate, you might see the error that is shown in bold:

```
READY
RACDCERT ID(TSM0001) GENCERT SUBJECTSDN(CN('MQ01 PERSONAL') T('QMGR')
OU('TRAIN
WITHLABEL('ibmWebSphereMQMQ01') SIGNWITH(CERTAUTH LABEL('ST01CA1'))
The certificate that you are creating has an incorrect date range. The
certificate is added with NOTRUST status.
READY
```

- If you see this error **you must follow the steps that are detailed in section *Removing and replacing a queue manager certificate*** before proceeding. This process involves deleting the certificate and re-creating it with a valid date range.
- It is important to check the results of each job carefully before proceeding to the next job so that fewer steps must be taken to reverse an incorrect situation.

Scenario 1 exercise instructions

The first SSL/TLS configuration scenario is illustrated in Figure 2-1. In this scenario:

- Both queue managers use CA certificates signed by the same CA, `ST##CA1`, where `##` stands for your primary student number.
- Both queue managers in Scenario 1 use the IBM MQ standard certificate label name, `ibmWebSphereMQ<queue_manager_name>`
- Scenario 1 corresponds to section *Using CA-signed certificates for mutual authentication of two queue managers* in the IBM Knowledge Center.
- `TSM00##` denotes your primary, or lowest numbered TSO ID or queue manager.
- `TSM00@@` denotes your secondary, or highest numbered TSO ID or queue manager.

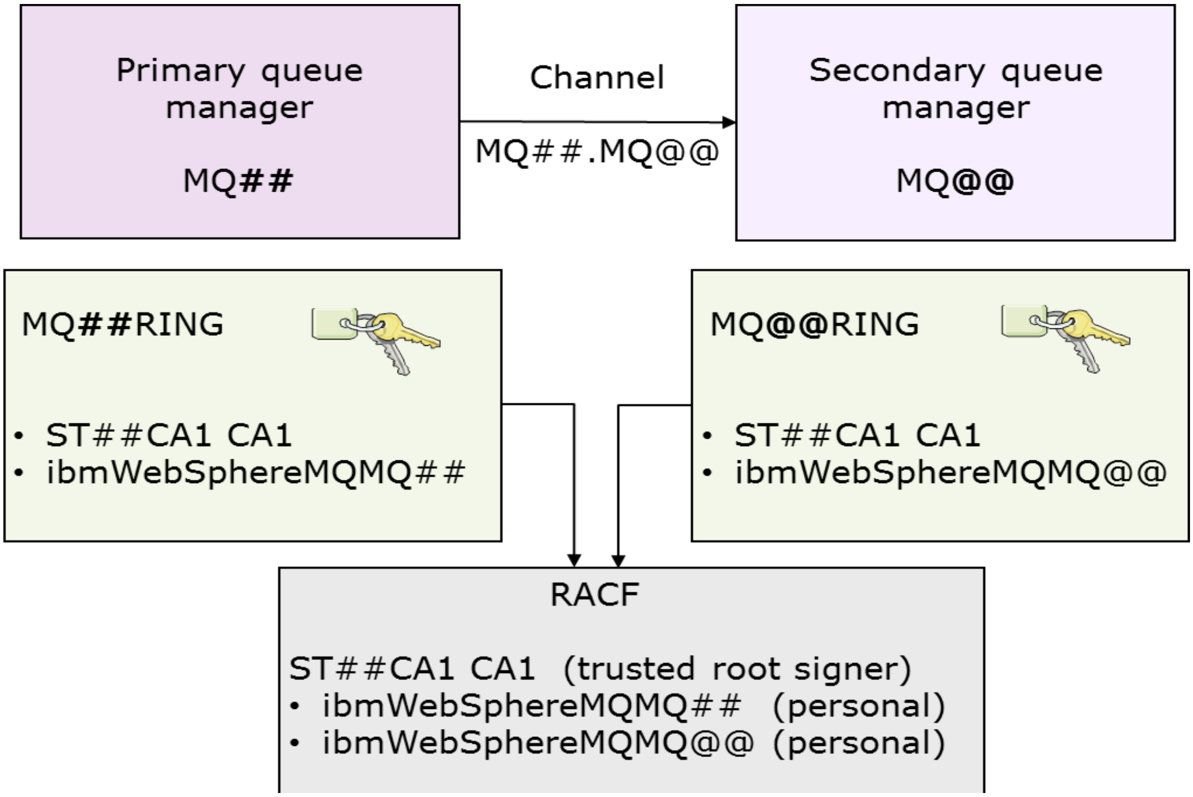


Figure 2-1. Scenario 1: Sender-receiver distributed channels

You work with two TSO sessions, one for each queue manager. You must ensure that you use the correct TSO ID as directed in the different sections of this exercise.

Section 1: Create signer CA for primary queue manager

Important

For these sections, **log on with your primary ID**.

A similar note is presented when it is time to switch to the secondary TSO ID.

__ 1. Proceed to the **`TSM00##.WM31.SCSQPROC`** data set and select to edit member **R111CRCA**.

__ a. Change all occurrences of the `##` placeholder to your primary student number.

```
RACDCERT CERTAUTH GENCERT -
SUBJECTSDN(CN('STUDENT ## CA1') -
T('IBM MQ') -
OU('TRAINING') -
O('IBM') -
L('HURSLEY') -
SP('HAMPSHIRE') -
C('UK')) -
WITHLABEL('ST##CA1') -
KEYUSAGE(CERTSIGN)
RACDCERT CERTAUTH LIST(LABEL('ST##CA1'))
```

__ b. Note the name of the job on the upper left view of the JCL as it should display on SDSF.

__ c. Submit the **R111CRCA** job.

__ d. Proceed to the SDSF panel by typing `=s;st` at the ISPF command input prompt and pressing the Enter (Ctrl) key.

__ e. Find your completed job, named `MQ##CRCA`, and select it by entering a question mark (?) to the left of the `MQ##CRCA` job name and pressing the Enter (Ctrl) key.

__ f. Find the output that is labeled `SYSTSPRT`, select it by entering an `S` to the left of the `SYSTSPRT` label, and press the Enter (Ctrl) key.

__ g. Your results should resemble the display in the information box. If your results are not successful, resolve any problems or request assistance if needed before proceeding.

__ h. Look for the READY prompt displays, followed by the command that is successfully completed, without any error messages.

__ i. R111CRCA JCL was coded to include a RACDCERT LIST following creation of the certificate. For this JCL, a detail listing of a successfully created certificate should follow the certificate creation confirmation.

__ j. Note the signer certificate with defaulted date ranges in the RACDCERT LIST display.

__ 2. Write down the valid dates of your signer certificate for reference before generating any queue manager certificates related to the ST##CA1 signer.

Valid from date: ____

Valid to date: ____

Information

The SDSF jobs have different names from the **TSM00##.WM31.SCSQPROC** JCL members to ensure that students can identify the jobs by their student or queue manager number. The expected output for job R111CRCA, output name `MQ##CRCA`, is shown. Results reflect student 01, queue manager MQ01, and TSO ID TSM0001. This explanation is not repeated.

```
READY
RACDCERT CERTAUTH GENCERT SUBJECTSDN(CN('STUDENT 01 CA1') T('IBM MQ')
OU('TRAINING') O('IBM') L('HURSLEY') SP('HAMPSHIRE') C('UK'))
WITHLABEL('ST01CA1') KEYUSAGE(CERTSIGN)
READY
RACDCERT CERTAUTH LIST(LABEL('ST01CA1'))

Digital certificate information for CERTAUTH:

  Label: ST01CA1
  Certificate ID: 2QiJmZmDhZmjgeLj8PHDwfFA
  Status: TRUST
  Start Date: 2015/04/21 00:00:00
  End Date:   2016/04/21 23:59:59
  Serial Number:
       >00<
  Issuer's Name:
       >CN=STUDENT 01 CA1.T=IBM MQ.OU=TRAINING.O=IBM.L=HURSLEY... ...
```

Section 2: Create the first certificate for the primary queue manager

__ 3. Return to the `TSM00##.WM31.SCSQPROC` data set and edit member **R112CRCR** to create the first queue manager certificate for the primary queue manager.

__ a. Change all instances of the `##` placeholder to your primary student number.

```
RACDCERT ID(TSM00##) GENCERT -
SUBJECTSDN(CN('MQ## PERSONAL') -
T('QMGR') -
OU('TRAINING') -
O('IBM') -
L('HURSLEY') -
SP('HAMPSHIRE') -
C('UK')) -
NOTBEFORE(DATE(20YY/MM/DD) TIME(00:00:00)) -
NOTAFTER(DATE(20YY/MM/DD) TIME(23:59:59)) -
WITHLABEL('ibmWebSphereMQMQ##') -
SIGNWITH(CERTAUTH LABEL('ST##CA1'))
RACDCERT LIST(LABEL('ibmWebSphereMQMQ##')) ID(TSM00##)
```

Hint

Always enclose your LABEL names and other parameter values that use mixed case in quotation marks. Ensure that your ISPF edit session is set to CAPS OFF if you need to work with mixed case.

__ b. If you did not write down your signer certificate date range information, it might be available in your SDSF job output `MQ##CRCA`. As an alternative, you can edit and run job R994DSCA to display your signer certificate.

__ c. Update the `MM/DD/20YY` placeholders in the `NOTBEFORE` and `NOTAFTER` parameters with dates that fall ***within*** or ***are equal to*** the signer CA date range. Be careful that the date and time stamps that are used in these dates work within a minute or so, as you use these certificates soon.

__ d. Note the name of the job as it should display in SDSF.

__ e. Submit `R112CRCR`.

__ 4. Find your completed job, named `MQ##CRCR`, and select it by entering a question mark (?) to the left of the `MQ##CRCR` job name and pressing the Enter (Ctrl) key.

__ 5. Find the output that is labeled `SYSTSPRT`, select it by entering an `S` to the left of the `SYSTSPRT` label, and press the Enter (Ctrl) key.

__ 6. Carefully check the results. An example of the output for a successful definition is shown. This output uses `MQ01` and TSO ID TSM0001 for illustrative purposes. The `RACDCERT` command portion of the output is truncated; the important parts to observe are:

__ a. The READY prompt displays preceding and following each `RACDCERT` command.

__ b. The absence of any warning or negative messages.

__ c. The RACDCERT list that is coded in the JCL follows the RACDCERT READY prompts for the GENCERT command, indicating that the certificate was created.

```
READY
RACDCERT ID(TSM0001) GENCERT SUBJECTSDN(CN('MQ01 PERSONAL') T('QMGR'... ...
NOTBEFORE(DATE(2015/04/22) TIME(00:00:00)) NOTAFTER(DATE(2016/04/20)... ...
... TH LABEL('ST01CA1'))
READY
RACDCERT LIST(LABEL('ibmWebSphereMQMQ01')) ID(TSM0001)

Digital certificate information for user TSM0001:

  Label: ibmWebSphereMQMQ01
  Certificate ID: 2Qfj4tTw8PDxiYKU5oWC4peIhZmF1NjU2PDx
  Status: TRUST
  Start Date: 2015/04/22 00:00:00
  End Date:   2016/04/20 23:59:59
```

STOP **Stop**

If there are date range errors and you see the message `The certificate is added with NOTRUST status`, you must stop and follow the process that is documented in section *Process to remove and replace a queue manager certificate* before proceeding.

Section 3: Create the primary queue manager certificate ring and connect the signer and personal certificates to the ring

__ 7. Select job **R113CRRG** for edit.

__ a. Change all occurrences of the ## placeholder to your primary student number.

```
RACDCERT ID(TSM00##) ADDRING(MQ##RING)
```

__ b. Note the JCL name MQ##CRRG and submit the JCL.

__ c. Select your job MQ##CRRG from SDSF and review the output within the READY prompt displays.

__ 8. If any errors are found, correct them or request assistance before proceeding. If the job ran successfully, the primary queue manager ring is now created, but empty. You now connect your signer and primary queue manager certificates to the ring.

__ 9. Select job **R114CNRG** for edit.

__ a. Change all occurrences of the ## placeholder to your primary student number.

```
RACDCERT ID(TSM00##) CONNECT(CERTAUTH LABEL('ST##CA1') -
RING(MQ##RING))
RACDCERT ID(TSM00##) CONNECT(ID(TSM00##) LABEL('ibmWebSphereMQMQ##') -
RING(MQ##RING))
RACDCERT LISTRING(*) ID(TSM00##)
```

__ b. Note the JCL name MQ##CNRG and submit the JCL.

__ c. Select your job MQ##CNRG from SDSF and review the output within the READY prompt displays.

After the certificates are connected to the ring, MQ##CNRG contains a RACDCERT command to list contents of the ring. An example of a successful ring display that uses MQ01 is shown:

```
Ring:
     >MQ01RING<
Certificate Label Name              Cert Owner     USAGE      DEFAULT
--------------------------------    ------------   --------   -------
ST01CA1                             CERTAUTH       CERTAUTH     NO

ibmWebSphereMQMQ01                  ID(TSM0001)    PERSONAL     NO
```

Section 4: Enable use of SSL/TLS in the primary queue manager

Important

If you alter the queue manager to set the SSLTASKS and SSLKEYR, the SCSQPROC(CSQ4INYG) member would override these attributes upon a queue manager restart. Update CSQ4INYG and restart the queue manager to preserve the settings and avoid unexpected messages later in the exercises. One such message you might encounter in later exercises if you fail to update CSQ4INYG is:

```
CSQX630E MQ04 CSQXRCTL Channel MQ##.MQ@@ requires SSL
```

You might also see a 2393 return in your Windows client.

__ 10. Proceed to your TSM00##.SCSQPROC and open member CSQ4INYG for edit.

__ 11. Locate field SSLTASKS and set it to 5.

__ 12. Locate field SSLKEYR and set it to MQ##RING where ## is your primary student number.

__ 13. Stop the queue manager by typing /MQ## STOP QMGR where ## is your primary student number, and press the Enter (Ctrl) key.

__ 14. Wait until you see the CSQ3100I SUBSYSTEM MQ## READY FOR START COMMAND for ***your*** queue manager.

__ 15. Start the queue manager by typing /MQ## START QMGR PARM(MQ##ZPRM) where ## is your primary student number, and press the Enter (Ctrl) key.

__ 16. As your MQ##CHIN task restarts, look for the CSQX080I and CSQX8081I messages. These messages should now show a number of SSL server subtasks and identification of the key ring for your primary queue manager. **The following segment uses MQ01 in the display for illustrative purposes**:

```
+CSQX080I MQ01 CSQXGIP SSLTASKS=5, SSLRKEYC=0
+CSQX081I MQ01 CSQXGIP SSLKEYR=MQ01RING
```

Important

There might be several students simultaneously restarting their `MQ##CHIN` task. If you check the log view instead of your `MQ##CHIN` started task, ensure the results that you check are for ***your*** `MQ##CHIN` task.

__ 17. Temporarily disable `CHLAUTH` after the queue manager restart as follows:

```
/MQ## ALTER QMGR CHLAUTH(DISABLED)
```

Note

You disable `CHLAUTH` to mitigate any possible issues with blocked channels in the SSL lab. You enable `CHLAUTH` later in Exercise 3. It is manually disabled because member `CSQ4INYG` enables `CHLAUTH` when you restart the queue manager.

Section 5: Change the selected channel to use SSL/TLS

Although you configured your primary queue manager to use SSL/TLS, all channels continue to run without use of SSL/TLS until the channels are individually configured to use SSL/TLS.

STOP **Stop**

If your `MQ##.MQ@@` sender channel failed to start in Exercise 1, resolve the problem with the channel before proceeding. Do not alter the channel to enable SSL/TLS or continue with any subsequent steps until you ensure that the `MQ##.MQ@@` channel runs.

__ 18. Return to your `TSM00##.WM31.SCSQPROC` library and select member `R117CHCL` for edit.

__ a. Review the contents of the `ALTER CHANNEL` command. Check the cipher specification that is preselected in the command. Leave the selected cipher as coded.

```
ALTER CHL(MQ##.MQ@@) CHLTYPE(SDR) +
      SSLCIPH(TLS_RSA_WITH_AES_128_CBC_SHA256)
```

__ b. Change all occurrences of the `##` placeholder to your primary student number.

__ c. Change all occurrences of the `@@` placeholder to the secondary student number.

__ d. Note the job name, `MQ##CHCH`.

__ 19. From the SDSF job list screen, select the output of job `MQ##CHCH` and scroll down to confirm results; do not depend solely on a zero condition code.

Important

You completed the work with your primary queue manager. You now complete the necessary work to implement SSL/TLS in the secondary queue manager.

To work with the secondary queue manager, log on to a TSO session by using your ***secondary*** TSO ID.

Section 6: Create the secondary queue manager certificate

The secondary queue manager certificate is signed with `ST##CA1`. It is not necessary to create a new signer certificate for Scenario 1. You continue the work by creating the secondary queue manager certificate.

__ 20. Return to the `TSM00@@.WM31.SCSQPROC` data set and edit member **R121CRC2** to create the first queue manager certificate for the primary queue manager. You use both your primary and secondary student numbers to update the `RACDCERT` command:

```
RACDCERT ID(TSM00@@) GENCERT -
SUBJECTSDN(CN('MQ@@ PERSONAL') -
T('QMGR') -
OU('TRAINING') -
O('IBM') -
L('HURSLEY') -
SP('HAMPSHIRE') -
C('UK')) -
NOTBEFORE(DATE(20YY/MM/DD) TIME(00:00:00)) -
NOTAFTER(DATE(20YY/MM/DD) TIME(23:59:59)) -
WITHLABEL('ibmWebSphereMQMQ@@') -
SIGNWITH(CERTAUTH LABEL('ST##CA1'))
RACDCERT LIST(LABEL('ibmWebSphereMQMQ@@')) ID(TSM00@@)
```

__ a. Change all instances of the `##` placeholder to your primary student number.

__ b. Change all instances of the `@@` placeholder to your secondary student number.

__ c. If you did not write down your signer certificate date range information, it might be available in your SDSF job output `MQ##CRCA`. As an alternative, you can edit and run job `MQ##DSCA` to display your signer certificate.

__ d. Update the `20YY/MM/DD` placeholders in the `NOTBEFORE` and `NOTAFTER` parameters with dates that fall within the signer CA date range.

__ e. Note the name of the job as it should display in SDSF, `MQ@@CRC2`.

__ 21. Submit `R112CRCR`.

__ a. Find your completed job, named `MQ@@CRC2`, and select it by entering a question mark (?) to the left of the `MQ@@CRC2` job name and pressing the Enter (Ctrl) key.

__ b. Find the output that is labeled `SYSTSPRT`, select it by entering an `S` to the left of the `SYSTSPRT` label, and press the Enter (Ctrl) key.

__ c. Carefully check the results that are shown between the two `READY` prompt displays.

Stop

If there are date range errors and you see message “The `certificate is added with NOTRUST status`”, you must stop and follow the process that is documented in section *Process to remove and replace a queue manager certificate* before proceeding.

Section 7: Create the secondary queue manager certificate ring and connect the signer and personal certificates to the ring

__ 22. Create the certificate ring for the secondary queue manager.

__ a. Open job `R122CRR2` for edit.

__ b. Change all `@@` placeholders to your secondary student number.

__ c. Submit and check the output.

```
RACDCERT ID(TSM00@@) ADDRING(MQ@@RING)
```

__ d. Make sure that you make any needed corrections before continuing.

Warning

Take special care with the `##` and `@@` placeholders when you make changes to the rest of the jobs. Many of the jobs require changes that use ***both*** the primary student number for the `##` placeholder, and the secondary student number for the `@@` placeholder.

__ 23. Connect your `MQ@@RING` to the `ST##CA1` signer and the `MQ@@` queue manager certificate.

__ a. Open job `R123CNR2` for edit.

```
RACDCERT ID(TSM00@@) CONNECT(CERTAUTH LABEL('ST##CA1') -
RING(MQ0@@ING))
RACDCERT ID(TSM00@@) CONNECT(ID(TSM00@@) LABEL('ibmWebSphereMQMQ@@') -
RING(MQ@@RING))
RACDCERT LISTRING(*) ID(TSM00@@)
```

__ b. Change all occurrences of ***both*** placeholders: `##` to your primary student number, and `@@` to your secondary student number.

__ c. Note the job name, `MQ@@CNR2`.

__ d. Submit the job and check the output. This job includes a listing of the certificate ring for queue manager `MQ@@`. There should be two certificates: the signer and the queue manager certificates. Results should resemble your `MQ##CNRG` job.

Section 8: Enable your secondary queue manager to use SSL/TLS

__ 24. Proceed to your TSM00@@.WM31.SCSQPROC and open member CSQ4INYG for edit.

__ 25. Locate field SSLTASKS and set it to 5.

__ 26. Locate field SSLKEYR and set it to MQ@@RING, where @@ is your secondary student number.

__ 27. Stop the queue manager by typing /MQ@@ STOP QMGR where @@ is your secondary student number, and press the Enter (Ctrl) key.

__ 28. Wait until you see the CSQ3100I SUBSYSTEM MQ@@ READY FOR START COMMAND message for ***your*** queue manager.

__ 29. Start the queue manager by typing /MQ@@ START QMGR PARM(MQ@@ZPRM) where @@ is your secondary student number, and press the Enter (Ctrl) key.

Note

Changes to certificates in the key ring do not become effective until either you type the command REFRESH SECURITY TYPE(SSL), or the channel initiator is restarted.

For Part 1 of this exercise, you do not need to type the command because you stop and start the queue manager.

__ 30. As your MQ@@CHIN task restarts, look for the CSQX080I and CSQX8081I messages. They should now show a number of SSL server subtasks and identification of the key ring for your primary queue manager. **For illustrative purposes, MQ21 is used in the display**:

```
+CSQX080I MQ21 CSQXGIP SSLTASKS=5, SSLRKEYC=0
+CSQX081I MQ21 CSQXGIP SSLKEYR=MQ21RING
```

Important

It might happen that several students restart their MQ@@CHIN task simultaneously. Ensure that the results that you check are for ***your*** MQ@@CHIN task.

Section 9: Change the selected channel to use SSL/TLS

__ 31. You should still be on the TSM00@@.WM31.SCSQPROC library. You need to change the receiver channel to use the same cipher as the sender. Open job R126CHC2 for edit.

__ a. Examine the JCL member. Check the cipher specification that is preselected in the command and compare it to the cipher specification of the sender channel in your MQ## queue manager. The ciphers for the sender and receiver should be the same.

__ b. If the ciphers agree, leave the selected cipher as coded. If the ciphers disagree, consult with the instructor.

__ c. You have multiple changes. Change all occurrences of `##` to your primary queue student number, and all occurrences of `@@` to your secondary student number.

```
ALTER CHL(MQ##.MQ@@) CHLTYPE(RCVR) +
   SSLCIPH(TLS_RSA_WITH_AES_128_CBC_SHA256)
```

__ d. Submit the job and check the results

__ e. Review the contents of the `ALTER CHANNEL` command output.

Section 10:Test the SSL/TLS setup

It is now time to check your SSL/TLS configuration.

Important

Start a TSO session with your primary ID. If you have the original session with your primary TSO ID open, ensure that you are logged on with the primary TSO ID.

__ 32. Proceed to the SDSF system log screen by typing `=s;log` in your ISPF command input prompt and pressing the Enter (Ctrl) key.

__ 33. Start your sender channel from your primary TSO ID session by typing the channel start command with the command prefix string. Replace the placeholders with your primary and secondary student numbers:

```
/MQ## START CHANNEL(MQ##.MQ@@)
```

__ 34. Your results should resemble the display. The `MQ01.MQ21` channel results are used for illustrative purposes:

```
MQ01 START CHL(MQ01.MQ21)
 CSQM134I MQ01 CSQMSCHL  START CHL(MQ01.MQ21) COMMAND ACCEPTED
 +CSQX500I MQ01 CSQXRCTL Channel MQ01.MQ21 started
 +CSQX500I MQ21 CSQXRESP Channel MQ01.MQ21 started
  connection 10.31.187.114
 CSQ9022I MQ01 CSQXCRPS ' START CHL' NORMAL COMPLETION
```

__ 35. If your channel did not start after the SSL/TLS configuration, review the error message and make necessary corrections.

Important

There might be several students simultaneously starting their `MQ##.MQ@@` channel. Ensure that the results that you check are for ***your*** `MQ##.MQ@@` channel.

Continue to use your primary TSO ID session.

__ 36. Display the channel status by typing the `DIS CHS` command ***with the parameters indicated***. Replace the placeholders with your primary and secondary student numbers.

```
/MQ## DIS CHS(MQ##.MQ@@) SSLCERTI SSLPEER
```

__ 37. Review your channel status information. Your results should be similar to the display. The `MQ01.MQ21` channel is used for illustrative purposes:

```
MQ01 DIS CHS(MQ01.MQ21) SSLCERTI SSLPEER
CSQM293I MQ01 CSQMDRTC 1 CHSTATUS FOUND MATCHING REQUEST CRITERIA
CSQM201I MQ01 CSQMDRTC  DIS CHSTATUS DETAILS 955
CHSTATUS(MQ01.MQ21)
CHLDISP(PRIVATE)
XMITQ(MQ21)
CONNAME(10.31.187.114)
CURRENT
CHLTYPE(SDR)
STATUS(RUNNING)
SUBSTATE(MQGET)
STOPREQ(NO)
RQMNAME(MQ21)
SSLCERTI(CN=STUDENT 01 CA1,T=IBM
MQ,OU=TRAINING,O=IBM,L=HURSLEY,ST=HAMPSHIRE,C=UK)
SSLPEER(SERIALNUMBER=04,CN=MQ21
PERSONAL,T=QMGR,OU=TRAINING,O=IBM,L=HURSLEY,ST=HAMPSHIRE,C=UK)
```

__ a. The status should reflect that the channel is running.

__ b. The `SSLCERTI` and `SSLPEER` values display the subject and issuer distinguished names.

__ 38. Put a few messages from `MQ##` to `MQ@@`.

__ a. Edit JCL member `R131MQPT`. This job runs a sample MQPUT program.

__ b. Change all occurrences of the `##` placeholder to your primary student number.

__ c. Change all occurrences of the `@@` placeholder to your secondary student number.

```
// PARM=('MQ## TO.MQ@@ 0003 S 0100 P')
```

__ d. The second parameter should contain the name of the remote queue you created in Exercise 1.

__ e. Note the job name, `MQ##MQPT`, and submit the job.

__ f. Check the return code.

Important

Log on with your secondary TSO ID: `TSM00@@`

__ 39. Check that the messages arrived in the `FROM.MQ##` local queue in the secondary queue manager. You can use message handler utility at ISPF option H.

__ a. Type `=H` at the ISPF command input prompt and press the Enter (Ctrl) key.

__ b. Type your secondary queue manager name `MQ@@` at the first entry line, replacing the placeholder with your secondary student number.

__ c. Type the name of the local queue `FROM.MQ##` in the queue name field. Substitute your primary student number for the `##` placeholder.

__ d. Press the Enter (Ctrl) key.

- If you did not clear the `FROM.MQ##` queue in Exercise 1, you should have six messages: three from Exercise 1 and three from this test.
- If you cleared your messages after Exercise 1, you should see three messages.

This step concludes the first SSL/TLS Scenario 1 work for Exercise 2.

Scenario 2 exercise instructions

The second SSL/TLS configuration scenario is illustrated in Figure 2-2. In this scenario, you work with different concepts in the same exercise:

- The use of multiple CA certificates in the same queue manager.
- Implementing SSL from an IBM MQ client to an IBM MQ server.
- Using nonstandard certificate label names in the IBM MQ client and queue manager.
- The queue manager and the IBM MQ client use CA certificates signed by a second CA in your primary queue manager, ST##CA2, where ## stands for your primary student number.
- Scenario 2 corresponds to section *Using CA-signed certificates for mutual authentication of a client and queue manager* in the IBM Knowledge Center.

TSM00@@, the secondary queue manager, is not used in this part of the exercise.

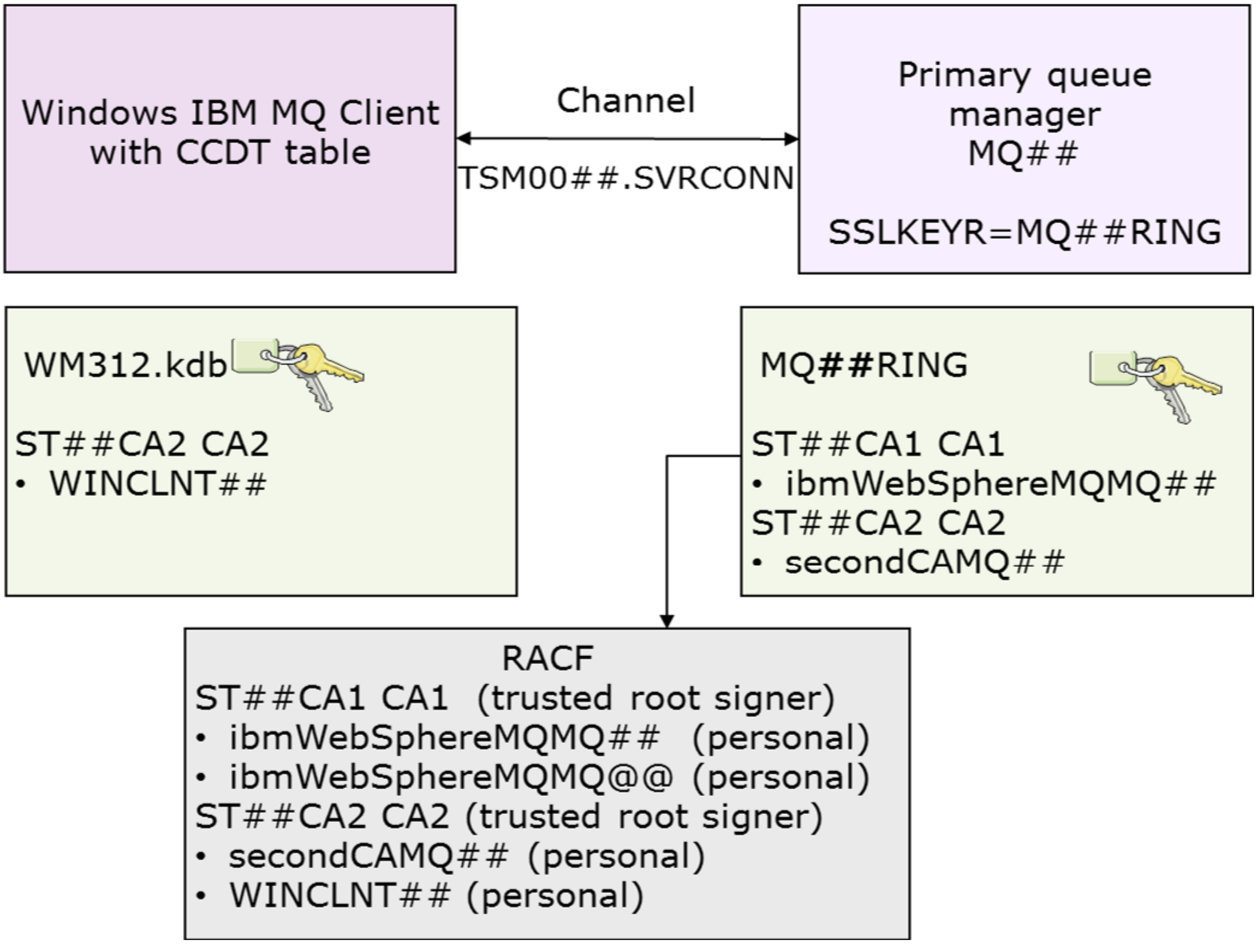

Windows IBM MQ Client with CCDT table
Channel
TSM00##.SVRCONN
Primary queue manager MQ##
SSLKEYR=MQ##RING
WM312.kdb
ST##CA2 CA2
• WINCLNT##
MQ##RING
ST##CA1 CA1
• ibmWebSphereMQMQ##
ST##CA2 CA2
• secondCAMQ##
RACF
ST##CA1 CA1 (trusted root signer)
• ibmWebSphereMQMQ## (personal)
• ibmWebSphereMQMQ@@ (personal)
ST##CA2 CA2 (trusted root signer)
• secondCAMQ## (personal)
• WINCLNT## (personal)

Figure 2-2. Scenario 2 IBM MQ client to queue manager SSL/TLS configuration WM3121.1

In this scenario, you add ST##CA2 as a new CA. You create a new certificate for MQ## named secondCAMQ## and a certificate for the IBM MQ client, WINCLNT##, both signed by ST##CA2.

Stop

If your amqsputc client connectivity test to `TSM00##.SVRCONN` failed to work in Exercise 1, ***you must resolve the problem before you continue with the work in this section.*** Do not proceed with any other steps until you ensure that the amqsputc to your `PRICE.CL` queue works. There should be messages in your `PRICE.CL` queue, and you see the `CSQX511I` in the z/OS log for your client channel, for example:

```
CSQX511I MQ## CSQXRESP Channel TSM00##.SVRCONN started
```

This section of the exercise depends on the client channel successfully working. Even if your client connectivity work was successful in Exercise 1, ***repeat the test now:***

- Go to your VMware workstation and open a command window.
- Type `amqsputc PRICE.CL` and press the Enter key.
- Go to the system log of your primary TSO session and look for the `CSQX511I` message for your TSM00##.SVRCONN channel.
- The `CSQX511I` results should indicate that the channel started in your z/OS queue manager.

Section 1: Create the second CA signer certificate ST##CA2

Important

Log on to a z/OS session with your primary TSO ID. In the next sections, you work with your primary queue manager.

__ 40. Proceed to **TSM00##.WM31.SCSQPROC** and edit member **R211CRCA**. You are creating a second signer to be used by the primary queue manager and client.

__ a. Change all occurrences of the `##` placeholder to your primary student number.

```
RACDCERT CERTAUTH GENCERT -
SUBJECTSDN(CN('STUDENT ## CA2') -
T('IBM MQ') -
OU('TRAINING') -
O('IBM') -
L('HURSLEY') -
SP('HAMPSHIRE') -
C('UK')) -
WITHLABEL('ST##CA2') -
KEYUSAGE(CERTSIGN)
RACDCERT CERTAUTH LIST(LABEL('ST##CA2'))
```

__ b. Change all occurrences of the `##` placeholder to your primary student number.

__ c. Note the job name and submit the job.

__ d. Check the results by going to SDSF, selecting the output for your job, and finding the READY prompt displays.

__ 41. Write down the valid dates of your signer certificate for reference before generating any queue manager certificates related to your ST##CA2 signer.

- Valid from date: ____
- Valid to date: ____

Section 2: Create the MQ## and IBM MQ client certificates that are signed by ST##CA2

__ 42. Select job **R212CRCS** for edit.

__ a. Change all occurrences of the ## placeholder to your primary student number

__ b. Replace the placeholders in the NOTBEFORE and NOTAFTER parameters with dates that fall within the ST##CA2 signer certificate valid dates.

```
RACDCERT ID(TSM00##) GENCERT -
SUBJECTSDN(CN('MQ## PERSONAL CA2') -
T('QMGR') -
OU('TRAINING') -
O('IBM') -
L('HURSLEY') -
SP('HAMPSHIRE') -
C('UK')) -
NOTBEFORE(DATE(20YY/MM/DD) TIME(00:00:00)) -
NOTAFTER(DATE(20YY/MM/DD) TIME(23:59:59)) -
WITHLABEL('secondCAMQ##') -
SIGNWITH(CERTAUTH LABEL('ST##CA2'))
RACDCERT LIST(LABEL('secondCAMQ##')) ID(TSM00##)
```

__ c. Review the definition before submitting. Note how the label departs from the queue manager label standards. **Remember this label name, secondCAMQ##, for the CERTLABL parameter of your server connection channel.**

__ d. Submit the job and check results. Be careful that the certificate was created without date range issues.

__ 43. Select job **R213CRCW** for edit. This job creates the IBM MQ client certificate that is signed by ST##CA2.

__ a. Change all occurrences of the ## placeholder to your primary student number.

```
RACDCERT ID(TSM00##) GENCERT -
SUBJECTSDN(CN('MQ## WIN CA2 PERS') -
T('CLIENT') -
OU('TRAINING') -
O('IBM') -
L('ORLANDO') -
SP('FLORIDA') -
C('US')) -
NOTBEFORE(DATE(20YY/MM/DD) TIME(00:00:00)) -
NOTAFTER(DATE(20YY/MM/DD) TIME(23:59:59)) -
WITHLABEL('WINCLNT##') -
SIGNWITH(CERTAUTH LABEL('ST##CA2'))
RACDCERT LIST(LABEL('WINCLNT##')) ID(TSM00##)
```

__ b. Replace the placeholders in the NOTBEFORE and NOTAFTER parameters with dates that fall within the ST##CA2 signer certificate valid dates.

__ c. Review the definition before submitting. Note how the label departs from the IBM MQ client label standards. **Remember this label name, WINCLNT##, for the CERTLABL parameter of your client connection channel.**

__ d. Submit the job and check results. Be careful that the certificate was created without date range issues.

Section 3: Add the new queue manager certificate to the MQ## ring

__ 44. Open job **R214CNC2** for edit.

__ a. Replace the ## placeholder with your primary student number. The Windows client certificate is ***not*** added to the ring.

```
RACDCERT ID(TSM00##) CONNECT(CERTAUTH LABEL('ST##CA2') -
RING(MQ##RING))
RACDCERT ID(TSM00##) CONNECT(ID(TSM00##) LABEL('secondCAMQ##') -
RING(MQ##RING))
```

__ b. Submit the job and check results. If necessary, make any corrections before continuing.

Section 4: Refresh SSL security on the primary queue manager

__ 45. Proceed to the system log by typing =s;log and pressing the Enter (Ctrl) key.

__ 46. Type /MQ## REFRESH SECURITY TYPE(SSL) where ## is your primary student number, and press the Enter (Ctrl) key.

Section 5: Export the IBM MQ client certificates and corresponding signer for download to the Windows server

You now export the ST##CA2 signer certificate and the IBM MQ Windows client certificates to corresponding files in preparation to transfer the certificates to the Windows server. The signer certificate is also referred to as the “CA certificate”. This certificate is stored in DER format.

__ 47. Open job **R216EXCA** for edit.

__ a. Change all occurrences of the ## placeholder to your primary student number.

```
RACDCERT CERTAUTH EXPORT ( -
LABEL('ST##CA2')) -
DSN('TSM00##.DIGCERT.SIGNER.DER') -
FORMAT(CERTDER)
```

__ b. Write the name of the data set where the RACDCERT EXPORT sends the signer certificate. You need this name to download the certificate to the Windows client.

____

__ c. Submit the job and check results. Make any corrections, if needed, before proceeding.

__ 48. Export the IBM MQ client certificate. Open job **R217EXWC** for edit.

__ a. Review the RACDCERT command. This job exports the IBM MQ client private key, which requires that the exported certificate be protected with a password, and that it is in PKCS12 format.

```
RACDCERT ID(TSM00##) -
EXPORT (LABEL('WINCLNT##')) -
DSN('TSM00##.DIGCERT.WIN.PKCS12') -
FORMAT(PKCS12DER) -
PASSWORD('ZOSPWD')
```

__ b. Change all occurrences of the ## placeholder to your primary student number.

__ c. Write the name of the data set where the RACDCERT export sends the IBM MQ client certificate. You might also want to save the password.

__ d. Submit the job and check results. If corrections are needed, make them before proceeding.

Section 6: Make required changes to the server connection channel

Note

When you changed the sender channel for Scenario 1, you used the standard IBM MQ certificate naming convention, `ibmWebSphereMQMQ##`. In this scenario, you add a second certificate for the same queue manager and must use the `CERTLABL` attribute when you configure `SSL/TLS` for the server connection channel. The quotations around the label name are critical to preserve the correct case in the name. It is a good practice to confirm that changes were done as expected by displaying the channel details.

__ 49. Open member **`R218CHSV`**.

__ a. Replace all occurrences of the `##` placeholder with your primary student number.

__ b. Ensure that the value in the `CERTLABL` parameter is enclosed in single quotation marks.

```
ALTER CHL(TSM00##.SVRCONN) CHLTYPE(SVRCONN) +
     CERTLABL('secondCAMQ##') +
     SSLCIPH(TLS_RSA_WITH_AES_128_CBC_SHA256)
```

__ c. Submit the job and check results. If necessary, make any needed corrections.

__ 50. Open an SDSF log session by typing `=s;log` in a command input prompt and pressing the Enter (Ctrl) key.

__ 51. Type the command to display the `TSM00##.SVRCONN` channel and include the attributes that you changed in the `R218CHSV` job as shown in the display. `MQ01` is used for illustrative purposes:

```
/MQ01 DIS CHL(TSM0001*) SSLCIPH CERTLABL

CSQM293I MQ01 CSQMDRTC 1 CHANNEL FOUND MATCHING REQUEST CRITERIA
CSQM201I MQ01 CSQMDRTC  DIS CHANNEL DETAILS
CHANNEL(TSM0001.SVRCONN)
CHLTYPE(SVRCONN)
QSGDISP(QMGR)
SSLCIPH(TLS_RSA_WITH_AES_128_CBC_SHA256)
CERTLABL(secondCAMQ01)
 END CHANNEL DETAILS
```

Section 7: Transfer the IBM MQ client CA signer and personal certificates to the Windows VMware server

Important

FTP is disabled in the z/OS image. To download the certificates that RACF creates to the Windows server, you **work with the IBM Personal Communications 3270 emulator (PCOM).** You also use your Windows VMware image.

Use the PCOM panel with your primary TSO ID to download the certificates.

Under some circumstances, the PCOM section might close while you are attempting the transfer. If your session closes, use your second TSO ID to cancel your primary TSO ID. To cancel your primary TSO ID from your secondary TSO session:

- If your secondary TSO session is not available, log on to the primary LPAR with your secondary TSO ID.
- Proceed to the SDSF log panel by typing `=s;log` in the command input prompt and pressing the Enter (Ctrl) key.
- Type `/C U=TSM00##` where `##` is your primary student number, and press Enter.
- Check that the command was accepted, and log on to a new session with your primary TSO ID.

__ 52. Proceed to your Windows VMware image.

__ a. Open a command window by selecting the command prompt icon.

__ b. Move to your `C:\` directory by typing `C:\` and pressing the Enter key.

__ c. You should see an empty directory called `certs`. If the directory exists, skip the step to create the directory.

__ d. If the directory does not exist, create the directory to receive your certificates. From the `C:\` directory, type `mkdir certs` and press the Enter key.

__ e. **Ensure that you have a `C:\certs` directory, and that the `C:\certs` directory is empty**.

Important

For the transfer to work, **you must be in the ISPF Command Shell panel *before* you start the process.**

Go to the ISPF Command Shell panel now by entering =6 in the command input prompt and pressing the Enter (Ctrl) key. Your screen should resemble the display.

```
            ISPF Command Shell
    Enter TSO or Workstation commands below:

    ===> _____________________________________
   ___________________________________________
```

__ 53. Configure your PCOM session to download the files to the `C:\certs` directory.

__ a. Select the primary TSO ID PCOM session 3270 emulator.

__ b. From the menu across the top, select **Edit > Preferences > Transfer**.

__ c. In the General tab, select **Browse** to the right of the **File Transfer Selection** label.

__ d. Locate and select the `C:\certs` directory and click **OK**.

__ e. Confirm that the File Transfer Selection box now shows `C:\certs`.

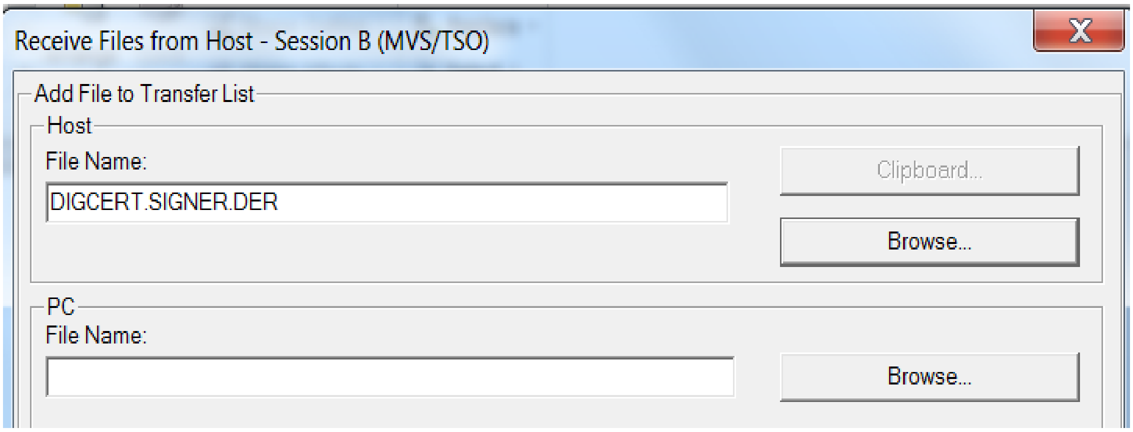
__ 54. Click **Apply**, then **OK** to exit the preferences box.

__ 55. From the PCOM menu, select **Actions > Receive File From Host**. The corresponding Receive File From Host panel appears.

__ 56. Select the **Browse** to the right of the Host section. **Wait a few moments until the data set list display is complete**.

__ 57. Locate the data set named `TSM00##.DIGCERT.SIGNER.DER`. If you are unable to see the full data set name, the file should be the **first** of the two data sets listed with names that start with: `TSM00##.DIGCERT`

Double-click the data set to select.



Receive Files from Host - Session B (MVS/TSO)

Add File to Transfer List

Host

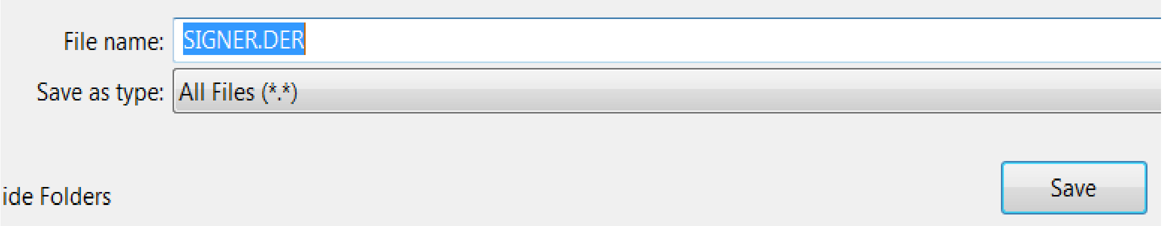
File Name: DIGCERT.SIGNER.DER

PC

File Name:

__ 58. Review the file name that was populated in the panel. The first-level qualifier, `TSM00##`, is dropped from the name, and you should see `DIGCERT.SIGNER.DER` as shown in the graphic. If your result is not the same, make the necessary correction before you proceed.

__ 59. Proceed to **Browse** to the right of the `PC File Name` box. Select the Windows `C:\certs` directory. The file name is prepopulated with `SIGNER.DER`. Select **Save**.

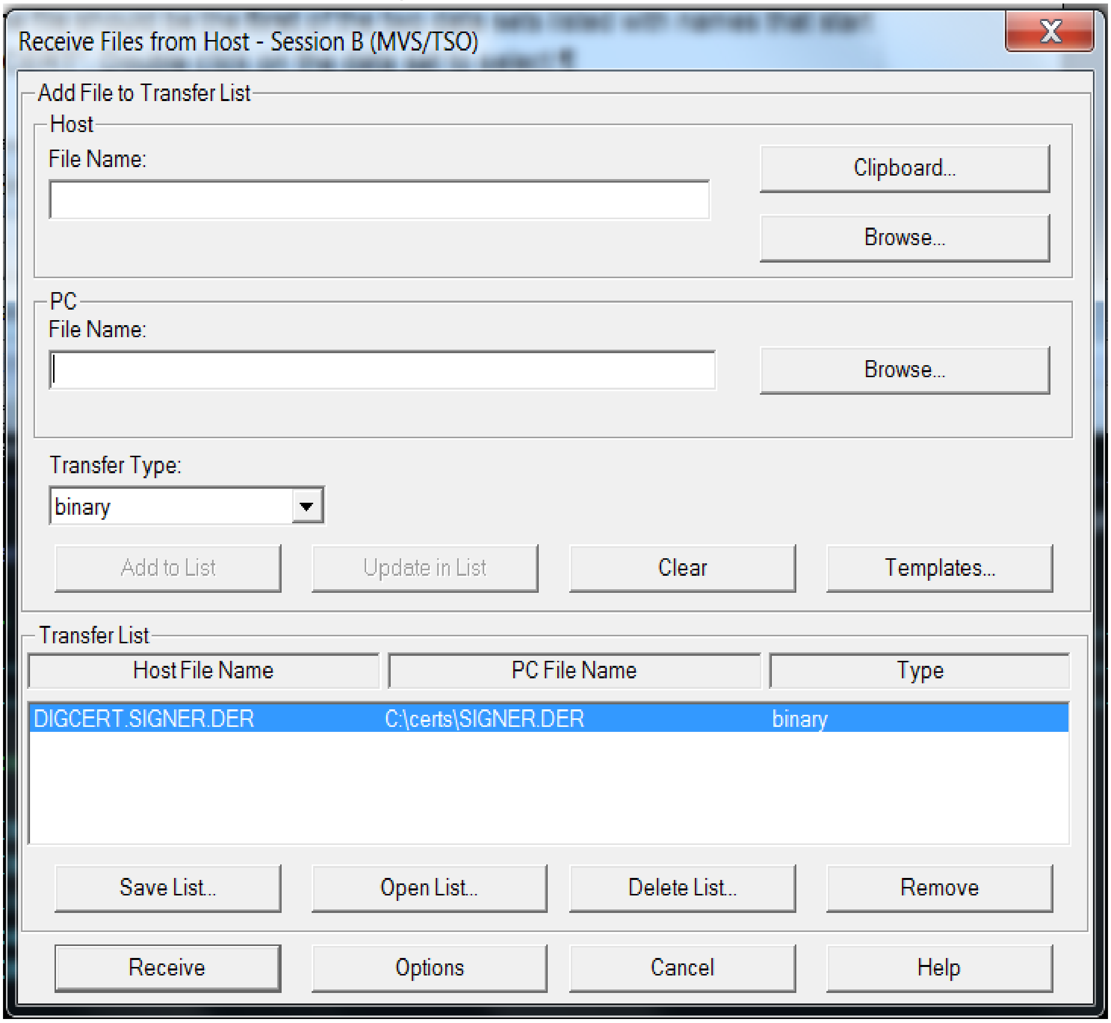


File name: SIGNER.DER

Save as type: All Files (*.*)

Figure 2-3.

__ **60. Carefully check the resulting PCOM Receive Files From Host panel:**



Receive Files from Host - Session B (MVS/TSO)

Add File to Transfer List

Host

File Name:

PC

File Name:

Transfer Type: binary

Transfer List

Host File Name	PC File Name	Type
DIGCERT.SIGNER.DER	C:\certs\SIGNER.DER	binary

Figure 2-4.

Warning

You might get two different results, which depend on information that might get saved in the PCOM session.

Proceed according to the result that you see in the PCOM panel after you complete your PC file destination to `C:\certs`.

__ 61. **Do this step if the Host and PC file names are still in the corresponding boxes and no transfers show in the Transfer List box; continue with the lettered substeps in this section. If your host and PC file selections show in the Transfer List, *proceed to the next numbered step* and skip these lettered substeps.**

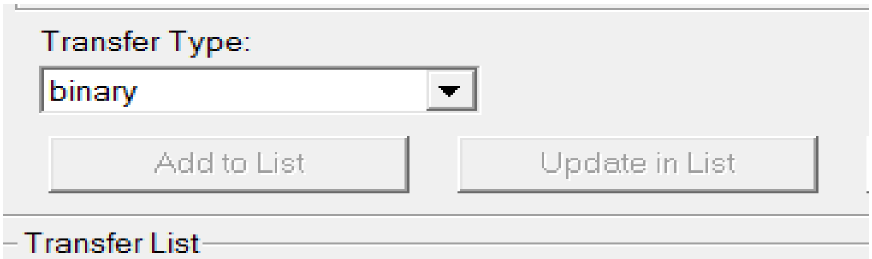
__ a. Confirm that the `Host` and `PC` file names are as expected.

__ b. The first level of the Host file name is truncated, for example: `DIGCERT.SIGNER.DER`

__ c. The PC name shows the corresponding name, for example: `c:\certs\SIGNER.DER`

__ d. Ensure that the Transfer Type is binary. If the Transfer Type is not binary, correct it to binary with the pull-down menu.

__ e. In this situation where the Transfer List box is empty, the “Add to List” button directly below the Transfer Type pull-down list is enabled. Click **Add to List**.



Transfer Type: binary

Figure 2-5.

__ 62. If the Host and PC names are in the **Transfer List** box, ensure that the **Transfer Type** is binary. **If the Transfer Type is not binary, follow the lettered substeps. If the Transfer Type is binary, proceed to the next numbered step.**

__ a. Select the transfer type that is listed in the Transfer List by placing the cursor over the listed transfer and clicking it one time.

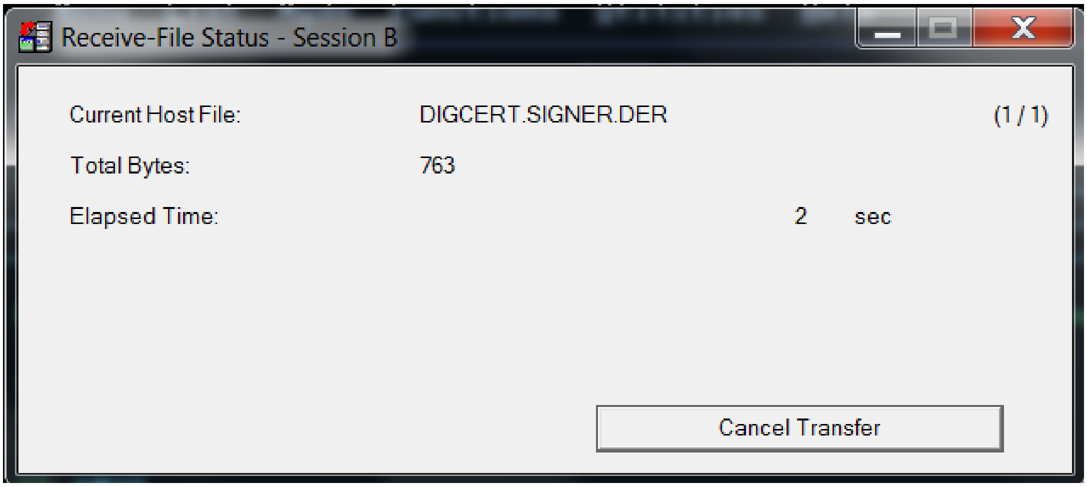
__ b. Click the Transfer Type pull-down menu and select **binary**. The **Update in List** button to the right below the Transfer Type menu is now enabled.

__ c. Select the **Update in List** button, and the Transfer Type in the Transfer List should be updated to binary.

Note

Before you continue, your PCOM “Receive Files from Host” transfer panel should look like the panel that is shown on the graphic.

__ 63. You now run the transfer. Select the **Receive** button in the left bottom of the “Receive Files from Host” panel and click it. A pop-up panel should surface, indicating the progress of the transfer. This process completes fast, so it is easy to miss the pop-up panel. For example:



Receive-File Status - Session B

Current Host File: DIGCERT.SIGNER.DER (1 / 1)

Total Bytes: 763

Elapsed Time: 2 sec

Figure 2-6.

If you do not see any errors in the pop-up panel, the transfer was completed successfully.

__ 64. Return to your VMware image. Check the contents of the `C:\certs` directory by using either Windows Explorer or the Command window. You should see the `SIGNER.DER` file in the `C:\certs` directory.

__ 65. Return to the ISPF Command Shell panel of your primary TSO ID session.

__ 66. Open a new PCOM “Receive Files from Host” panel. From the PCOM menu, select **Actions > Receive File From Host**.

__ 67. Repeat the process that you followed to download the signer certificate. Make the indicated changes:

__ a. Click **Browse** to locate the `Host File` name. **Wait a few seconds until the file name list is populated.** Select your `TSM00##.DIGCERT.WIN.PKCS12` file. If you are unable to see the full file name in the PCOM box, the file that contains the private key is the second file with the `TSM00##.DIGCERT` prefix.

The Host File Name field in the PCOM panel should now show as `DIGCERT.WIN.PKCS12`.

__ b. Click **Browse** to locate the PC file name. If you are not automatically in the `C:\certs` directory, locate and select the `C:\certs` directory. The prefilled file name in the **Save** button of the panel should show as `WIN.PKCS12`. Click **Save**.

The PC File field name in the PCOM panel should now show as `C:\certs\WIN.PKCS12`.

__ 68. Continue the same process that you used to download the signer file. When you are finished, the `C:\certs` directory should contain two files with the exact names shown: `SIGNER.DER` and `WIN.PKCS12`. If these files are in the `C:\certs` directory, you completed the download. If these files are not in your directory, you must take necessary steps to download the files correctly before you can continue with this section of the exercise.

Section 8: Create the key database, add signer and personal certificates, and display the key database contents

Important

The work in the next set of sections is completed in your VMware Windows server.

- Keep your TSO session open.
- Proceed to your VMware workstation and open a Windows command prompt.

To avoid typographical errors, you can edit, then copy and paste the commands from the course file templates that are found at `C:\LabFiles`. File names are specified in corresponding steps.

Do not copy the commands from the Student Exercise book because they might have special characters; copy the commands that are found in the `C:\LabFiles` directory.

The next set of instructions direct you to edit the command templates; then, copy and paste the commands to avoid typographical errors.

Note

For the purposes of the lab environment for this course, the certificates and the key database are in the same directory. This configuration should not be construed as a recommendation to use the same setup. It is not necessary to keep the key database and the certificates in the same directory.

Key databases and the directories where the key databases and certificates are stored should be adequately secured with limited access. Your organization should provide the proper guidance according to their specific security requirements.

__ 69. If you do not have a command prompt already open in your Windows VMware session, proceed to open a command window.

__ 70. **Change to the `C:\certs` directory by typing `cd C:\certs` and pressing the Enter key.**

__ 71. Edit template `5crtKdb.txt` by typing `notepad C:\LabFiles\5crtKdb.txt` and pressing the Enter key. The template file with the command to create the key database should open.

__ 72. This file does not need any changes. Review the contents of the file:

```
runmqakm -keydb -create -db wm312.kdb -pw passw0rd  -stash
```

__ 73. **Ensure that you are at the `C:/certs` directory.**

__ 74. Copy the contents of `5crtKdb.txt`, paste them in your command prompt, and press the Enter key.

__ 75. Check the results for any error messages.

__ 76. Display the `C:\certs` directory by typing `dir` and pressing the Enter key. You should see a series of `wm312`-named files, including the `wm312.kdb` file. You also see your certificate files.

__ 77. Open template `6addSigner.txt` by typing: `notepad C:\LabFiles\6addSigner.txt`

Important

The next set of steps use an abbreviated version of the command to identify the command to run. For example, if you see `-cert -list -all`, it means that you run the entire `runmqakm` command that contains the specified abbreviation. In the case of the example in this text box, it means to run

```
runmqakm -cert -list all -db C:\certs\wm312.kdb -pw passw0rd -type cms
```

__ 78. There are two command examples in the file. You select the first command for the next step.

__ 79. List the empty key database to confirm the password. Copy the line that contains the `-cert -list -all` command. It does not require any changes:

```
runmqakm -cert -list all -db C:\certs\wm312.kdb -pw passw0rd -type cms
```

__ 80. Paste the `-cert -list -all` command in the command window and press the Enter key. The response should indicate that no certificates were found.

__ 81. From the same `6addSigner.txt` template, customize the `-cert -add` command by substituting the `-label ST##CA2` so that the `##` placeholder is your primary student number.

Important

Ensure that you change the `##` placeholder in the script to your primary student number before you copy and paste it.

```
runmqakm -cert -add -db C:\certs\wm312.kdb -pw passw0rd -label ST##CA2
-format binary -trust enable -file C:\certs\SIGNER.der
```

__ 82. Paste the `-cert -add -db` command in the command prompt and press the Enter key. Ensure that there are no error messages.

__ 83. List the key database again. Instead of copying the command again, you can return the `-cert -list -all` command back into the command prompt by using your up arrow and scrolling through the commands until you find the `-cert -list -all` command. You should see your signer certificate.

__ 84. Open template `7importPers.txt` by typing: `notepad C:\LabFiles\7importPers.txt`

Review the contents. No changes are needed.

```
runmqakm -cert -import -file C:\certs\WIN.PKCS12 -pw ZOSPWD -type pkcs12
-target C:\certs\wm312.kdb -target_pw passw0rd -target_type cms
```

__ 85. Copy and paste the `-cert -import -file` command and press the Enter key.

__ 86. Retrieve the `-cert -list -all` command by using the up arrow, and press the Enter key to see the contents of the key database. The display should show both signer and personal certificates. Your results should resemble the display that uses student ID 01. ***Do not worry about the signer name; see the following notes about the appearance of the display.***

Stop

The trusted signer certificate might be displayed with its full distinguished name instead of the label. It might now show as ST##CA2, but preceded by the `!` sign to indicate trusted.

```
5724-H72 (C) Copyright IBM Corp. 1994, 2014.
Certificates found
* default, - personal, ! trusted, # secret key
!       ST01CA2
-       WINCLNT01
```

If you see a certificate that begins with an exclamation point (!) at the start of the display name, do not worry if the label name appears to change.

__ 87. If you see a similar display, you completed the certificate work. You now work with the IBM MQ client. If you do not see both certificates in the key database, make any needed corrections before you continue to the next section.

Section 9: Change the client connection channel to enable SSL/TLS

__ 88. From the `C:\LabFiles` directory, open file `8altCltChl.mqsc` for edit. You can use Notepad.

__ 89. Carefully review the changes to your client channel, including the label name.

```
ALTER CHL(TSM00##.SVRCONN) chltype(CLNTCONN) +
  CERTLABL(WINCLNT##) +
   SSLCIPH(TLS_RSA_WITH_AES_128_CBC_SHA256)
```

__ 90. Change both occurrences of the `##` placeholder to your primary student number, and save the file .

__ 91. If you are not in the `LabFiles` directory, change to that directory in your command prompt window by using the `cd` command.

__ 92. Enter the command `runmqsc -n < 8altCltChl.mqsc` and press the Enter key.

__ 93. Display the channel to confirm that the change was done correctly. From the command prompt, type `runmqsc -n` and press the Enter key.

__ 94. Type the `dis chl` command that is shown in this step. Replace the `##` placeholder with your primary student number.

```
dis chl(TSM00##.SVRCONN) CERTLABL SSLCIPH
```

__ 95. Confirm that the label and cipher specification changed correctly in the channel. Your results should resemble the display:

```
dis chl(TSM0005.SVRCONN) CERTLABL SSLCIPH
     2 : dis chl(TSM0005.SVRCONN) CERTLABL SSLCIPH
AMQ8414: Display Channel details.
   CHANNEL(TSM0005.SVRCONN)                CHLTYPE(CLNTCONN)
   CERTLABL(WINCLNT01)
   SSLCIPH(TLS_RSA_WITH_AES_128_CBC_SHA)
end
```

__ 96. Type `exit` to end the command script shell.

Section 10:Inform the IBM MQ client of the location of the key database

For purposes of this exercise, you use the IBM MQ environment option to indicate the location of the key database.

__ 97. From the `C:\LabFiles` directory, open member `10setMQSSLKEYR.txt`. This template does not require any changes. **The command must be entered *exactly* as shown, *without* the file suffix**.

```
set MQSSLKEYR=C:\certs\wm312
```

__ 98. Copy and paste the command in your Windows command prompt screen. Press the Enter key.

__ 99. Confirm that the `MQSSLKEYR` variable was set correctly. Type `echo %MQSSLKEYR%` and press the Enter key. The results should display:

```
C:\certs\wm312
```

Warning

If your `MQSSLKEYR` variable is not set correctly, the IBM MQ client cannot find the key database and the connectivity test fails with a **2381** return code.

It is assumed that the IBM MQ client test configuration was successfully tested without SSL/TLS in Exercise 1 before reaching this section of the work.

Section 11:Test the IBM MQ client to queue manager SSL/TLS setup

Note

In this step, you use the `amqsputc` sample client to test the SSL/TLS configuration for the MQI channels. Since an MQI channel ends when the application closes, ***you need to wait before you end the*** ***`amqsputc`*** ***session*** to capture the channel status on the z/OS console. Ensure that you obtain the channel status after you initiate the `amqsputc` session, but before you put the messages.

Do not close the ***`amqsputc`*** ***session by inadvertently pressing the Enter key***.

__ 100.Log on with your primary TSO ID and open an SDSF session by typing `=s;log` in the ISPF input prompt and pressing the Enter (Ctrl) key.

__ 101.Return to your VMware Windows command screen.

Note

After your client channel connects, you need to display the channel status from the z/OS side. To ensure that the channel continues to run, you keep the `amqsputc` program open. Take care to press Enter only one time after you type the `amqsputc` command.

__ 102.***Without pressing the Enter key***, type: `amqsputc PRICE.CL`

__ a. Press the Enter key one time after typing the command.

__ b. ***Do not type any text and <u>do not press the Enter key again until explicitly instructed</u>***. The response should be:

```
C:\certs>amqsputc PRICE.CL
Sample AMQSPUT0 start
target queue is PRICE.CL
```

__ c. **If your response was an error code**: enter the `mqrc` command in the Windows prompt, followed by the error code, and resolve any problems before you continue; for example: `mqrc 2381`

__ d. **If you accidentally pressed Enter more than one time and exited the amqsputc program**: pull up your previous `amqsputc PRICE.CL` command by using the up arrow. This time, be careful to press Enter one time after pulling up your command, so it resembles the expected display. You should be at prompt: `target queue is PRICE.CL`

__ e. Return to the z/OS log and find your channel. You can type `F TSM00##` in the command input prompt and press the Enter (Ctrl) key. The response should be similar to the display:

```
+CSQX511I MQ## CSQXRESP Channel TSM00##.SVRCONN started
 connection 10.61.23.250
```

Troubleshooting

If you receive a 2393 error code on the Windows server, check the corresponding message in the primary z/OS LPAR. If you see the message similar to the one in the display, which uses MQ04 as an example, restart your channel initiator and it should clear up:

```
+CSQX645E MQ04 CSQXRESP Certificate secondCAMQ04 missing for channel
TSM0004.SVRCONN
```

However, if you typed the `REFRESH SECURITY TYPE(SSL)` command after the last update of the z/OS ring, the `CSQX645E` message should not be displayed.

__ 103.If your results were as expected, ***leave the command prompt without further input for now and proceed to the next numbered step.*** If you see an error message, correct the problem before you continue. **Some suggestions to correct the problems are:**

__ a. Check the error by using the `mqrc` command from the Windows command prompt followed by the error number if you need an explanation of the error.

__ b. If you need more information, also check the error that is associated with your `TSM00##.SVRCONN` channel in the z/OS log. If there is no corresponding error in the z/OS side, then the problem should be in the Windows client. Check your `CLNTCONN` channel definition and the setting of the `MQSSLKEYR` variable.

__ c. If there is an error on the z/OS side, review the error. Remember that the `CLNTCONN` channel needs the label of the Windows certificate, and the `SVRCONN` channel needs the label of the `secondCAMQ##` certificate. Both channels need the same `SSLCIPH` value.

__ 104.You are expected to reach this step after a successful invocation of the `amqsputc` sample program. ***Do not enter any data in the Windows command prompt or press the Enter key yet***.

__ a. You now check the channel status in the z/OS side.

__ b. The `amqsputc` program should still be running.

__ c. From the z/OS SDSF log panel, type:

```
MQ## DIS CHS(TSM00##.SVRCONN) SSLCERTI SSLPEER
```

__ d. Press the Enter (Ctrl) key and review the results. Since other students work simultaneously in the same z/OS image, ensure that you are looking at the results for your channel.

__ e. Note the values that are shown in the SSLPEER field when the channel is active. The display shows *selected output of the `DIS CHS ALL` option*, and uses `TSM0001.SVRCONN` for illustrative purposes:

```
CSQM201I MQ01 CSQMDRTC  DIS CHSTATUS
CHSTATUS(TSM0001.SVRCONN)
CURRENT
CHLTYPE(SVRCONN)
STATUS(RUNNING)
SUBSTATE(RECEIVE)
...
QMNAME(MQ01)
RAPPLTAG(Sphere MQ bin64 amqsputc.exe)
SSLCERTI(CN=STUDENT 01 CA2,T=IBM
MQ,OU=TRAINING,O=IBM,L=HURSLEY,ST=HAMPSHIRE,C=UK)
SSLCERTU(TSM0001)
SSLRKEYS(0)
SSLKEYTI()
SSLKEYDA()
SSLPEER(SERIALNUMBER=02,CN=MQ01 WIN CA2 PERS,T=CLIENT,OU=TRAINING,
O=IBM,L=ORLANDO,ST=FLORIDA,C=US)
...
RVERSION(08000001)
MCAUSER(TSM0001)
LOCLADDR(10.31.187.114(3101))
```

__ 105.After you review the output of the channel status command, return to your VMware image, and send three messages by typing three lines of `amqsputc` input.

__ 106.Press the Enter key one time, without typing any text, to end `amqsputc`.

__ 107.Return to z/OS and check your messages.

__ a. From the Command input, type `=H` to go to the message handler utility.

__ b. Type your queue manager name `MQ##` in the Queue Manager Name input line.

__ c. Type `PRICE.CL` for the Queue Name line.

__ d. Press the Enter (Ctrl) key. You should see the messages. The contents of these messages are not expected to be converted to EBCDIC by the message handler utility, but you should see the messages in the queue.

You completed the SSL lab.

Process to remove and replace a queue manager certificate

Note

Use this section only if needed to rectify date range problems. The instructions in this section are provided to resolve date range problems such as the “certificate not trusted” condition in the lab environment for the WM312 course. Steps needed to resolve this problem outside the WM312 environment might differ, depending on the situation.

The step numbers in this section restart at 1, indicating a corrective process section that is skipped under normal circumstances.

These instructions assume that the certificate error is discovered before connecting the certificate to the queue manager ring. If the certificate is added to the queue manager ring, then job R993RMFR must be edited and submitted first to remove the certificate from the ring.

Important

The date range error might occur when you are working with either the primary or secondary certificates. When you work through the process that is described in this section, take care to log on with the appropriate TSO ID.

If the problem is with the **primary queue manager** first or second certificate, or with the **Windows client certificate**, log on with your lowest numbered TSO ID. If the problem is with the **secondary queue manager** certificate, log on with your highest numbered TSO ID.

An example of the outputs and inputs that are detailed in this section is included at the end of this section.

__ 1. Edit job `R992DLCR`. Take care to provide the correct certificate label name and TSO ID, depending on which certificate needs to be removed.

  __ a. Inspect the job to confirm the job name that displays in SDSF.

  __ b. Submit the job.

  __ c. Check results, ensuring that there are no error indications at the READY prompt displays.

__ 2. Display the signer CA date range. If the job output that created the associated signer CA is still available in SDSF, review the signer CA dates contained on that job output.

__ 3. If the job output that created the associated signer CA is not available, edit job `R994DSCA` replacing all pertinent placeholders with the correct information for the signer CA needed.

  __ a. Note the name of the job as it displays on SDSF: `MQ##DSCA`

  __ b. Review the results.

__ 4. Edit the job that was originally used to generate the queue manager certificate and include the correct `GENCERT` date range parameters. Check the example in this section for the syntax.

Note

The original certificate generation job might be `R112CRCR`, `R122CRC2`, `R212CRCS`, or `R213CRCW`.

__ a. Resubmit the specific queue manager certificate generation job.

__ b. Carefully check the output to ensure that the date range situation is rectified and the certificate was created and displayed successfully.

Example

The following example uses `MQ01` with CA `ST01CA1` for illustration. You deleted the queue manager certificate that was created with `NOTRUST` status. You now need to determine the problem with the queue manager certificate date range. Observe the End Date of the signer CA: 2016/04/21. Now check the End Date of the queue manager certificate: 2016/04/22, which exceeds the End Date for the signer.

The signer CA date range shows:

```
Digital certificate information for CERTAUTH:

  Label: ST01CA1
  Certificate ID: 2QiJmZmDhZmjgeLj8PHDwfFA
  Status: TRUST
  Start Date: 2015/04/21 00:00:00
  End Date:   2016/04/21 23:59:59
```

The queue manager certificate date range shows:

```
READY
RACDCERT ID(TSM0001) GENCERT SUBJECTSDN(CN('MQ01 PERSONAL') T('QMGR')...
... WITHLABEL('ibmWebSphereMQMQ01') SIGNWITH(CERTAUTH LABEL('ST01CA1'))
The certificate that you are creating has an incorrect date range. The
certificate is added with NOTRUST status.
READY
RACDCERT LIST(LABEL('ibmWebSphereMQMQ01')) ID(TSM0001)

Digital certificate information for user TSM0001:

  Label: ibmWebSphereMQMQ01
  Certificate ID: 2Qfj4tTw8PDxiYKU5oWC4peIhZmF1NjU2PDx
  Status: NOTRUST
  Start Date: 2015/04/22 00:00:00
  End Date:   2016/04/22 23:59:59
```

You must resubmit the certificate generation job for the particular queue manager by specifying a `RACDCERT GENCERT NOTAFTER` date range parameter that does not exceed the End Date date range of the signer certificate. The following example shows a partial `GENCERT` with date parameters:

```
...
C('UK')) -
NOTBEFORE(DATE(2015/04/22) TIME(00:00:00)) -
NOTAFTER(DATE(2016/04/20) TIME(23:59:59)) -
WITHLABEL('ibmWebSphereMQMQ01')
...
```

Remember to check the output of the `RACDCERT GENCERT` job carefully to ensure that the certificate was created without errors.

End of exercise

Exercise review and wrap-up

For the first part of the exercise, Scenario 1, you:

- Worked with the RACF RACDCERT utility to generate CA signers and queue manager certificates
- Learned how to check the certificate valid date ranges
- Associated the certificates to specific queue managers that use a certificate ring
- Enabled queue managers to use SSL/TLS
- Configured selected channels to use SSL/TLS

In the second part of the exercise, Scenario 2, you:

- Extended the RACF RACDCERT work that was completed in Part 1 to add a second CA signer and a second queue manager certificate by using a nonstandard label name
- Learned how to configure channels in the situation where a queue manager can have certificates that are signed by more than one CA
- Learned how to configure the Windows IBM MQ client and use the CERTLABL

If you had to resolve the `NOTRUST` date situation, you:

- Learned how to check and compare data range issues between the CA signer and the queue manager certificate
- Deleted and re-created a queue manager certificate with valid date ranges

Exercise 3. Configuring channel authentication rules

What this exercise is about

In this exercise, you use preferred practices to configure channel authentication with focus on implementing SSL/TLS type rules.

What you should be able to do

After completing this exercise, you should be able to:

- Determine whether channel authentication is enabled in the queue manager
- Allow administrative users to connect to a selected channel
- Set up the “back-stop” rule
- Create a channel authentication rule that maps an incoming IP address to a user
- Create an SSL-based channel authentication rule that verifies the subject and issuer distinguished names

Introduction

In this exercise, you learn to work with channel authentication rules. You see the effects of some of the default rules first. You set the “back-stop” rule, and add necessary rules to allow known connections to your queue manager.

Many IBM MQ administrators disable the CHLAUTH feature when they experience blocked connections. In this exercise, you learn to use CHLAUTH to strengthen the security of your queue manager.

Requirements

Successful completion of Scenarios 1 and 2 in Exercise 2.

Exercise instructions

Preface

Warning

The objective of this exercise is to make you familiar with the channel authentication feature. Security needs vary among organizations. The channel authentication (CHLAUTH) work is intended to provide an initial understanding of how to set rules. It is not intended to be used as the standard on which rules to set. The course material offers suggestions on how to approach the work, but the rules set in your organization or engagement depend on the needs of the business and must be properly analyzed, planned, and evaluated.

Overall you should strive to work around the default rules, but leave them in place. It is a good idea to start with the back-stop rule, and add rules as needed. More rules, or changes to the recommendation, depend on the needs of the business.

Section 1: Review the channel authentication rules

Important

Log on with your primary TSO ID. You work with your primary queue manager first.

In this section, several users might enter commands simultaneously. ***Be careful that the results you check apply to your queue manager***.

Hint

By using the IBM MQ ISPF CSQUTIL option on Action 8 of the IBM MQ ISPF panel to type commands, you can isolate the results to your own TSM00##.CSQUTIL.OUTPUT view. You can also type commands in multiple lines by providing the **+** continuation character.

Before you press Ctrl (Enter) to go to the command editor, ensure that ***all three queue manager entry fields*** in the ISPF panel reflect the queue manager that you need to alter.

You do not need to enter the CPF string (`/MQ##`) when you use the ISPF CSQUTIL option. You are presented an Edit panel where you enter your MQSC commands, press the PF3 field to run the commands, and review the results in the `TSM00##.CSQUTIL.OUTPUT` file.

__ 1. Proceed to the ISPF IBM MQ panel by typing `=M` in the command input prompt and pressing the Enter (Ctrl) key.

  __ a. Type an `8` in the Action field.

__ b. Set **Object type** to `QLOCAL`, **Name** to `*`, and **disposition** to `QMGR`

__ c. Ensure that all three fields that contain the queue manager name – Connect Name, Target Queue Manager, and Action Queue Manager – contain your primary queue manager name: `MQ##`

__ d. Take all other defaults and press the Enter (Ctrl) key. You are presented with an Edit panel for a file named `TSM00##.CSQUTIL.COMMANDS`. Use this file as an ISPF edit file.

__ e. If there are any leftover commands in this file, delete them. Add one new line to type the command to display the channel authentication rules.

__ 2. Your queue manager is new, and no CHLAUTH rules are set yet. Any existing rules are the default rules that are included in the new queue manager. Type the command as shown in the display and press PF3 to see your results in the `TSM00##.CSQUTIL.OUTPUT` file.

```
DIS CHLAUTH(*) ALL
```

__ 3. Review the default rules, making sure to note the description, or purpose, for each rule.

Information

When you reviewed the rules, you noticed the `CHLAUTH('*') TYPE(BLOCKUSER)` for `USERLIST('*MQADMIN')`. Proceed to SDSF `=s;da`, and locate the channel initiator started task for your primary queue manager, `MQ##CHIN`. What user ID does it run under?

The user ID under which the channel initiator task runs qualifies as an MQADMIN type user. ***Keep this fact in mind for the next steps***.

Section 2: Enable channel authentication

__ 4. Display the queue manager channel authentication setting. Enter `=s;log` at the command input prompt and press the Enter (Ctrl) key to reach the SDSF system log. At the system log, type:

```
/MQ## DIS QMGR CHLAUTH
```

__ 5. If channel authentication (`CHLAUTH`) for your primary queue manager is disabled, enable it by typing the command in the display from the z/OS SDSF log, and pressing the Enter (Ctrl) key:

```
/MQ## ALTER QMGR CHLAUTH(ENABLED)
```

__ 6. Confirm that the command completed successfully.

Section 3: Review the results of the use of an administrative user ID with the channel authentication default rules enabled

__ 7. Start IBM MQ Explorer by double-clicking the IBM MQ Explorer icon in your Windows VMware image.

__ 8. In your Queue Managers menu, right-click your `MQ##` primary queue manager and click **Connect**. If you are still connected, right-click **Disconnect**, and then **Connect** again. What happens now?

__ 9. Go to your primary TSO ID session and look at the SDSF log panel. Find messages specific to the channel you used for IBM MQ Explorer, `TSM00##.XPL`. You should see a message similar to the display. ***The display uses queue manager MQ21 for illustrative purposes:***

```
+CSQX776E MQ21 CSQXRESP Channel TSM0021.XPL from 10.61.23.250 has been
blocked due to userid, Detail: MCAUSER(TSM0021)
CLNTUSER(Administrator)
```

STOP **Stop**

The `CSQX776E` message can be misleading. The `CHLAUTH` rule is blocking due to the administrative user ID, but in this case, it is referring to TSM0021. TSM0021 is considered an administrative user ID because the channel initiator for queue manager MQ21 runs under TSM0021.

This error is not exclusive to IBM MQ Explorer. The same error occurs if an application that runs under what IBM MQ considers an administrative ID tries to connect to a queue manager. For example, if you run the test with `amqsputc` by using the `TSM00##.XPL` channel, instead of using the channel that is defined in the CCDT, you also receive the `CSQX776E` channel blocked message. You now test the same `TSM00##.XPL` channel with `amqsputc`.

__ 10. You now create a rule to allow access to the `TSM00##` administrative user for channel `TSM00##.XPL`.

Section 4: Set a rule to enable a specific channel to be excluded from the BLOCKUSER default rule

Rather than remove the BLOCKUSR rule, bypass the BLOCKUSR rule by adding a new, more specific BLOCKUSER rule.

For the new rule, by using the full channel name, the rule is more specific. The `USERLIST` cannot be blank because a blank user list does not override the default rule, which has at least one ID value. This value is (`'*MQADMIN'`). To get around the rule behavior, you provide a bogus user ID.

__ 11. The channel that you configured in your IBM MQ Explorer client is `TSM00##.XPL`.

__ 12. Proceed to the IBM MQ ISPF MQSC panel by entering `=m` from the command input prompt.

__ 13. At the MQSC command edit file, remove any old commands and enter the command to remove the BLOCKUSR rule. Type the command exactly as shown, taking care to include the apostrophes, and then press PF3. Do ***not*** copy and paste.

```
SET CHLAUTH('TSM00##.XPL') TYPE(BLOCKUSER) USERLIST('GETMETRU')
```

It is assumed GETMETRU does not exist as a privileged user. The command should fit on one line of the command input file, but if you end up by using two lines, add the **+** continuation character. After pressing PF3:

__ a. Confirm that the command ended successfully with “failed 0” results for commands completed.

__ b. Return to your ISPF MQSC command input, and display CHLAUTH again. At the command input data set, enter `DIS CHLAUTH(*)` and press PF3. You should see ***four*** rules, which should include the new rule.

Hint

When you first start working with CHLAUTH rules, it is a good idea to keep count of the number of rules, particularly when the action is ALTER. When you use ALTER, you must confirm that a rule was changed, not inadvertently added.

__ c. Return to IBM MQ Explorer, right-click the `MQ##` queue manager, and select **Connect**. The connection to the queue manager should now work.

__ 14. If any part of the previous step did not work as expected, stop and make any needed corrections before proceeding. You are not able to continue until the previous step works. Request assistance if necessary.

Section 5: Test your client connection and set any needed rules

__ 15. Open a Windows command prompt in your VMware image.

__ 16. Set the environment variable so the IBM MQ client can locate the key repository by typing the command *exactly as shown*, without a suffix at the end of the key database name:

```
set MQSSLKEYR=C:\certs\wm312
```

__ 17. Test the client connection by typing: `amqsputc PRICE.CL`

From the SDSF log in your primary TSO session, your result should resemble the display, which uses TSM0004 as an example:

```
+CSQX776E MQ04 CSQXRESP Channel TSM0004.SVRCONN from 10.8.66.1 has been
blocked due to userid, Detail: MCAUSER(TSM0004)
  CLNTUSER(Administrator) SSLPEER(SERIALNUMBER=02,CN=MQ04 WIN CA2
 +CSQX512I MQ04 CSQXRESP Channel TSM0004.SVRCONN no longer active
```

Information

TSM00##.SVRCONN is the channel that you configured with SSL. Notice how the SSLPEER value of the channel is displayed in the error.

__ 18. Channel TSM00##.SVRCONN has the channel initiator user ID in the MCAUSER attribute, which requires a TYPE(BLOCKUSER) rule. Return to the ISPF MQSC panels by typing `=m` in the command input prompt and pressing the Enter (Ctrl) key.

__ 19. The action should still be 8; press the Enter (Ctrl) key to reach the MQSC command file. The rule to be typed is similar to the earlier rule. If your previous command is still there, you change the channel name, and the rest of the rule is the same. The CHLAUTH rule is:

```
SET CHLAUTH('TSM00##.SVRCONN') TYPE(BLOCKUSER) USERLIST('GETMETRU')
```

__ 20. Check the results to make sure that the rule is created successfully.

__ 21. Retest from the Windows command prompt by typing: `amqsputc PRICE.CL`

__ 22. The test is expected to work. If you still have errors, resolve the problem before you continue. Request assistance if necessary.

Note

You enabled channel authentication in the queue manager and created BLOCKUSER type rules to allow administrative users to connect to the queue manager.

Next, you create the back-stop rule to connectivity information other than the user ID to block connections for the queue manager.

After the back-stop rule is added, each failure must be addressed independently.

Section 6: Set the back-stop rule

The back-stop rule is a concept and a name that is given to a specific CHLAUTH of type ADDRESSMAP definition that inhibits all connections. Since this rule can cause significant inconvenience to unsuspecting users, it can be set in “warning” mode first. You can use warning mode to see what connections would be disallowed, and it provides information on which rules should be set to allow valid connections. In class, the rule is entered in normal blocking mode, not in warning mode.

Administrators do not often use the description attribute, but when setting CHLAUTH rules, it is a good practice to document what the rule is intended for.

You now set the “back-stop” rule in regular mode. **Do *not* set the rule in warning mode for this exercise**.

__ 23. Proceed to the IBM MQ ISPF MQSC command input prompt by entering `=m` at the command input prompt and pressing the Enter (Ctrl) key.

__ 24. Confirm that the panel is set to your queue manager in all three queue manager fields. Enter an 8 in the Action field and press the Enter (Ctrl) key.

__ 25. At the `CSQUTIL` command input file, enter the back-stop rule as shown:

Important

Take care to use the apostrophes as shown in all CHLAUTH commands. Missed apostrophes can lead to errors that are difficult to troubleshoot.

Do not attempt to copy and paste, as it results in invalid characters. Type all commands.

Remember, you are ***not*** using warning mode. ***You do not see any channel locked messages in the system log if you use warning mode***.

```
SET CHLAUTH('*') TYPE(ADDRESSMAP) ADDRESS('*') +
USERSRC(NOACCESS) DESCR('Back-stop rule')
```

Section 7: Check your IBM MQ Explorer connection and add an ADDRESSMAP type rule for the TSM00##.XPL channel

__ 26. Return to, or reopen, IBM MQ Explorer.

__ 27. If your session to your primary queue manager is connected, right-click your primary queue manager and select **Disconnect**.

__ 28. Right-click your primary queue manager and select **Connect**. It is expected that the connection fails. Review the results.

__ 29. Proceed to your primary TSO session, type `=s;log` and press the Enter (Ctrl) key.

__ 30. Look for your channel name. It is expected that the message looks similar to the display, which uses TSM0004 as an example:

```
+CSQX511I MQ04 CSQXRESP Channel TSM0004.XPL started
 connection 10.8.66.1
+CSQX777EMQ04CSQXRESPChannelTSM0004.XPLfrom10.8.66.1(10.8.66.1)
 has been blocked due to USERSRC(NOACCESS), Detail:
 CLNTUSER(Administrator)
+CSQX512I MQ04 CSQXRESP Channel TSM0004.XPL no longer active
```

__ 31. When you locate your channel name, make note of the blocked IP address and remember the MCAUSER, TSM00##. You need this information to create the ADDRESSMAP type rule.

Important

Notice how the resulting error changed from CSQX776E to CSQX777E. The error no longer mentions that the channel was blocked due to the user ID. This detail is an important distinction when you work with and troubleshoot channel authentication rules.

__ 32. Create a rule to allow the channel to connect. Return to the ISPF MQSC command by typing `=m` in the command input prompt and pressing the Enter (Ctrl) key.

__ 33. Press Enter (Ctrl) again to get to the MQSC command edit panel (it is assumed that Action is still set to 8 in the IBM MQ ISPF panel).

__ 34. You create the type ADDRESSMAP rule by using the template in the display. Replace the `##` placeholder with your primary student number, and the `??` placeholders with the IP address that you saved from the CSQX777E message:

```
SET CHLAUTH('TSM00##.XPL') TYPE(ADDRESSMAP) +
ADDRESS('??.??.??.??') MCAUSER(TSM00##)
```

__ 35. Check your rule carefully before you create it. After you reviewed your rule, press PF3 to create the rule.

__ 36. Check the results in the MQSC output panel and make any needed corrections. If you must make corrections, ensure that you use all required attributes to prevent creating an extra rule instead of changing the intended rule.

__ 37. Return to IBM MQ Explorer, right-click your primary queue manager, and select **Connect**. It is expected that after you set the new `ADDRESSMAP` rule, the connection is allowed.

__ 38. Proceed to your primary TSO session and check the start message for your channel.

Section 8: Check your client and set an SSLPEERMAP rule to allow the client connection to the TSM00##.SVRCONN rule

__ 39. Return to your Windows command prompt.

__ 40. Test your client connection now that the channel authentication “back-stop” rule is in place. Type `amqsputc PRICE.CL` and press the Enter key.

__ 41. The test is expected to fail with return code 2035.

__ 42. Proceed to your primary TSO session and find the last message for your `TSM00##.SVRCONN` channel. The result should look similar to the display, which uses TSM0004 as an example:

```
+CSQX777E MQ04 CSQXRESP Channel TSM0004.SVRCONN from 10.8.66.1
 (10.8.66.1) has been blocked due to USERSRC(NOACCESS), Detail:
 CLNTUSER(Administrator) SSLPEER(SERIALNUMBER=02,CN=MQ04 WIN CA2
+CSQX512I MQ04 CSQXRESP Channel TSM0004.SVRCONN no longer active
```

Information

IBM MQ V8 added the ability to check the certificate issuer along with the user certificate. You use the SSLPEER attribute to match the user certificate distinguished name, and the SSLCERTI attribute to match the issuer distinguished name.

Pay special attention to single and double quotation marks in the syntax.

__ 43. To create this rule, you need the exact distinguished names from the issuer and the client certificates. You can obtain this information in two ways:

__ a. If your `MQ##CRCW` job output is still available, you can obtain the Issuer and Subject name from the job listing.

__ b. Check the two JCL members that are used to create the primary queue manager, secondary CA, and client certificate. These members are R211CRCA and R213CRCW in your TSM00##.SCSQPROC library.

__ 44. After you locate the needed distinguished names, proceed to the IBM MQ ISPF panels by typing =m and pressing the Enter (Ctrl) key, followed by the Enter (Ctrl) key to reach the MQSC edit panel.

__ 45. Clear any remaining rules, and type your SSLPEERMAP rule. Use the template that is provided, replacing the ## placeholder with your primary student number. **Do not copy and paste from the student guide text as it results in errors from invisible special characters. Be careful with the required single and double quotation marks**:

```
SET CHLAUTH('TSM00##.SVRCONN') TYPE(SSLPEERMAP) +
SSLPEER('CN="MQ## WIN CA2 PERS",O="IBM" ') +
SSLCERTI('CN="STUDENT ## CA2",O="IBM" ') +
MCAUSER(TSM00##)
```

__ 46. Carefully review your rule.

__ 47. Press PF3 to create it. Check the results in the MQSC output panel.

__ 48. Return to your Windows command prompt. Test the client by typing amqsputc PRICE.CL and pressing the Enter key.

__ 49. Check results. If you are able to type messages, your rule was successful. If you had any errors, make necessary corrections before you proceed.

Note

You completed the channel authentication rule exercise and set the “back-stop rule” in your primary queue manager. Subsequent use of channels to your primary queue manager requires new rules. There are two instances in the *Intra-group and distributed queuing with queue-sharing groups* exercise that require a new rule. Where a new rule is required in the course, a sample template is provided for the rule.

You are urged to keep channel authentication enabled in your primary queue manager. Take the opportunity to set the two extra rules that are needed in the *Intra-group and distributed queuing with queue-sharing groups* exercise to reinforce your practice with channel authentication rules.

End of exercise

Exercise review and wrap-up

In this lab exercise, you:

- Checked whether channel authentication is enabled in the queue manager
- Set up rules to allow administrative users to connect to a selected channel without the need to disable channel authentication
- Learned to use and set the “back-stop” rule as a first step to implement channel authentication
- Created a channel authentication rule that maps an incoming IP address to a user
- Created an SSL-based channel authentication rule that verifies the subject and issuer distinguished names

Exercise 4. Configuring queue-sharing groups and shared queues

What this exercise is about

This exercise reinforces the topics in the queue-sharing group unit. It provides configuration experience on the IBM MQ queue-sharing group implementation. You also define how to implement message offload by using shared message data sets (SMDS), and how to back up your queue-sharing group.

What you should be able to do

After completing this exercise, you should be able to:

- Review existing coupling facility information
- Define a queue-sharing group to DB2
- Add a queue manager to the queue-sharing group in DB2
- Complete the steps necessary to implement the queue-sharing group in IBM MQ
- Explain how to create the IBM MQ CFSTRUCT object
- Define shared queues
- Put and get messages from the shared queues
- Configure message offload to a shared message data set (SMDS)
- Issue necessary commands to administer queue-sharing groups and shared message data sets
- Back up the structures

Introduction

In this exercise you configure a queue-sharing group, add the primary and secondary queue managers to this group, and work with shared queues and related commands.

Requirements

- Successful completion of Exercise 1 with primary and secondary queue managers available
- Preconfigured coupling facility structures

- Understanding of “primary” and “secondary” terminology for this course

Exercise instructions

Preface

The coupling facility configuration for this exercise uses the primary (`##` placeholder) and secondary (`@@` placeholder) student numbers.

Important

You might want to mark this page; you need the information in this preface section to complete this exercise.

The names students use for this lab are:

- Queue-sharing group: One per student, named `SG##`. You use the same queue-sharing group name for your primary and secondary queue managers, as they both need to join the same group.
- List or application structures:
  - Queue manager structure for administrative use only: `SG##CSQ_ADMIN`

    If you need to provide the administrative structure name to an IBM MQ command, you use: `CSQ_ADMIN`
  - Primary queue manager CF structure name: `SG##APPL1`

    For IBM MQ, this name is: `APPL1`
  - Secondary queue manager CF structure name: `SG##APPL2`

    For IBM MQ, this name is: `APPL2`

Example

For a student with primary ID = 03 and secondary ID = 23:

- Queue-sharing group name is `SG03`
- Coupling facility structure that is used by primary queue manager MQ03 is named: `SG03APPL1`
  When you provide the structure name to IBM MQ, you use the short name: `APPL1`
- Coupling facility structure that is used by secondary queue manager MQ23 is named `SG03APPL2`
  When you provide the structure name to IBM MQ, you use the short name: `APPL2`
- The administrative structure is: `SG03CSQ_ADMIN`

  If you need to provide the administrative structure name to IBM MQ, you use `CSQ_ADMIN`.

The DB2 information that is needed for this exercise is:

- DB2 STEPLIB DD for MSTR procedure: **DSNDBA0.DB2B10.SDSNLOAD**
- DB2 data sharing group: **DSNDBA0**
- DB2 SSID depends on the LPAR being used:
  - JCL that runs in LPAR1 (where you configured your primary queue manager) uses SSID **DBA1**
  - JCL that runs in LPAR2 (where you configured your secondary queue manager) uses SSID **DBA2**

JCL members are copied and partially preconfigured to expedite the work in this exercise. A list of the original JCL member and the corresponding renamed member is included in Table 1.

Throughout this exercise:

- You might want to review the contents of the original JCL member before you start each section, but **always use the renamed JCL members for all work in this exercise.**
- Check that you are working in the correct TSO session and use the correct TSO ID.
- You use some members for the primary queue manager, other members for the secondary queue manager. Members have different “placeholders” to replace.

Table 4: Configuration JCL for the queue-sharing group exercises

Original JCL	Renamed JCL	Comments
CSQ45AQS	SG451AQS	Add sharing group SG## to DB2.
CSQ45AQM	SG452AQM	Add primary queue manager MQ## to sharing group SG##.
CSQ45AQM	SG453AQM	Add secondary queue manager MQ@@ to sharing group SG##.
CSQ4SMDS	SGQ4SMDS	Define and format the shared message data set.
CSQ4SMDS	SGQSMDS2	Define and format the shared message data set for the secondary queue manager.
CSQ4INSS	SGQ4INSS	CSQ4INSS input to CSQINP2 with updated IBM MQ CFSTRUCT objects.
CSQ4INSS	SGQINSS2	Second updated CSQ4INSS input to CSQINP2.
CSQ4ZPRM	SGQ4ZPRM	Parameter module that is updated to enable queue-sharing groups.
CSQ4MSTR	MQ##MSTR or MQ@@MSTR depending on your student numbers	Queue manager MSTR procedure.

Table 4: Configuration JCL for the queue-sharing group exercises

Original JCL	Renamed JCL	Comments
N/A	`SGMQPUT1`	MQPUT to primary queue manager
N/A	`SGMQPUT2`	MQPUT to secondary queue manager
N/A	`SGMQGET1`	MQGET from primary queue manager
N/A	`SGMQGET2`	MQGET from secondary queue manager

STOP **Stop**

The values that are used in the queue-sharing group configuration for this exercise are exclusively for lab purposes. ***It should not be assumed* that these values are optimal settings for a user environment.**

Review the IBM Knowledge Center and the capacity and performance references that are provided in the unit that precedes this exercise when planning your queue-sharing group environment.

Part 1: Create a queue-sharing group and configure and test the primary queue manager as the first member of this group

Section 1: Review the coupling facility information

Important

Log on with your primary TSO ID.

You might want to split your screen to alternate between your `TSM00##.WM31.SCSQPROC` PDS and the SDSF panels.

In this work, you differentiate across the components that work together in queue-sharing groups, which are coupling facility components, DB2 components, or IBM MQ components.

__ 1. The coupling facility structures are predefined in the z/OS system. Look at the structures for your primary queue manager by typing the z/OS display structure command exactly as shown in this step. Replace the `##` placeholder with your primary student number. Note the use of the asterisk as a wildcard character.

```
/D XCF,STR,STRNAME=SG##*
```

Important

Other students might use the same command simultaneously. Ensure that you look at the results for your assigned structure.

Do not worry if you see “`NOT ALLOCATED`” to the right of the structures displayed.

__ 2. Write down the name of the structures you find; there should be four structures. These structures use the z/OS naming convention, which includes the queue-sharing group name.

____

____

____

____

Section 2: Create your queue-sharing group by adding the entry to DB2

__ 3. Proceed to your `TSM00##.WM31.SCSQPROC` and edit member **`SG451AQS`**.

  __ a. Replace **all** occurrences of the `##` placeholder with your primary student number.

  __ b. Replace the `++DSGNAME++` placeholder with the name of the `DB2` data sharing group, `DSNDBA0`, while taking care to make the last character the number zero.

__ c. Replace the ++DB2SSID++ placeholder with the name of the DB2 SSID. For your primary queue manager, use DBA1.

__ d. Look at the command included in the PARM line:

```
'ADD QSG,SG##,DSNDBA0,++DB2SSID++'
```

__ e. A completed line is provided in the display, which uses SG03 as an example:

```
'ADD QSG,SG03,DSNDBA0,DBA1'
```

__ f. Scroll to the beginning of the JCL and note the job name: MQ##AQS1

__ g. Submit the job and check the results. Your results should look like the display. Student number 03 is used as an example:

```
CSQU000I CSQ5PQSG IBM WebSphere MQ for z/OS V8.0.0
 CSQU500I CSQ5PQSG Queue-sharing Group Utility - 2015-06-18 10:50:38
 CSQU501I  ADD QSG function requested
 CSQU503I  QSG=SG03, DB2 DSG=DSNDBA0, DB2 ssid=DBA1
 CSQU526I  Connected to DB2 DBA1
 CSQU537I  QSG SG03 entry successfully added to DB2 table CSQ.ADMIN_B_QSG
 CSQU551I  QSG SG03 entry successfully added
 CSQU528I  Disconnected from DB2 DBA1
 CSQU148I CSQ5PQSG Utility completed, return code=0
```

__ h. **If your results did not get a return code zero or did not indicate that the entry was successfully added, resolve the problem before you continue to the next step**.

Section 3: Add your primary queue manager to the queue-sharing group

__ 4. Return to your TSM00##.WM31.SCSQPROC PDS and edit member SG452AQM.

__ a. Replace **all** occurrences of the ## placeholder with your primary student number.

__ b. Check for the ++DGSNAME++ placeholder. If the ++DSGNAME++ placeholder is not prefilled, replace it with the name of the DB2 data sharing group, DSNDBA0, while taking care to make the last character the number zero.

__ c. Replace the ++DB2SSID++ placeholder with the name of the DB2 SSID. Use **DBA1**.

__ d. Ensure that all fields in the PARM command line are set.

__ e. Scroll to the beginning of the JCL and note the job name: MQ##AQM2

__ 5. Submit the job and check the results. Your results should have return code 0 and look like the display. Student number 03 is used as an example:

```
CSQU000I CSQ5PQSG IBM WebSphere MQ for z/OS V8.0.0
 CSQU500I CSQ5PQSG Queue-sharing Group Utility - 2015-06-18 11:46:43
 CSQU501I  ADD QMGR function requested
 CSQU502I  Queue manager=MQ03
 CSQU503I  QSG=SG03, DB2 DSG=DSNDBA0, DB2 ssid=DBA1
 CSQU560I  Full name of admin structure is SG03CSQ_ADMIN
 CSQU566I  Unable to get attributes for admin structure, CF not found or not
allocated
 CSQU526I  Connected to DB2 DBA1
 CSQU549I  QMGR MQ03 entry successfully added to QSG SG03
 CSQU528I  Disconnected from DB2 DBA1
 CSQU148I CSQ5PQSG Utility completed, return code=0
```

Note

Do not worry if you see the `CSQU566I` message; it is expected.

The structure is allocated later in this exercise. When you add the second queue manager in Part 2 of this exercise, you notice the difference in messages.

Section 4: Create your shared message data set for structure APPL1

Reminder

When you use ***more than one application structure*** for each queue manager, you must create one SMDS for each structure in the queue-sharing group. For example, in the case of the lab environment:

- **You create *one* SMDS in this step:** `MQ.MQ##.APPL1.SMDS`
- You define `MQ.MQ##.APPL2.SMDS` **later,** when you get ready to incorporate the secondary queue manager to the queue-sharing group
- ***After you have both queue managers in the queue-sharing group*,** there are six SMDS data sets, which include the CSQSYSAPPL data sets:

```
MQ.MQ##.APPL1.SMDS
MQ.MQ##.APPL2.SMDS
MQ.MQ##.SYSAPPL.SMDS
MQ.MQ@@.APPL1.SMDS
MQ.MQ@@.APPL2.SMDS
MQ.MQ@@.SYSAPPL.SMDS
```

When you provide the shared message data set name:

- To the `IDCAMS` and `CSQJUFMT` utilities, you use the full data set name

- To the DSGROUP attribute in the IBM MQ CFSTRUCT object, you:
  - Use the SMDS name for the application structure that matches the name of the structure that is used by the queue manager that is being configured, in this case APPL1
  - Replace the queue manager name with an asterisk (*****)

The shared message data set name is partially configured. You complete the customization.

__ 6. Return to your TSM00##.WM31.SCSQPROC PDS and edit member SGQ4SMDS.

__ a. Replace the ++cfstructure++ placeholder with the structure name you use for your primary queue manager: APPL1

__ b. Replace all instances of the ## placeholder with your primary student number.

__ c. Note the data set name. You use it when you define the IBM MQ CFSTRUCT object. Scroll to the beginning of the JCL and note the job name: MQ##SMD1

__ d. Submit the job and check results. There are two steps to check, the IDCAMS to create the data set, and then the format, CSQJUFMT. The results should resemble the display:

```
IDCAMS  SYSTEM SERVICES                                           TIME

   DELETE 'TSM0003.MQ03.APPL1.SMDS' ERASE CLUSTER
IDC3012I ENTRY TSM0003.MQ03.APPL1.SMDS NOT FOUND
IDC3009I ** VSAM CATALOG RETURN CODE IS 8 - REASON CODE IS IGG0CLA3-42
IDC0551I ** ENTRY TSM0003.MQ03.APPL1.SMDS NOT DELETED
IDC0001I FUNCTION COMPLETED, HIGHEST CONDITION CODE WAS 8

   SET MAXCC=0

   DEFINE CLUSTER                        -
          (NAME(MQ.MQ03.APPL1.SMDS)      -
           MEGABYTES( 75 75 ) -
           LINEAR                        -
           SHAREOPTIONS(2 3) )           -
       DATA                              -
          (NAME(MQ.MQ03.APPL1.SMDS.DATA) )
IDC0508I DATA ALLOCATION STATUS FOR VOLUME SMS001 IS 0
IDC0181I STORAGECLASS USED IS BASE
IDC0181I MANAGEMENTCLASS USED IS STANDARD
IDC0001I FUNCTION COMPLETED, HIGHEST CONDITION CODE WAS 0

IDC0002I IDCAMS PROCESSING COMPLETE. MAXIMUM CONDITION CODE WAS 0
CSQJ440I CSQJUFMT IBM WebSphere MQ for z/OS V8.0.0
CSQJ491I CSQJUFMT Log Data Set Preformatter Utility - 2015-06-22 17:04:59
CSQJ492I Log data set name = MQ.MQ03.APPL1.SMDS
CSQJ496I Log preformat completed successfully, 256140 records formatted
```

__ e. Type `=3.4` in your ISPF command input line and press the Enter (Ctrl) key to look for your shared message data set. At the `Dsname` level entry field, type `MQ.MQ##` but replace the `##` placeholder with your primary student number.

__ f. Check that you find two entries; the second entry has the DATA suffix in the name.

Section 5: Create SMDS for CSQSYSAPPL group recovery structure

__ 7. Return to your `SGQ4SMDS` member and update the job to create the GROUPUR-related SMDS by changing all instances of `APPL1` to: **`SYSAPPL`**

__ 8. Check your changes, and submit the JCL.

__ 9. Check the results of your job. If necessary, correct any errors before you continue.

Check your SMDS data sets by typing `=3.4` in the command input prompt and checking for any files that start with `MQ.MQ##`, where `##` is your primary student number.

Important

The check that the file names are created as expected is **critical**.

- First, check that the data sets were created similar to the display in the next step.
- Next, check that you do not see any data sets named `MQ.MQ##.APPL1.SMDS` or `MQ.MQ##.SYSAPPL.SMDS` **that use the actual `##` placeholder instead of your queue manager name**.
- If you find any data sets with an actual `##` placeholder in the data set list, **make sure that you did not create it** by going back to your JCL output and checking. The `MQ##` data set might belong to another student. This check prevents errors that require extra time to check later.

__ a. At the `Dsname Level` entry field, type `MQ.MQ##` and press the Enter (Ctrl) key. Your results should resemble the example output, which uses MQ24:

```
MQ.MQ04.APPL1.SMDS
MQ.MQ04.APPL1.SMDS.DATA
MQ.MQ04.SYSAPPL.SMDS
MQ.MQ04.SYSAPPL.SMDS.DATA
```

__ b. You **do not** want to find any: **`MQ.MQ##.APPL1.SMDS`** or **`MQ.MQ##.SYSAPPL.SMDS`**.

__ c. **If your results did not get a return code zero, or if you do not find your SMDS entries in the list, resolve the problem before you continue to the next step**.

Section 6: Customize the IBM MQ CFSTRUCT and other IBM MQ queue-sharing group object definitions – CSQ4INSS

Reminder

As noted in the lecture material, SCSQPROC member CSQ4INSS does not reflect the most recent IBM MQ `CFSTRUCT` object attributes. If you work with an IBM MQ V7.1 or later environment outside this course, you might need to manually add the new attributes. You might also need to update existing attributes in the IBM MQ `CFSTRUCT` object definitions that are found in CSQ4INSS.

__ 10. Return to your `TSM00##.WM31.SCSQPROC` PDS and **browse** member `CSQ4INSS`.

__ a. Find the string `'system cf structure'` exactly as shown including the quotation marks.

__ b. Observe how these definitions show `CFLEVEL(3)`, and there are no new attributes such as the SMDS offload values or CFCONLOS.

__ c. From the top of the data set, do a find for string `++cfstructure++`.

__ d. Note the different objects where this string is used.

__ e. Exit CSQ4INSS and proceed to the renamed, updated version of this member.

Important

Ensure that you use member SGQ4INSS in the rest of this section.

__ 11. Open member `SGQ4INSS` at `TSM00##.WM31.SCSQPROC`.

__ 12. Notice that this `SGQ4INSS` is the updated member and contains new attributes. If you do not find new attributes such as CFCONLOS in the IBM MQ `CFSTRUCT` object, ensure that you are updating the correct member, `SGQ4INSS`.

__ a. If you find any occurrences of the `++cfstructure++` placeholder, replace them with APPL1. **The ++cfstructure++ placeholder might be already preconfigured to APPL1.** APPL1 is the name of the XCF application structure that is used in the primary queue manager.

Note

`SYSTEM.QSG.TRANSMIT.QUEUE` uses `CFSTRUCT` **APPL1**.

`SYSTEM.QSG.CHANNEL.SYNCQ` and `SYSTEM.QSG.UR.RESOLUTION.QUEUE` use the SYSAPPL structure.

The shared SYSTEM.* queues need to be defined one time in the first queue manager to join the queue-sharing group.

__ b. SGQ4INSS has some attributes that are preconfigured for this step. Check the values in the file by typing `F CFSTRUCT` in the edit command input line.

__ c. Repeat the find one time by pressing PF6 so that you reach the IBM MQ `CFSTRUCT` object definition for your primary queue manager application structure. Your screen should look like the display:

```
DEFINE CFSTRUCT( 'CSQSYSAPPL') +
          DESCR( 'System CF structure' ) +
          CFLEVEL( 5 ) +
          OFFLOAD(SMDS) +
          OFFLD1TH(25) +
          OFFLD1SZ(1K) +
          OFFLD2TH(45) +
          OFFLD2SZ(2K) +
          OFFLD3TH(70) +
          OFFLD3SZ(0K) +
          DSGROUP('MQ.*.SYSAPPL.SMDS') +
          DSBLOCK(256K) +
          RECOVER( YES )
```

__ d. The short form of the structure name is used in several places in CSQ4INSS and SGQ4INSS. Observe the `*` in the DSGROUP SMDS name. It is correct; leave as is.

__ e. Observe that the offload percentage and sizes (**`OFFLDxTH`** and **`OFFLDxSZ`**) parameters are preconfigured, and changed from the initial values to cause messages to overflow to the SMDS sooner for this course. Leave these values as coded in SG04INSS. Check that CFLEVEL is 5.

__ f. Check that the structure is recoverable.

- `OFFLOAD = SMDS`
- `DSGROUP` name = **`MQ.*.SYSAPPL.SMDS`**

__ 13. Repeat find for your application structure, APPL1. Remember that you are providing this name to IBM MQ, so you use the short name for the structure: `APPL1`

__ a. Notice how the SMDS name is specified. Check the following items:

- Check that the structure is recoverable.
- `OFFLOAD = SMDS`
- `DSGROUP` name = **`MQ.*.APPL1.SMDS`**
- The SMDS offload rule attributes are also modified as documented earlier for purposes of this exercise.

__ b. Save `SGQ4INSS`. These definitions are created the next time the queue manager starts.

Section 7: Update the primary queue manager MQ##MSTR procedure

__ 14. Edit member MQ##MSTR where the ## placeholder represents your primary student number.

__ 15. Find the STEPLIB concatenation.

__ 16. Check that the DB2 load library is added to the end of the STEPLIB concatenation. The entry should be:

```
DD DSN=DSNDBA0.DB2B10.SDSNLOAD,DISP=SHR
```

__ 17. If the DB2 library is already in the STEPLIB concatenation, ***proceed to the next step***. If the DB2 library is not at the end of the STEPLIB concatenation, add a DD statement to include the DB2 library in the MQ##MSTR procedure.

__ 18. Find DSN=TSM00##.WM31.SCSQPROC(CSQ4INYG) in the CSQINP2 concatenation.

__ 19. Add your **SG**Q4INSS member directly before the CSQ4INYG data set. **SG**Q4INSS is the modified copy of the queue-sharing group definition member, CSQ4INSS. The CSQINP2 concatenation should now look like the partial display:

```
DD ....
DD DSN=TSM00##.WM31.SCSQPROC(SGQ4INSS),DISP=SHR
DD DSN=TSM00##.WM31.SCSQPROC(CSQ4INYG),DISP=SHR
DD ....
```

Warning

If you do not have SGQ4INSS ***before*** CSQ4INYG, group unit of recovery (GROUPUR) does not get enabled in the queue manager.

This error occurs because the ALTER QMGR GROUPUR(ENABLED) is done in the CSQ4INYG member before the SYSTEM.QSG.UR.RESOLUTION.QUEUE gets defined by the SGQ4INSS member. IBM MQ checks prerequisites before the ALTER QMGR completes, and queue SYSTEM.QSG.UR.RESOLUTION.QUEUE is one of the prerequisites checked. In this case, you see the message CSQM507E MQxx CSQMAMMS GROUPUR queue manager attribute was not enabled, CODE=2 in your queue manager started task.

If you saw the CSQM507E error, for the lab environment, you can restart the queue manager. Since the queue is now created, GROUPUR proceeds to get enabled. In your organization, ensure that your DD statements are in the correct order in the CSQINP2 concatenation.

__ 20. Take a moment to confirm that you used the correct member: **SG**Q4INSS

Use of the incorrect queue-sharing group definition member causes errors later in this exercise.

__ 21. Save your changes to MQ##MSTR.

__ 22. Copy the updated `MQ##MSTR` to replace the member with the same name at `SYS3.STUDENT.PROCLIB` by using the =3.3 option in the ISPF panels. Ensure that you enable replacement of an existing PDS member by typing a forward slash (`/`) in the "Replace like-named members" field of the second copy panel.

```
/  Replace like-named members
```

__ 23. Go to `SYS3.STUDENT.PROCLIB` and confirm that the updated procedure replaced the old copy of the `MQ##MSTR` procedure.

STOP **Stop**

Take special care that your `MQ##MSTR` procedure is indeed copied to `SYS3.STUDENT.PROCLIB`. If the updated procedure is not copied to `SYS3.STUDENT.PROCLIB`, the queue manager starts without the definitions from the `SGQ4INSS` member. If the queue manager starts without `SGQ4INSS`, it attempts to create a `CFSTRUCT` definition, which is created incorrectly.

This situation might be difficult to resolve.

Section 8: Update, reassemble, and rename the queue manager system parameter module

__ 24. Edit member `SGQ4ZPRM`.

__ 25. Replace all occurrences of the `##` placeholder with the name of the queue-sharing group that you created. Remember to use the short form of the name as IBM MQ expects: `APPL1`

__ 26. Replace all occurrences of `+DBA+` with the DB2 subsystem ID that you used to create the queue-sharing group. For this primary queue manager, use `DBA1`.

__ 27. Find the `CSQ6SYSP` macro.

__ a. Find the **`OPMODE`** parameter. It needs to be set to `OPMODE=(NEWFUNC,800)`, but observe the suggestion in the hint:

Hint

To ensure that the **`X`** at the end of the line stays lined up, change `OPMODE` to **`NEWFUNC`** in the command input line by typing:

```
C COMPAT NEWFUNC
```

and pressing the Enter (Ctrl) key. If the **`X`** continuation is not lined up in the JCL member, the assembly fails, and it might be time-intensive to correct.

__ b. Double check the **`QSGDATA`** parameter. It should reflect the global changes that you made. For example, for a group that uses SSID DBA1 and queue-sharing group SG03, QSGDATA needs to be set to:

```
QSGDATA=(SG03,DSNDBA0,DBA1,4,4)
```

__ 28. Go to the start of the SGQ4ZPRM JCL member. Carefully scroll down and confirm that all the **X** continuation characters line up on the right margin. The job fails if the Xs are not lined up. Make any correction if needed to line up the Xs.

__ 29. Scroll down to the bottom of the JCL. Notice the new name of the parameter module. After you submit the job, when it assembles correctly, this module is place in the expected library.

__ 30. Write down the name of the new parameter module if necessary. The renamed version has the parameters that are required to enable your queue manager to be in the queue-sharing group. The name should be in the form: MQ##ZQSG

You need to use this parameter module name for all subsequent restarts of the queue manager.

__ 31. Scroll to the top of the parameter module JCL and note the job name, MQ##ZPR1, where you replace ## with your primary student number.

__ 32. Submit the job and check results. There should not be any error messages, and all condition codes should be zero. Correct any errors before proceeding with further work on this exercise.

Section 9: Enable group unit of recovery and intra-group queuing

__ 33. Edit member CSQ4INYG.

__ 34. Do a find for queue manager attribute IGQ.

__ 35. Change it from DISABLED to **ENABLED** and press PF3 to save your changes.

__ 36. Locate attribute GROUPUR. If it is DISABLED, change it to **ENABLED**.

Section 10:Restart the queue manager with the new parameter module

__ 37. Proceed to SDSF by typing =S;log in the command input prompt and pressing the Enter (Ctrl) key.

__ 38. Type /MQ## STOP QMGR where ## is your primary student number. Wait until you see the CSQ3100I MQ## CSQ3EC0X - SUBSYSTEM MQ## READY FOR START COMMAND message before you restart the queue manager.

__ 39. Start your queue manager by entering the start command with the new parameter module:

```
/MQ## START QMGR PARM(MQ##ZQSG)
```

Troubleshooting

If your queue manager does not start, it might be a problem with the parameter module or an error in your MQ##MSTR procedure. Go to the failed queue manager job by typing =s;st and pressing the Enter (Ctrl) key. Select the failed MQ##MSTR task for your queue manager and carefully review the output for any error messages, other than the memory dump reason and code.

STOP Stop

All subsequent starts of the queue manager for this course must use the new parameter module; *do not use `MQ##ZPRM`*. For example, if starting queue manager MQ03, the new start command to use the rest of this course is:

```
/MQ03 START QMGR PARM(MQ03ZQSG)
```

__ 40. Proceed to your `MQ##MSTR` started task output in the SDSF typing `=s;da` in the command input prompt and pressing the Enter (Ctrl) key. Select your running `MQ##MSTR` task.

__ 41. Review the messages that are displayed as the `MQ##MSTR` region starts. There should not be any error messages.

__ 42. Display your IBM MQ `CFSTRUCT` object for your `MQ##` queue manager to ensure that the application structure definition completed correctly.

__ a. Go to the system log by typing `=s;log` in the command input prompt and pressing the Enter (Ctrl) key.

__ b. Type `/MQ## DIS CFSTRUCT(APPL1) ALL` and press the Enter (Ctrl) key. Ensure that you use the **`ALL`** option.

__ c. Confirm that the IBM MQ application structure `APPL1` is defined as expected. You should see the new attributes. For example, for queue manager MQ03 the output should resemble the display:

```
CSQM201I MQ03 CSQMDRTC  DIS CFSTRUCT DETAILS
 CFSTRUCT(APPL1)
 DESCR(System CF structure)
 CFLEVEL(5)
 RECOVER(YES)
 OFFLOAD(SMDS)
 OFFLD1TH(25)
 OFFLD1SZ(2K)
 OFFLD2TH(45)
 OFFLD2SZ(1K)
 OFFLD3TH(70)
 OFFLD3SZ(0K)
 DSGROUP(MQ.*.APPL1.SMDS)
 DSBLOCK(256K)
 DSBUFS(100)
 DSEXPAND(YES)
 RECAUTO(YES)
 CFCONLOS(ASQMGR)
... ... ...
```

Some of the attributes that you see are the offload percentages and sizes. The values that are used are different from the default values to help force offload to SMDS, which is done in a later step. Notice your shared message data set name in the DSGROUP parameter, and the connection toleration CFCONLOS set to ASKQMGR.

Important

If the CFLEVEL, OFFLOAD, and DSGROUP CFSTRUCT attributes do not have the values that are shown in the CFSTRUCT display, there is a problem with the CFSTRUCT definition. If you notice these problems, see Part 3, *Procedure to resolve incorrectly created CFSTRUCT objects*, for resolution.

__ 43. Repeat the review for the CSQSYSAPPL structure.

__ 44. Confirm that the queue manager is set to tolerate connection loss. From the SDSF log, type:

```
/MQ## DIS QMGR CFCONLOS
```

replacing the ## placeholder with your primary student number. Press the Enter (Ctrl) key. The results should indicate CFCONLOS(TOLERATE).

Information

To use CFCONLOS(TOLERATE), all queue managers in the queue-sharing group should be at IBM MQ 7.1 or later, and OPMODE set to NEWFUNC.

__ 45. Display your queue-sharing group by entering the DIS GROUP command. From the SDSF log panel, type /MQ## DIS GROUP replacing ## with your primary student number. Press the Enter (Ctrl) key.

__ 46. Your results should resemble the display, which uses MQ03 as an example:

```
MQ03 DIS GROUP
 CSQ5100I MQ03 DISPLAY GROUP report ...
 CSQ5102I MQ03 Queue managers in group SG03 196
 --------------------------------------------------------
  Name  Num  Prefix    Status      Ver  DB2   Connection
  ----  ---  --------  ----------  ---  ----  ----------
  MQ03    1  MQ03      ACTIVE      800  DBA1  ACTIVE
 --------------------------------------------------------
  End of queue managers report
```

Section 11:Display coupling facility structure status information

__ 47. Obtain the status that is associated with the application and administrative structures by typing the command that is shown:

```
/MQ## DIS CFSTATUS(*)
```

__ 48. Check that the results resemble the display, which uses MQ05 as an example. You should expect three structures: APPL1, CSQ_ADMIN, and CSQSYSAPPL:

```
MQ05 DIS CFSTATUS(*)
 CSQM293I MQ05 CSQMDRTC 3CFSTATUS FOUND MATCHING
 CSQM205I MQ05 CSQMDRTC  DIS CFSTATUS DETAILS
 CFSTATUS(APPL1)
 TYPE(SUMMARY)
 CFTYPE(APPL)
 STATUS(ACTIVE)
 OFFLDUSE(SMDS)
 SIZEMAX(50176)
 SIZEUSED(1)
 ENTSMAX(16985)
 ENTSUSED(37)
 FAILTIME()
 FAILDATE()
END CFSTATUS DETAILS
 CSQM201I MQ05 CSQMDRTC
 CFSTATUS(CSQ_ADMIN)
 TYPE(SUMMARY)
 CFTYPE(ADMIN)
 STATUS(ACTIVE)
 OFFLDUSE()
 SIZEMAX(20480)
 SIZEUSED(1)
 ENTSMAX(8804)
 ENTSUSED(12)
 FAILTIME()
 FAILDATE()
  END CFSTATUS DETAILS
CFSTATUS(CSQSYSAPPL)
TYPE(SUMMARY)
CFTYPE(APPL)
STATUS(ACTIVE)
OFFLDUSE(SMDS)
SIZEMAX(20480)
SIZEUSED(1)
ENTSMAX(3971)
ENTSUSED(36)
FAILTIME()
FAILDATE()
 END CFSTATUS DETAILS
```

__ 49. Review the type of information that this command displays. The information includes the administrative structure and application structures. The example that is shown is the default form of the command, which shows TYPE(SUMMARY).

Section 12:Display information that is related to the SMDS data sets

__ 50. Proceed to the system log by typing `=s;log` in the command input prompt and press the Enter (Ctrl) key.

__ 51. Display information that indicates whether your queue manager is able to access and open your SMDS data sets by typing the `DIS SMDSCONN` command. This command requires the queue manager name, and the name of the structure:

```
/MQ## DIS SMDSCONN(MQ##) CFSTRUCT(APPL1)
```

__ 52. Your results should resemble the output that is shown. This output uses MQ05 as an example:

```
MQ05 DIS SMDSCONN(MQ05) CFSTRUCT(APPL1)
 ... ...
 CSQM201I MQ05 CSQMDRTC  DIS SMDSCONN DETAILS
 SMDSCONN(MQ05)
 CFSTRUCT(APPL1)
 OPENMODE(UPDATE)
 STATUS(OPEN)
 AVAIL(NORMAL)
 EXPANDST(NORMAL)
  END SMDSCONN DETAILS
```

__ 53. Take note of the type of information that is displayed by using the `SMDSCONN` command.

__ 54. Repeat the command for the SMDS allocated for the `CSQSYSAPPL` structure. This command does not fit in the command input prompt. You need to use either the ISPF MQSC command panel, or the SDSF option by typing a `/` in an SDSF command input prompt and pressing the Enter (Ctrl) key. If using the latter, omit the `/` from the command.

```
/MQ## DIS SMDSCONN(MQ##) CFSTRUCT(CSQSYSAPPL)
```

__ 55. Check that the results are similar to the earlier `DIS SMDSCONN` output.

__ 56. Check the status of the SMDS data sets by typing the command:

```
/MQ## DIS CFSTATUS(*) TYPE(SMDS)
```

__ 57. The results should be similar to the output shown, which uses MQ05 as an example:

```
CSQM201I MQ05 CSQMDRTC
 CFSTATUS(APPL1)            <=== APPL1 structure SMDS
 TYPE(SMDS)
 SMDS(MQ05)
 STATUS(ACTIVE)
 ACCESS(ENABLED)
 RCVTIME(17.46.32)
 RCVDATE(2015-07-24)
 FAILTIME()
 FAILDATE() END CFSTATUS DETAILS
CSQM201I MQ05 CSQMDRTC  DIS CFSTAT
 CFSTATUS(CSQSYSAPPL)              <=== CSQSYSAPPL SMDS
 TYPE(SMDS)
 SMDS(MQ05)
 STATUS(ACTIVE)
 ACCESS(ENABLED)
 RCVTIME(17.46.32)
 RCVDATE(2015-07-24)
 FAILTIME()                        =
 FAILDATE()
  END CFSTATUS DETAILS
```

__ 58. Contrast the type of information that is obtained in the `DIS SMDSCONN` and the `DIS CFSTATUS TYPE (SMDS)`.

Warning

If you found any problems such as those listed in this box, make any needed corrections now.

- The IBM MQ `CFSTRUCT` is displayed, not showing the expected attributes
- Queue-sharing group-related error messages in the queue manager start output
- Any other questionable results

Do not proceed to the next step if you find any errors.

Section 13:Test: Define a shared queue and allocate the structure by using (putting messages) to the shared queue

__ 59. Define a local shared queue by using the ISPF MQSC command panel.

__ a. Type `=M` in the command input prompt and press the Enter (Ctrl) key.

__ b. Check that the Connect, Target, and Action queue manager names in the ISPF IBM MQ panel are set to your primary queue manager, `MQ##`, where `##` is your primary student number.

__ c. Type an 8 in the Action entry field and press the Enter (Ctrl) key. You are presented with an EDIT panel to enter your `DEFINE QLOCAL` command. If you are not familiar with the use of the MQSC panel, read the directions at the bottom of the panel for navigation instructions.

__ d. Remove any leftover definitions from the MQSC edit panel.

__ e. Type your `DEFINE QLOCAL` command. Use the **+** continuation character if needed to enter multiple lines. Name your queue `TSM00##.SHAREDQ` where `##` is your primary student number. The MQ03 queue manager is used in the example:

Example

```
DEFINE QL(TSM0003.SHAREDQ) CFSTRUCT('APPL1') QSGDISP(SHARED)
```

__ 60. Press PF3 to complete the command. You are presented with a file named `TSM00##.CSQUTIL.OUTPUT`, which contains the MQSC results of the command you provided.

__ 61. Review the results. Make any corrections if needed.

__ 62. Proceed to the SDSF log by typing `=s;log` in the command input prompt and pressing the Enter (Ctrl) key.

__ 63. Display the z/OS coupling facility structure. You need to use the full structure name in the form of `SG##APPL1`. Type the command as shown, substituting your primary student number for the `##` placeholder:

```
/D XCF,STR,STRNAME=SG##APPL1
```

__ 64. Review the results. The status might still show “`NOT ALLOCATED`” and the display might end at “`EXCLUSION LIST IS EMPTY`”, without a usage report below the structure details. If your structure still shows “`NOT ALLOCATED`”, it changes later after you put the first message to the shared queue, which uses the coupling facility for the first time.

Information

You use sample program `CSQ4BCK1` for the `MQPUT`. This program uses the following positional parameters:

`PROGRAM CSQ4BCK1 ISSUES MQPUT TO A QUEUE.`

- `FIRST` is the queue manager name
- `SECOND` is the queue name
- `THIRD` is the THE NUMBER OF MESSAGES TO PUT-(9999)
- `FOURTH` is THE PADDING CHARACTER
- `FIFTH` is THE LENGTH OF EACH MESSAGE-(9999)
- `SIXTH` determines (P)ERSISTENT or (N)ON PERSISTENT

__ 65. Return to the TSM00##.WM31.SCSQPROC PDS and edit member SGMQPUT1.

__ a. Change the positional parameters for the first MQPUT as indicated. The PARM record needs to be set to place 1500 2700-byte persistent messages to queue TSM00##.SHAREDQ. The message contains all As. Your PARM record should resemble the display:

```
PARM=('MQ# TSM00##.SHAREDQ 1500 A 2700 P')
```

__ b. Replace all occurrences of the ## placeholder with your primary student number.

__ c. Ensure that the rest of the parameters look as noted.

__ d. Make note of the job name that is shown in the JOB record, MQ##JPUT, where ## is your primary student number.

__ e. Submit the job and check results. **If there are any errors, resolve them before proceeding with any other work in this exercise**.

Reminder

The structure is allocated when it is first used. The structure is used when the first message is put to a shared queue defined to use the structure. After you submit the job to put messages to the shared queue, look for an IXC prefixed message that indicates the structure is allocated. For example:

```
IXC582I STRUCTURE SG03APPL1 ALLOCATED BY SIZE/RATIOS.
   PHYSICAL STRUCTURE VERSION: CF1F60B1 471FD34C
```

__ 66. Return to the system log by typing =s;log in the command input prompt and pressing the Enter (Ctrl) key.

__ 67. Locate message IXC582I for your structure. If you do not see this message when you first look at the system log, you might need to find previous entries by typing: F IXC582I PREV

Ensure that you look at the message for *your* structure. Other students might be allocating their structures simultaneously.

__ 68. Repeat the display of the structure by typing the z/OS command.

```
/D XCF,STR,STRNAME=SG##APPL1
```

Ensure that you replace the ## placeholder with your primary student number.

__ a. Scroll down if necessary and confirm that you have the display for ***your*** structure.

__ b. The last time that you displayed the structure, the status showed “NOT ALLOCATED” and the display ended at “EXCLUSION LIST IS EMPTY”. Check the status now; it should show “ALLOCATED”.

__ 69. Scroll below the EXCLUSION LIST IS EMPTY line and review the report below header ACTIVE STRUCTURE.

__ a. The ENTRIES and ELEMENTS are displayed under the “SPACE USAGE” header. What are the numbers shown **under the TOTAL column** for ENTRIES and for ELEMENTS?

____

__ b. Check the IBM MQ ENTRY to ELEMENTS ratio. Calculate the result of the division of the figures under the TOTAL column: ELEMENTS/ENTRY. What is your result? It should round off to 6.

____

You use the same SDSF system log view in the next section.

Section 14:Test the offload to the shared message data set

Reminder

Do you remember the SMDS offload rules that you specified in the IBM MQ `CFSTRUCT` object?

For purposes of this course, the rules were customized so that:

- When the structure was 25% full, all messages over 2 K were offloaded:

  `OFFLD1TH(25) OFFLD1SZ(2K)`

- When the structure reached 45% full, all messages over 1 K were offloaded:

  `OFFLD2TH(45) OFFLD2SZ(1K)`

- If the structure reaches 70% full, all messages are offloaded:

  `OFFLD3TH(70) OFFLD3SZ(0K)`

In the last section, you reviewed the percentage of storage used. The usage did not cause any of the rules to become active. You now confirm that no messages were offloaded to SMDS.

__ 70. In the SDSF system log panel, type the command to view SMDS usage, replacing the `##` placeholder with your primary student ID:

```
/MQ## DIS USAGE TYPE(SMDS)
```

If the offload rules do not activate, that is, the structure did not reach the 25% full threshold, there should not be any offloaded messages. An example of the expected output is shown, which uses MQ03:

```
MQ03 DIS USAGE TYPE(SMDS)
CSQE280I MQ03 SMDS usage ... 178
 Application    Offloaded      Total  Total data   Used data  Used
 structure       messages     blocks      blocks      blocks  part
_APPL1                  0       4002        4001           0    0%
```

__ 71. Remember that the SMDS offload rule is set to become active for messages over 2 K after the structure reaches 25% full. Determine the percentage of your structure that is in use by typing the `DIS CFSTATUS` command. Replace the `##` placeholder with your primary student number.

```
/MQ## DIS CFSTATUS(APPL1)
```

__ 72. What is the value shown in the SIZEUSED field for your output?

____________________.

__ 73. It is expected that the SIZEUSED figure in your display is at or below 25%.

__ a. If SIZEUSED has a value in excess of 25%, **stop and determine whether any error messages pertain to your SMDS configuration**.

__ b. If the value is at or below 25%, continue with the next step.

__ 74. Return to the `TSM00##.WM31.SCSQPROC` PDS and edit member `SGMQPUT1`.

__ a. Change the parameters to put 1000 more 2700-byte persistent messages to queue `TSM00##.SHAREDQ` in your primary queue manager. Change the fill character from A to B. The `PARM` record should look similar to the display, replacing the `##` placeholder with your primary student number:

```
PARM=(MQ## TSM00##.SHAREDQ 1000 B 2700 P)
```

__ b. Submit the job and check results.

__ 75. Repeat the SMDS usage display by typing the command: `/MQ## DIS USAGE TYPE(SMDS)`

Ensure that you replace the `##` placeholder with your primary student number.

__ 76. How many messages are offloaded? ____

__ 77. Repeat the `DIS CFSTATUS` command to determine how full your structure is now. What value displayed at the SIZEUSED label?

____

__ 78. Repeat the display of the structure by typing the z/OS command.

```
/D XCF,STR,STRNAME=SG##APPL1
```

Ensure that you replace the `##` placeholder with your primary student number. Your expected results should resemble the display, which uses `SG03APPL1` ***as an example. Obtain the results from your own execution of the command***:

```
ACTIVE STRUCTURE
 ----------------
  ALLOCATION TIME: 06/30/2015 17:37:33
... ... ...
... ... ...
   SPACE USAGE     IN-USE       TOTAL    %
    ENTRIES:         2535       17103   14
    ELEMENTS:       26794      102932   26
    EMCS:              12        7882    0
    LOCKS:                       1024
```

__ 79. Scroll below the `EXCLUSION LIST IS EMPTY` line and review the report below header `ACTIVE STRUCTURE`. How many ENTRIES and ELEMENTS show under the USED column?

____

Note

Do not remove the messages from your TSM00##.SHAREDQ. You use them later in this exercise.

Section 15:Back up the structures that are associated with queue manager MQ##

__ 80. Display the backup status for your SMDS data sets by typing the command show in the display. Replace the ## placeholder with your primary student number.

```
/MQ## DIS CFSTATUS(*) TYPE(BACKUP)
```

__ 81. Review the results. The BKUPSRBA and BKUPERBA are zero, which indicate that no backups are taken. The BKUPSIZE indicates the size of the last backup taken. It also should indicate zero, which means that no previous information was backed up.

Reminder

Other students might be working on the same steps simultaneously.

Ensure that the results you see are for ***your*** queue manager and structures.

__ 82. Issue the command to back up the structures that are associated with your queue manager. Replace the ## placeholder with your primary student number. Be ready to follow the progress of the command in SYSLOG.

```
/MQ## BACKUP CFSTRUCT(*)
```

__ 83. Proceed to the system log and follow the progress of your structure backups. Look for messages that indicate a successful start and end to your backup. The output should resemble the example that is shown, which uses MQ05:

```
CSQE105I MQ05 CSQELRBK BACKUP task initiated for structure APPL1
CSQE105I MQ05 CSQELRBK BACKUP task initiated for structure CSQSYSAPPL
CSQ9022I MQ05 CSQELRBK ' BACKUP CFSTRUCT' NORMAL COMPLETION
CSQESG05MQ0501 disconnected
```

It is possible that your messages might not be contiguous, depending on the activity in the LPAR.

__ 84. Repeat the command to display the backup status and review the results. The command is shown in this step:

```
/MQ## DIS CFSTATUS(*) TYPE(BACKUP)
```

You should have status for APPL1 and CSQSYSAPPL backups, and values in the BKUPSRBA and BKUPERBA fields.

`BKUPSIZE` should also be greater than zero. The value in `BKUPSIZE` is expressed in megabytes, so it might be smaller than expected, but it should be greater than zero.

Note

You completed the first part of this exercise. While you are in the primary TSO session, proceed to create the second SMDS in preparation to incorporate the second queue manager in the queue-sharing group.

Section 16:Create the second SMDS in preparation to add a second queue manager in a separate structure to the queue-sharing group

__ 85. Return to your `TSM00##.WM31.SCSQPROC` PDS and edit member `SGQ4SMDS`.

__ 86. The instructions in the next step assume that the last data set created in `SGQ4SMDS` was for `MQ.MQ##.`**`SYSAPPL`**`.SMDS`. If the JCL shows any name other than SYSAPPL in the third node of the SMDS name, substitute `SYSAPPL` ***in the next step***, with the name that you find in your JCL.

__ 87. Change all occurrences of the shared message data set to have the SMDS for the `APPL2` structure, that is, `MQ.MQ##.APPL2.SMDS`. Replace the string `SYSAPPL` with the string `APPL2`. Check that queue manager name is already set to your primary student number.

__ 88. Carefully scroll through the JCL and check that all occurrences of the SMDS name are correct, considering that one of the IDCAMS data sets has the DATA suffix.

__ 89. Submit the job, and check results. Ensure that the data sets were created and formatted.

Stop

As you proceed to Part 2, do not assume that you repeat all the steps from Part 1. You must follow the steps in Part 2 or expect problems with your configuration.

Proceed to the second part of this exercise.

Part 2: Add a second queue manager to the queue-sharing group and work with messages in the shared queues

Important

Log on with your secondary TSO ID.

The secondary queue manager CFSTRUCT uses coupling facility structure **`SG##APPL2`**.

Section 1: Add your secondary queue manager to the queue-sharing group

__ 1. Proceed to your `TSM00@@.WM31.SCSQPROC` PDS and edit member `SG453AQM`.

__ a. In the substitutions for the variables in this JCL, the placeholders are repeated in the JCL instructions. If you edit by using a **`C`** in the command input line, ensure that you type “change all”, such as `C ## 99 all` so you also change the parameters, not only the JCL comments.

__ b. You use the primary student number to set the queue-sharing group name. Replace **all** occurrences of the `##` placeholder with your primary student number.

__ c. Replace all occurrences of the `@@` placeholder with your secondary student number.

__ d. Replace the `++DSGNAME++` placeholder with the name of the DB2 data sharing group, `DSNDBA0`, while taking care to make the last character the number zero.

__ e. Replace the `++DB2SSID++` placeholder with the name of the `DB2 SSID`. It is: `DBA2`

__ f. Review the command that is included in the `PARM` line and ensure that it is correct. Your parameter should look similar to the example display, which uses queue-sharing group SG04, queue manager MQ24, as an example:

```
PARM='ADD QMGR,MQ24,SG04,DSNDBA0,DBA2'
```

__ g. Scroll to the beginning of the JCL and note the job name: `MQ@@AQM3`

__ 2. Submit the job and check the results. Proceed to the SDSF job display by typing `=s;st` in the command input prompt and pressing the Enter (Ctrl) key.

__ 3. Select your **`MQ@@AQM3`** job and scroll to the end of the display.

__ a. Review the results at the end of the job output.

__ b. **If any errors are found, make necessary corrections before you continue with any other steps**.

__ c. Notice the difference in the messages that are displayed now that the queue-sharing group administrative structure, SG03CSQ_ADMIN, is allocated. MQ23 is used as the example display. Longer output messages are truncated:

```
CSQ5PQSG IBM WebSphere MQ for z/OS V8.0.0
 CSQ5PQSG Queue-sharing Group Utility - 2015-07-09 13:23:36
  ADD QMGR function requested
  Queue manager=MQ23
  QSG=SG03, DB2 DSG=DSNDBA0, DB2 ssid=DBA2
  XCF group CSQGSG03 already defined
  Summary information for XCF group CSQGSG03
  Member=MQ03, state=ACTIVE, system=S101
  User data=E2F1F0F1D4D8F0F340404040C4C2C1F1E00000000505050000010
  QMGR number=1
  Full name of admin structure is SG03CSQ_ADMIN
  Admin structure is defined in CF CF12, allocated size 20480 KB,
  Connected to DB2 DBA1
  QMGR MQ23 entry successfully added to QSG SG03
  Disconnected from DB2 DBA2
 CSQ5PQSG Utility completed, return code=0
```

Application structure SG03APPL2 in the example is still unallocated because it is not used. You confirm that the second structure is not allocated in the next step.

__ 4. Proceed to the SDSF system log view by typing =s;log in the command input prompt and pressing the Enter (Ctrl) key.

__ 5. Display your second application structure by typing /D XCF,STR,STRNAME=SG##APPL2 and replacing the ## placeholder with your primary student number. Press the Enter (Ctrl) key.

__ 6. What is the status of the SG##APPL2 structure from the display? ____________________

Section 2: Create your shared message data sets

Reminder

You need to create the three SMDS data sets, one at a time.

The queue manager name is preconfigured in this JCL. The shared message data set name is partially configured. You complete the customization.

__ 7. Return to your TSM00@@.WM31.SCSQPROC PDS and edit member SGQSMDS2.

__ a. Replace the ++cfstructure++ placeholder, ***or*** the structure name currently in the JCL with the structure name that you use for your secondary queue manager, APPL2.

__ b. Replace all instances of the @@ placeholder with your **secondary** student number. **Ensure that you are using member SGQSMDS2.**

__ c. Note the data set name. You use it when you define the IBM MQ CFSTRUCT object.

__ d. Scroll to the beginning of the JCL and note the job name, **`MQ@@SMD1`**.

__ e. Submit the job and check results. There are two steps to check, the IDCAMS to create the data set, and then the format, CSQJUFMT. The results in the display use queue manager MQ23 as an example. The IDCAMS maximum condition code should be 0, and you should see the CSQJ496I indicating the number of records successfully formatted:

```
IDC0002I IDCAMS PROCESSING COMPLETE. MAXIMUM CONDITION CODE WAS 0
CSQJ440I CSQJUFMT IBM WebSphere MQ for z/OS V8.0.0
CSQJ491I CSQJUFMT Log Data Set Preformatter Utility - 2015-07-09 13:50:56
CSQJ492I Log data set name = MQ.MQ23.APPL2.SMDS
CSQJ496I Log preformat completed successfully, 256140 records formatted
```

__ f. Check that you find two entries; the second entry has the DATA suffix in the name.

__ g. **If your results did not get a return code zero, or you do not find your SMDS entries in the list, resolve the problem before you continue to the next step**.

Repeat to create this secondary queue manager SMDS for the `APPL1` structure:

__ 8. Return to your `TSM00@@.WM31.SCSQPROC` PDS and edit member `SGQSMDS2`.

__ 9. Change all occurrences of the shared message data set to have the SMDS for the APPL1 structure. If your `SGQSMDS2` member still shows APPL2, you can replace all occurrences of APPL2 with APPL1.

__ 10. Carefully scroll through the JCL. Check that all occurrences of the SMDS name are correct, considering that one of the IDCAMS data sets has the DATA suffix, and the data set is specified for the formatting utility.

__ 11. Submit the job, and check results. Ensure that the data sets were created and formatted.

__ 12. If any corrections are needed, make them before you continue with any other steps of this exercise.

Repeat to create the SMDS for structure `CSQSYSAPPL` for the secondary queue manager:

__ 13. Return to your `TSM00@@.WM31.SCSQPROC` PDS and edit member `SGQSMDS2`.

__ 14. If your `SGQSMDS2` member still shows APPL1, you can replace all occurrences of `APPL1` with `SYSAPPL`.

__ 15. Carefully scroll through the JCL. Check that all occurrences of the SMDS name are correct, considering that one of the IDCAMS data sets has the DATA suffix, and the data set is specified for the formatting utility.

__ 16. Submit the job, and check results. Ensure that the data sets were created and formatted.

Important

The check that the file names are created as expected is **critical**.

- First check that the data sets were created similar to the display.

- Next, check that you do not see any data sets named `MQ.`**`MQ`**`##.APPL1.SMDS`, `MQ.`**`MQ`**`##.APPL2.SMDS`, or `MQ.`**`MQ`**`##.SYSAPPL.SMDS` **that use the actual `##` placeholder instead of your queue manager name**.
- If you find any data sets with an actual `##` placeholder in the data set list, **make sure that you did not create it** by going back to your JCL output and checking. The `MQ##` data set might belong to another student, but this check prevents errors that require extra time to check later.

__ a. Type `=3.4` in your ISPF command input line and press the Enter (Ctrl) key to look for your shared message data set. At the Dsname Level entry field, type: `MQ.MQ*`

Do not enter any number portions on the list parameter. Your results should resemble the display output, which uses MQ24 as an example:

```
MQ.MQ24.APPL1.SMDS
MQ.MQ24.APPL1.SMDS.DATA
MQ.MQ24.APPL2.SMDS
MQ.MQ24.APPL2.SMDS.DATA
MQ.MQ24.SYSAPPL.SMDS
MQ.MQ24.SYSAPPL.SMDS.DATA
```

__ b. You do ***not*** want to find any instances of `MQ.MQ##.APPL1.SMDS`, `MQ.MQ##.APPL2.SMDS`, or `MQ.MQ##.SYSAPPL.SMDS`.

__ 17. If any corrections are needed, make them before you continue with any other steps of this exercise.

Section 3: Customize the IBM MQ CFSTRUCT and other IBM MQ queue-sharing group object definitions by using SGQINSS2

__ 18. Edit member `SGQINSS2` at `TSM00@@.WM31.SCSQPROC`.

__ a. `SGQINSS2` is the second updated member and contains new attributes. If you do not find new attributes such as `CFCONLOS` in the IBM MQ `CFSTRUCT` object, ensure that you are updating the correct member, `SGQINSS2`.

__ b. Do a find for string `++cfstructure++` by typing `F ++cfstructure++` in the edit command input line.

__ c. Repeat the find one time by pressing PF6 so that you reach the IBM MQ CFSTRUCT object definition for your secondary queue manager application structure. Your screen should look like the display:

```
DEFINE CFSTRUCT( '++cfstructure++') +
          DESCR( 'System CF structure' ) +
          CFLEVEL( 5 ) +
          OFFLOAD(SMDS) +
          OFFLD1TH(25) +
          OFFLD1SZ(1K) +
          OFFLD2TH(45) +
          OFFLD2SZ(2K) +
          OFFLD3TH(70) +
          OFFLD3SZ(0K) +
          DSGROUP('MQ.*.++cfstructure++.SMDS') +
          DSBLOCK(256K) +
          RECOVER( YES )
```

__ d. The ++cfstructure++ placeholder is used in several places in SGQINSS2. Observe the * in the DSGROUP SMDS name. It is correct; leave as is.

__ e. Observe that the offload percentage and sizes (OFFLDxTH and OFFLDxSZ) parameters are preconfigured, and changed from the initial values to cause messages to overflow to the SMDS sooner for this course. Leave these values as coded in SG04INSS.

__ f. Replace all instances of the ++cfstructure++ placeholder with the application structure that is used for your secondary queue manager. Remember that you are providing this name to IBM MQ, so you use the short name for the structure: APPL2

__ 19. Remove the CFSTRUCT definition for the CSQSYSAPPL structure. This CFSTRUCT object was defined in the first queue manager that was added to the queue-sharing group. **The only item that is needed for the subsequent queue managers to join the group is the separate SMDS that you already created in an earlier step**.

__ 20. Review your SGQINSS2 member definitions and check all queue and CFSTRUCT updates.

Note

Some of the SYSTEM.* objects that were defined as shared for the first queue manager that joined the queue-sharing group might show as duplicate errors when this queue manager starts. These duplications are not a problem.

In the interest of time, and to avoid the possibility of unintended errors, the definitions can be left in SGQINSS2. They result in a duplicate error, which is not a problem for purposes of this course. In an actual organization, these definitions can be removed from the queue manager start procedure.

__ g. Save SGQINSS2. These definitions get created the next time the queue manager starts.

Section 4: Update the secondary queue manager `MQ@@MSTR` procedure

__ 21. Edit member `MQ@@MSTR`, where `@@` is your secondary student number.

__ 22. Find the `STEPLIB` concatenation.

__ 23. Determine whether the DB2 load library is included at the end of the STEPLIB concatenation. If The entry should be:

```
DD DSN=DSNDBA0.DB2B10.SDSNLOAD,DISP=SHR
```

__ 24. If the DB2 data set is not included in the STEPLIB concatenation, add it after data set `SYS2.MQ8000.SCSQAUTH`. If the DB2 data set is included in the STEPLIB concatenation, proceed to the next step.

__ 25. Find `DSN=TSM00@@.WM31.SCSQPROC(CSQ4INYG)` in the `CSQINP2` concatenation.

__ 26. Add your **`SGQINSS2`** member directly before the `CSQ4INYG` data set. The `CSQINP2` concatenation should now look as the partial display:

```
DD DSN=TSM00@@.WM31.SCSQPROC(SGQINSS2),DISP=SHR
DD DSN=TSM00@@.WM31.SCSQPROC(CSQ4INYG),DISP=SHR
```

Warning

The problem that is described in this box should not occur after the first queue manager is added to the queue-sharing group. ***The shared SYSTEM.* queues are defined one time in the first queue manager to join the queue-sharing group***, and should be present. The information in this box is repeated as a precaution.

If you do not have `SGQ4INSS2` ***before*** `CSQ4INYG`, group unit of recovery (`GROUPUR`) does not get enabled in the queue manager.

This error occurs because the `ALTER QMGR GROUPUR(ENABLED)` is done in the `CSQ4IYNG` member before the `SYSTEM.QSG.UR.RESOLUTION.QUEUE` gets defined by the `SGQINSS2` member. IBM MQ checks prerequisites before the `ALTER QMGR` completes, and queue `SYSTEM.QSG.UR.RESOLUTION.QUEUE` is one of the prerequisites checked. In this case, you see message `CSQM507E MQxx CSQMAMMS GROUPUR queue manager attribute was not enabled, CODE=2` in your queue manager started task.

If you saw the `CSQM507E` error, for the lab environment, you can restart the queue manager. Since the queue is now created, `GROUPUR` proceeds to get enabled. In your organization, ensure that your `CSQ4INSS` DD statements are in the correct order in the `CSQINP2` concatenation.

__ 27. Save your changes to `MQ@@MSTR`.

__ 28. Copy the updated `MQ@@MSTR` to replace the member with the same name at `SYS3.STUDENT.PROCLIB` by using the `=3.3` option in the ISPF panels. Remember to place a forward slash by the `Replace like-named members` option.

__ 29. Go to `SYS3.STUDENT.PROCLIB`, and confirm that the updated procedure replaced the old copy of the `MQ@@MSTR` procedure.

STOP **Stop**

Take special care that your `MQ@@MSTR` procedure is indeed copied to **`SYS3.STUDENT.PROCLIB`**. If the updated procedure is not copied to **`SYS3.STUDENT.PROCLIB`**, the queue manager starts without the definitions from the `SGQ4INSS` member. If the queue manager starts without `SGQ4INSS`, it attempts to create a `CFSTRUCT` definition, which is created incorrectly.

This situation might be difficult to resolve.

Section 5: Update, reassemble, and rename the queue manager system parameter module

__ 30. Edit member `SGQ4ZPRM`.

__ 31. Replace all occurrences of the `##` placeholder with your **primary** student ID. This substitution changes the queue-sharing group name.

__ 32. Replace all occurrences of `+DBA+` with the DB2 subsystem ID that you used to add the queue manager to the queue-sharing group, `DBA2`.

__ 33. Scroll down and confirm that all the **`X`** continuation characters line up on the right margin. The job fails if the Xs are not lined up. Carefully make any correction if needed to line up the Xs.

__ 34. Find the `CSQ6SYSP` macro. Check that the `OPMODE` and `QSGDATA` parameters are set as expected:

__ a. `OPMODE` should be set to: `OPMODE=(NEWFUNC,800)`

__ b. `QSGDATA` should reflect the global changes that you made. For example, for a group that uses SSID DBA2 and queue-sharing group `SG03`, `QSGDATA` needs to be set to:

```
QSGDATA=(SG03,DSNDBA0,DBA2,4,4)
```

The parameter string that is shown is an example only; use *your* values for the QSGDATA attribute.

__ c. Confirm again that all the X continuation characters line up on the right margin. The job fails if the Xs are not lined up. Carefully make any correction if needed to line up the Xs.

__ 35. Scroll down to the bottom of the JCL. Notice the new name of the parameter module. After you submit the job, this module is placed in the correct library.

__ 36. Write down the name of the new parameter module if necessary. The renamed version has the parameters that are required to enable your queue manager to be in the queue-sharing group. The name should be in the form: `MQ@@ZQSG`

__ 37. Scroll to the top of the parameter module JCL and note the job name: `MQ@@ZPR1`

Replace `@@` with your secondary student ID.

__ 38. Submit the job and check results. There should not be any error messages, and all condition codes should be zero. Correct any errors before proceeding with further work on this exercise.

Section 6: Enable intra-group queuing

__ 39. Edit member `CSQ4INYG`.

__ 40. Do a find for queue manager attribute IGQ.

__ 41. Change it from `DISABLED` to **`ENABLED`** and press PF3 to save your changes.

__ 42. Check the `GROUPUR` attribute. If it is `DISABLED`, **ensure that you change it to `ENABLED`**.

Section 7: Restart the queue manager with the new parameter module

__ 43. Proceed to SDSF by typing `=s;log` in the command input prompt and pressing the Enter (Ctrl) key.

__ 44. Type `/MQ@@ STOP QMGR` where `@@` is your secondary student number. Confirm that your queue manager stopped by looking at the messages in the system log.

__ 45. Start your queue manager by entering the start command with the new parameter module:

```
/MQ@@ START QMGR PARM(MQ@@ZQSG)
```

STOP **Stop**

All subsequent starts of the queue manager for this course should use the new parameter module. D***o not use MQ@@ZPRM.*** For example, if starting queue manager MQ23, the new start command to use the rest of this course is:

```
/MQ23 START QMGR PARM(MQ23ZQSG)
```

__ 46. Proceed to your `MQ@@MSTR` started task output in the SDSF, typing `=s;da` in the command input prompt and pressing the Enter (Ctrl) key. Select your running **`MQ@@MSTR`** task.

Warning

Do not worry if you see duplicate errors for the SYSTEM.* queues when the queue manager starts. These queues were created as shared queues when you added the first queue manager to the queue-sharing group and shows up as duplicate errors. The PDS members were left to resemble the original CSQ4INSS. In your site, you can remove the definitions that are not needed.

__ 47. Review the messages that are displayed as the `MQ@@MSTR` region started. There should not be any error messages except for duplicate SYSTEM object definition attempts.

__ 48. Display your IBM MQ `CFSTRUCT` object for your `MQ@@` queue manager to ensure that the application structure definition completed correctly.

__ a. Go to the system log by typing `=s;log` in the command input prompt and pressing the Enter (Ctrl) key.

__ b. Type `/MQ@@ DIS CFSTRUCT(APPL2)` and press the Enter (Ctrl) key.

__ c. Confirm that the IBM MQ application structure APPL2 is defined as expected. You should see the new attributes. For example, for queue manager MQ03 the output should resemble the display:

```
CSQM201I MQ03 CSQMDRTC  DIS CFSTRUCT DETAILS
 CFSTRUCT(APPL2)
 DESCR(System CF structure)
 CFLEVEL(5)
 RECOVER(YES)
 OFFLOAD(SMDS)
 OFFLD1TH(25)
 OFFLD1SZ(2K)
 OFFLD2TH(45)
 OFFLD2SZ(1K)
 OFFLD3TH(70)
 OFFLD3SZ(0K)
 DSGROUP(MQ.*.APPL2.SMDS)
 DSBLOCK(256K)
 DSBUFS(100)
 DSEXPAND(YES)
 RECAUTO(YES)
 CFCONLOS(ASQMGR)
... ... ...
```

Some of the attributes you should see are the offload percentages and sizes different from the default values, your shared message data set name in the DSGROUP parameter, and the connection toleration, CFCONLOS, set to ASKQMGR.

STOP **Stop**

If the CFLEVEL, OFFLOAD, and DSGROUP CFSTRUCT attributes do not have the values that are shown in the CFSTRUCT display, there is a problem with the CFSTRUCT definition. If you notice this problem, see Part 3 of this exercise, *Procedure to resolve incorrectly created CFSTRUCT objects*, for resolution.

__ 49. Confirm that the queue manager is set to tolerate connection loss. From the SDSF log, type:

```
/MQ@@ DIS QMGR CFCONLOS
```

replacing the ## placeholder with your primary student number. Press the Enter (Ctrl) key. The results should indicate CFCONLOS(TOLERATE).

Information

To use CFCONLOS(TOLERATE), all queue managers in the queue-sharing group should be at IBM MQ 7.1 or later, and OPMODE set to NEWFUNC.

__ 50. Display your queue-sharing group by entering the `DIS GROUP` command. From the SDSF log panel, type `/MQ@@ DIS GROUP` replacing `@@` with your secondary student number. Press the Enter (Ctrl) key.

__ 51. Your results should resemble the display, which uses `MQ23` as an example:

```
CSQ5102I MQ23 Queue managers in group SG03 095
 --------------------------------------------------------
  Name  Num  Prefix    Status      Ver  DB2   Connection
  ----  ---  --------  ----------  ---  ----  ---------
  MQ03    1  MQ03      ACTIVE      800  DBA1  ACTIVE
  MQ23    2  MQ23      ACTIVE      800  DBA2 ACTIVE
 --------------------------------------------------------
  End of queue managers report
```

__ 52. If you entered command `/MQ@@ DIS CFSTATUS(*) TYPE SMDS` to display the status of the shared message data sets, how many CFSTATUS entries do you expect to see?

Hint

There is one line for each expected CFSTATUS entry in this step.

1. ____
2. ____
3. ____
4. ____

__ 53. Proceed to enter the command `/MQ@@ DIS CFSTATUS(*) TYPE SMDS` and confirm the results.

Warning

If you encountered any problems such as those listed here, make any needed corrections now.

- The IBM MQ `DIS CFSTRUCT` not showing the expected attributes
- Queue-sharing group-related messages in the queue manager start output
- `DISABLED` or other than `AVAILABLE CFSTATUS` for SMDS
- Any other questionable results

Do not proceed to the next step if you find any errors.

Section 8: Test: Define a shared queue and allocate the structure by using (putting messages) to the shared queue

__ 54. Define a local shared queue by using the ISPF MQSC command panel.

__ a. Type `=M` in the command input prompt and press the Enter (Ctrl) key.

__ b. Check that the Connect, Target, and Action queue manager names in the ISPF IBM MQ panel are set to your primary queue manager, `MQ@@`, where `@@` is your secondary student number.

__ c. Type an 8 in the Action entry field and press the Enter (Ctrl) key. You are presented with an EDIT panel to enter your `DEFINE QLOCAL` command. If you are not familiar with the use of the MQSC panel, read the directions at the bottom of the panel for navigation instructions.

__ d. Type your `DEFINE QLOCAL` command. Use the **+** continuation character if needed to enter multiple lines. Name your queue `TSM00@@.SHAREDQ` where `@@` is your primary student number. The `MQ03` queue manager is used in the example:

Example

```
define ql(TSM0023.SHAREDQ) CFSTRUCT('APPL2') QSGDISP(SHARED)
```

__ 55. Press PF3 to complete the command. You are presented with a file named `TSM00@@.CSQUTIL.OUTPUT`, which contains the MQSC results of the command you provided.

__ 56. Review the results. Make any corrections if needed.

__ 57. Proceed to the SDSF log by typing `=s;log` in the command input prompt and pressing the Enter (Ctrl) key.

__ 58. Display the z/OS coupling facility structure. You need to use the full structure name in the form of `SG##APPL2`. Type the command as shown, replacing the `##` placeholder with your primary student number:

```
/D XCF,STR,STRNAME=SG##APPL2
```

__ 59. Review the results. Notice that the status still shows “`NOT ALLOCATED`” and the display ends at “`EXCLUSION LIST IS EMPTY`”, without a usage report below the structure details.

Information

Use sample program CSQ4BCK1. This program uses the following positional parameters:

PROGRAM CSQ4BCK1 ISSUES MQPUT TO A QUEUE.

- FIRST is the queue manager name
- SECOND is the queue name
- THIRD is the THE NUMBER OF MESSAGES TO PUT-(9999)
- FOURTH is THE PADDING CHARACTER
- FIFTH is THE LENGTH OF EACH MESSAGE-(9999)
- SIXTH determines (P)ERSISTENT or (N)ON PERSISTENT

__ 60. Return to the TSM00@@.WM31.SCSQPROC PDS and edit member SGMQPUT2.

__ a. Place 1500 2700-byte persistent messages to queue TSM00@@.SHAREDQ.

__ b. Replace all occurrences of the @@ placeholder with your primary student number. Your PARM statement should resemble the display:

```
PARM=('MQ@@ TSM00@@.SHAREDQ 1500 C 2700 P')
```

__ c. Make note of the job name that is shown in the JOB record, MQ@@JPUT, where @@ is your secondary student number.

__ d. Submit the job and check results. **If there are any errors, resolve them before proceeding with any other work in this exercise**.

Reminder

The structure is allocated when it is first used. The structure is used when the first message is put to a shared queue defined to use the structure. After you submit the job to put messages to the shared queue, look for an IXC prefixed message that indicates the structure is allocated. For example:

```
IXC582I STRUCTURE SG03APPL1 ALLOCATED BY SIZE/RATIOS.
   PHYSICAL STRUCTURE VERSION: CF1F60B1 471FD34C
```

__ 61. Return to the system log by typing =s;log in the command prompt and pressing the Enter (Ctrl) key.

__ 62. Locate message IXC582I for your structure. If you do not see this message when you first look at the system log, you might need to find previous entries by typing: F IXC582I PREV

Ensure that you look at the message for *your* structure. Other students might be allocating their structures simultaneously.

__ 63. Repeat the display of the structure by typing the z/OS command:

```
/D XCF,STR,STRNAME=SG##APPL2
```

Ensure that you replace the ## placeholder with your primary student number.

__ a. Scroll down if necessary and confirm that you have the display for ***your*** structure.

__ b. The last time that you displayed the structure, the status showed `NOT ALLOCATED` and the display ended at `EXCLUSION LIST IS EMPTY`. Check the status now; it should show `ALLOCATED`.

__ 64. Scroll below the `EXCLUSION LIST IS EMPTY` line and review the report below header `ACTIVE STRUCTURE`.

__ a. The ENTRIES and ELEMENTS are displayed under the `SPACE USAGE` header. What are the numbers shown **under the TOTAL column** for ENTRIES and for ELEMENTS?

____

__ b. Check the IBM MQ ENTRY to ELEMENTS ratio. Calculate the result of the division of the figures under the TOTAL column: ELEMENTS/ENTRY. What result do you get?

____

Section 9: Test that SMDS offload is working

__ 65. Return to the SDSF log by typing `=s;log` in the command input prompt and pressing the Enter (Ctrl) key.

__ 66. Determine whether any messages offloaded to SMDS by entering the command:

```
/MQ@@ DIS USAGE TYPE(SMDS)
```

__ 67. If the output shows a number of messages that are offloaded, you completed the SMDS offload step; proceed to the next **numbered** step. If you do not see a number of messages that are offloaded, continue with these substeps:

__ a. If the output does not show any downloaded messages, determine how full the SMDS is by typing the command: `/MQ## DIS CFSTATUS(APPL2)`

__ b. What is the value reported by the `SIZEUSED` attribute? __________________. This value is expected to be approximately `SIZEUSED(18)`.

__ c. Return to your `SCSQPROC` member `SGMQPUT2`, and add 1000 more messages to your `TSM00##.SHAREDQ`. An example `PARM` record, where you replace the `@@` placeholder with your secondary student number, is displayed:

```
PARM=('MQ@@ TSM00@@.SHAREDQ 1000 C 2700 P')
```

__ d. Repeat the `/MQ## DIS CFSTATUS(APPL2)` command. If the `SIZEUSED` is 25 or more, some messages should offload to SMDS.

__ e. Repeat the `/MQ@@ DIS USAGE TYPE(SMDS)` command and confirm that messages are offloaded. If the results of the display show some offloaded messages, **you completed this section of the exercise**.

__ f. If the display does not show any offloaded messages, and `SIZEUSED` is 25 or greater, you might have a problem with the SMDS setup. Resolve the problem before you continue.

Section 10:Back up the structures

__ 68. Back up the structures that were defined with your secondary queue manager by typing the command that is shown. Replace the `@@` placeholder with your secondary student ID.

```
/MQ@@ BACKUP CFSTRUCT(*) CMDSCOPE(MQ@@)
```

__ 69. Follow the progress of the backup in the system log. What structures were backed up?

(three expected): ____.

__ 70. Obtain the status of the backup by typing: `/MQ@@ DIS CFSTATUS(*) TYPE(BACKUP)` Substitute your secondary student number for the `@@` placeholder.

__ 71. Are there values other than zero in attributes `BKUPSIZE`, `BKUPSRBA`, and `BKUPERBA` for structures `APPL2`, `APPL1`, and `CSQSYSAPPL`? ______________________________.

Section 11:Access the messages in the TSM00##.SHAREDQ

In this section, you confirm how each queue manager can access messages from shared queues that are originally defined in other queue managers in the group. Remember, shared queues are defined in DB2, and both queue managers can access the definitions due to the data sharing group.

__ 72. Proceed to the message handler utility by typing `=H` in the command input prompt and pressing the Enter (Ctrl) key.

__ 73. For the Queue Manager Name field, type `MQ@@` and replace the `@@` placeholder with your secondary student ID.

__ 74. For the Queue Name field, type `TSM00##.SHAREDQ` and replace the `##` placeholder with your primary student ID. **The `##` placeholder is correct**; you access the queue that is defined from **`MQ##`** in the first part of this exercise.

__ 75. Press the Enter (Ctrl) key to view the messages. You should see the messages that are put to the `TSM00##.SHAREDQ` queue in Part 1 of this exercise.

You completed the queue-sharing group configuration and verification exercise.

Part 3: Procedure to resolve incorrectly created CFSTRUCT objects

STOP **Stop**

Do not work through this procedure unless you need to correct the CFSTRUCT objects.

This process should work, depending on the situation.

Depending on how far you progressed before you noticed a problem, the application structure might be now allocated, and in use.

__ 1. Determine whether the structure is allocated by typing the `/D XCF,STR,STRNAME=SG??` command that is shown in the first part of this exercise.

__ 2. Go to the ISPF MQSC command input prompt by typing `=M` and pressing the Enter (Ctrl) key.

__ 3. Ensure that Action is 8 and the queue manager names are correct; press Enter (Ctrl) again.

__ 4. Clear any commands that are left in the edit screen.

__ 5. If you defined a queue and placed messages in it, empty the queue of messages by typing the `CLEAR QL` command and pressing the PF3 key.

Reminder

Remember you must use the `QSGDISP(SHARED)` attribute value for any commands other than to display the shared queue, such as:

```
CLEAR QL(TSM00??.SHAREDQ) QSGDISP(SHARED)
```

__ 6. Return to the MQSC edit panel and delete the queue by typing:

```
DELETE QL(TSM00??.SHAREDQ)
```

Use the QSGDISP attribute. Press PF3.

__ 7. Clear the CSQUTIL MQSC file of any commands.

__ 8. Go to your `TSM00??.WM31.SCSQPROC` PDS and open member `SGQ4INSS` or `SGQINSS2`, depending on which queue manager you need to correct.

__ 9. Copy the CFSTRUCT definition that needs correcting by highlighting the required definition and by using the `PCOMM` **Edit > Copy**.

__ 10. Return to your MQSC edit panel, which should be empty, and paste the `CFSTRUCT` command.

__ 11. Change the `DEFINE CFSTRUCT` to: `ALTER CFSTRUCT`

__ 12. Press PF3 to run the `ALTER` command and carefully check results.

Note

If the `ALTER CFSTRUCT` fails because the structure is still allocated, it might be held by the partner queue manager. For example, if you are working to correct a structure in your secondary queue manager, you might need to stop the primary queue manager.

If you are working with an application structure, you might need to type the `RESET CFSTRUCT(????) ACTION(FAIL)` command. Refer to the IBM Knowledge Center bookmark in your VMware Firefox browser.

End of exercise

Exercise review and wrap-up

In this exercise, you:

- Learned to display coupling facility information
- Defined a queue-sharing group to DB2
- Added a queue manager to the queue-sharing group in DB2
- Completed the steps necessary to implement the queue-sharing group in IBM MQ
- Created the IBM MQ `CFSTRUCT` object by incorporating CSQ4INSS to the queue manager procedure
- Defined shared queues
- Put and retrieved messages from the shared queues across both queue managers in the queue-sharing group
- Configured message offload to a shared message data set (SMDS)
- Backed up the structures
- Entered the commands necessary to administer queue-sharing groups

Exercise 5. Intra-group and distributed queuing with queue-sharing groups

What this exercise is about

This exercise provides hands-on practice in the use of shared channels in a distributed queuing environment.

What you should be able to do

After completing this exercise, you should be able to:

- Configure a peer recovery channel scenario from a queue manager outside the queue-sharing group to the queue-sharing group
- Start the generic port listener for the shared channel
- Start the shared channel and check its status
- Test a channel recovery configuration
- Incorporate a queue-sharing group into an existing cluster
- Use intra-group queuing to put messages to a clustered queue
- Determine the route that a message takes in a combined cluster-queue-sharing group environment

Introduction

This exercise focuses on key activities that are related to queue-sharing groups. The first part of the exercise provides the skills that are needed to do the IBM MQ configuration for peer shared channel recovery. Next, you learn how to configure the queue-sharing group to join an existing cluster. In the last part of the exercise, you experience how IBM MQ chooses the best route to route messages in a mixed cluster and queue-sharing group environment.

Requirements

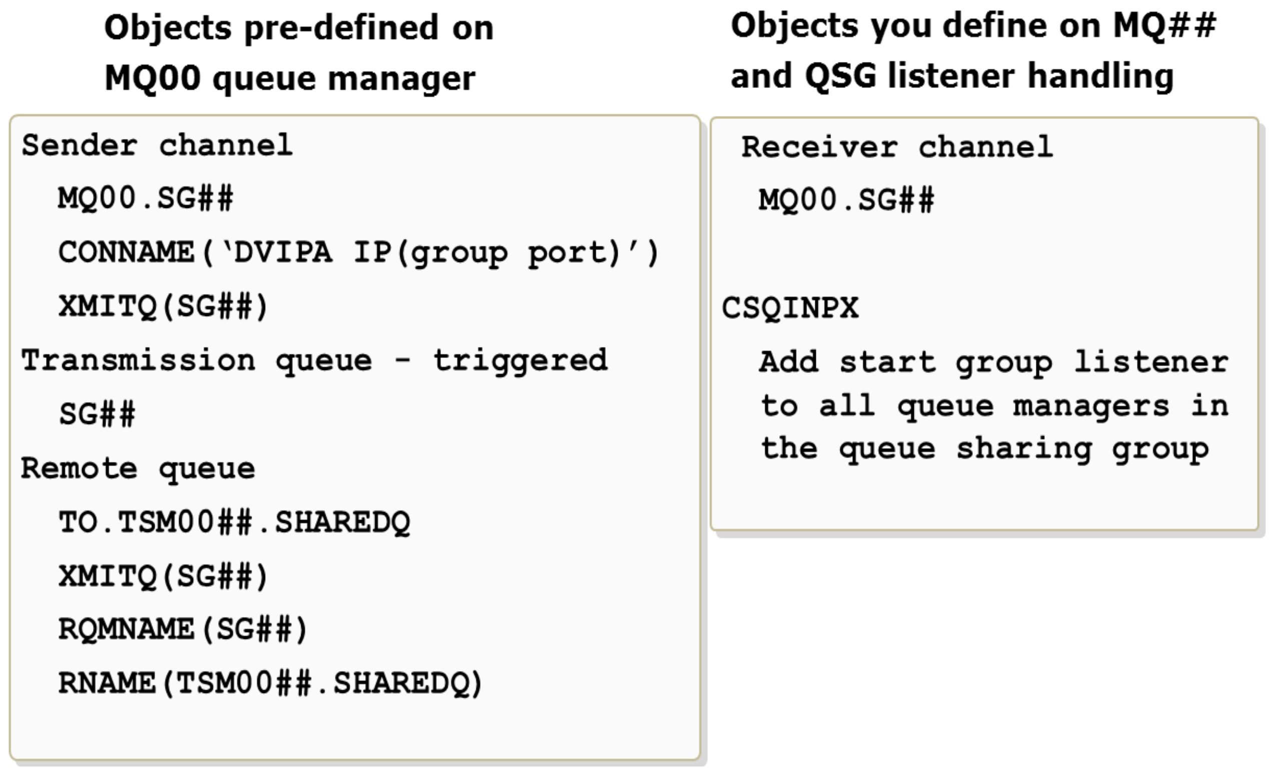
Successful completion of Exercise 4.

Part 1: Configure channels for peer recovery

The ability to do peer recovery of channels is a queue-sharing group feature that promotes high availability. In this section of the exercise, you complete the configuration and test peer recovery.

The IBM MQ sender objects, which include the transmission queue, sender channel, and remote queue, are predefined in instructor queue manager MQ00. You create the receiving channel definition in queue manager **MQ##**, and configure the start of the group listener in all queue managers in your queue-sharing group. You test the peer recovery by alternating stops and starts of the channel initiators in your queue-sharing group.

.



Objects pre-defined on MQ00 queue manager

```
Sender channel
  MQ00.SG##
  CONNAME('DVIPA IP(group port)')
  XMITQ(SG##)
Transmission queue - triggered
  SG##
Remote queue
  TO.TSM00##.SHAREDQ
  XMITQ(SG##)
  RQMNAME(SG##)
  RNAME(TSM00##.SHAREDQ)
```

Objects you define on MQ## and QSG listener handling

```
 Receiver channel
  MQ00.SG##

CSQINPX
  Add start group listener
  to all queue managers in
  the queue sharing group
```

Figure 5-1. Objects for Exercise 5.1

Reminder

The environment for this part of the exercise is configured with a dynamic virtual IP address (DVIPA) for each queue-sharing group. This configuration enables the peer shared channel takeover capability of queue-sharing groups. An example of the DVIPA configuration is provided in the lecture that precedes this exercise.

Important

Log on with your primary TSO ID.

Section 1: Display the predefined objects in queue manager MQ00

__ 1. Your instructor provided a list of virtual IP addresses and ports that are available to each queue-sharing group to set up peer recovery channels. Identify the IP and port that are listed for your queue-sharing group and record this information.

IP:________________________________ Port: ____________________________

You use this information to check the sender channel to the queue-sharing group.

__ 2. Proceed to the ISPF IBM MQ panels MQSC option by typing `=M` in the ISPF command input prompt and pressing the Enter (Ctrl) key.

__ 3. Check the following fields in the IBM MQ ISPF main menu.

Important

The objects that you need to view are defined to MQ00.

Ensure that you set the ISPF IBM MQ panel to `MQ00` as directed for this step. The display does not work for any other queue manager.

__ a. Action = 8.

__ b. Connect name, Target queue manager, and Action queue manager are all `MQ00`.

__ c. Press the Enter (Ctrl) key to proceed to the panel to enter the MQSC display commands.

__ 4. Enter commands to display selected fields for the sender channel, the transmission queue, and the remote queue. A suggested set of display commands is shown. Replace the `##` placeholder with your primary student number.

```
TSM00##.CSQUTIL.COMMANDS
 ***************************** Top of Data ******************
 DIS Q(SG##) TRIGGER INITQ TRIGDATA CURDEPTH
 DIS CHL(MQ00.SG##) CONNAME XMITQ
 DIS QR(TO.TSM00@@.SHAREDQ) XMITQ RQMNAME RNAME
 **************************** Bottom of Data ****************
```

__ 5. Press PF3 to run the MQSC commands and review the output. ***Write the CONNAME and port for your channel `MQ00.SG##`***:

____.

STOP **Stop**

Pay particular attention to the CONNAME attribute IP and port that is used in your `MQ00.SG##` sender channel and write the IP and port in the space provided. This connection uses the dynamic virtual IP (DVIPA) and generic port.

You need this information for the start command for the generic port.

__ 6. Compare this IP to the information provided by the instructor for your class. If this information does not match the list that is provided by your instructor for your queue-sharing group, obtain assistance before proceeding.

__ 7. The expected results are partially displayed for student number 05 as an example. The transmission queue is triggered.

```
QUEUE(SG05)
 TYPE(QLOCAL)
 QSGDISP(QMGR)
 CURDEPTH(0)
 TRIGGER
 INITQ(SYSTEM.CHANNEL.INITQ)
 TRIGDATA(MQ00.SG05)
... ... ...
CHANNEL(MQ00.SG05)
  CHLTYPE(SDR)
  QSGDISP(QMGR)
  XMITQ(SG05)
  CONNAME(10.31.???.???(31??)) <=== Confirm CONNAME IP and Port
 ... ... ...
QUEUE(TO.TSM0005.SHAREDQ)
  TYPE(QREMOTE)
  QSGDISP(QMGR)
  RNAME(TSM0005.SHAREDQ)
  RQMNAME(SG05)
  XMITQ(SG05)
 ... ... ...
 CSQU058I  3 commands issued and responses received, 0 failed
```

Section 2: Define the receiver channel, set up the group listener, and start the channel

__ 8. Return to your ISPF IBM MQ command editor. Clear the display commands, and create the definition for the shared receiver channel. The definition should resemble the display, with your primary student number in place of the `##` placeholder.

Important

If you use IBM MQ ISPF panels to create the channel, ***make sure that you reset the queue manager name to your primary queue manager before you proceed.*** In a recent step, you set the queue manager to MQ00. You must set the three queue manager name input fields in the IBM MQ ISPF main menu to queue manager `MQ##`, where `##` is your primary student number.

```
DEF CHANNEL(MQ00.SG##) CHLTYPE(RCVR) TRPTYPE(TCP) QSGDISP(GROUP)
```

__ 9. If you used the IBM MQ ISPF panels, press PF3 to create the channel definition and review the results.

__ 10. Regardless of the method you used to create the channel, make any corrections *if necessary* before you proceed to the next step.

__ 11. Proceed to edit your `TSM00##.WM31.SCSQPROC` PDS and open member `CSQINPX`.

__ 12. Add two lines after the queue manager `START LISTENER` command and type a second `START LISTENER` command for the generic port. Your command should resemble the example that is given with the substitutions indicated. Replace the IP and PORT attribute `+++` placeholders with the virtual IP and group port you confirmed and saved from your `MQ00.SG##` channel display. When you are done, there are two `START LISTENER` statements in `CSQINPX`.

```
START LSTR TRPTYPE(TCP) PORT(31++) IPADDR(++.++.+++.++) +
   INDISP(GROUP)
```

This command will start the listener in the next restart of the queue manager.

__ 13. Restart the channel initiator. From the SDSF system log panel, type `/MQ## STOP CHINIT` but replace the `##` placeholder with your primary student number. Press the Enter (Ctrl) key.

__ 14. Wait until you see the `CSQ9022I xxx STOP CHINIT NORMAL COMPLETION` message in the log for ***your*** channel initiator.

__ 15. Still in the SDSF system log panel, type `MQ## START CHINIT` but replace the `##` placeholder with your primary student number. Press the Enter (Ctrl) key.

__ 16. Carefully review the results to ensure whether both listeners started. Your output should be similar to the display, which uses MQ04 as an example:

```
+CSQX251I MQ04 CSQXSTRL Listener started, TRPTYPE=TCP INDISP=QMGR
+CSQX023I MQ04 CSQXLSTT Listener started, 299
  port 3104 address *
  TRPTYPE=TCP INDISP=QMGR
+CSQX251I MQ04 CSQXSTRL Listener started, TRPTYPE=TCP INDISP=GROUP
+CSQX023I MQ04 CSQXLSTT Listener started,
  port 3134 address 10.31.188.54
  TRPTYPE=TCP INDISP=GROUP
```

Notice how the disposition (`INDISP`) value for the listener output indicates whether the listener is private to the queue manager, or shared.

Important

Log on, or switch to your secondary TSO session.

You must start the generic listener on your secondary queue manager.

__ 17. Proceed to edit your `TSM00@@.WM31.SCSQPROC` PDS and open member `CSQINPX`.

__ 18. The command that you type is identical to the command entered in `CSQINPX` for the primary queue manager. Add two lines after the queue manager `START LISTENER` command and type a second `START LISTENER` command for the generic port. Your command should resemble the example that is given, but replace the IP and PORT attribute `+++` placeholders with the virtual IP and group port that you confirmed. The result is two `START LISTENER` statements in `CSQINPX`.

```
START LSTR TRPTYPE(TCP) PORT(31++) IPADDR(++.++.+++.++) +
   INDISP(GROUP)
```

This command will start the listener in the next restart of the queue manager.

__ 19. Restart the channel initiator. From the SDSF system log panel, type `/MQ@@ STOP CHINIT` but replace the `@@` placeholder with your secondary student number. Press the Enter (Ctrl) key.

__ 20. Wait until you see the "`CSQ9022I xxx STOP CHINIT NORMAL COMPLETION`" message in the log for ***your*** channel initiator.

__ 21. Remaining in SDSF the system log panel, type `MQ@@ START CHINIT` but replace the `@@` placeholder with your secondary student number. Press the Enter (Ctrl) key.

__ 22. Carefully review the results to ensure that both listeners started.

Important

Return to your primary TSO session.

Reminder

Remember, the sender channel to your queue-sharing group is defined in queue manager MQ00.

You must enter the start command from the MQ00 command prefix string.

__ 23. Start the channel by typing the command that is shown in this step. Replace the ## placeholder with your primary student number:

```
/MQ00 START CHL(MQ00.SG##)
```

__ 24. Press the Enter (Ctrl) key and check the results.

Note

If you completed all previous exercises, an error message is expected. Can you tell what happened? The output of the expected error is displayed, which uses MQ04 as an example.

```
MQ00 START CHL(MQ00.SG04)
 CSQM134I MQ00 CSQMSCHL  START CHL(MQ00.SG04) COMMAND ACCEPTED
 +CSQX500I MQ00 CSQXRCTL Channel MQ00.SG04 started
 CSQX558E MQ00 CSQXRCTL Remote channel MQ00.SG04 not available
 CSQ9023E MQ00 CSQXCRPS ' START CHL' ABNORMAL COMPLETION
 +CSQX558E MQ00 CSQXRCTL Remote channel MQ00.SG04 not available
 +CSQX599E MQ00 CSQXRCTL Channel MQ00.SG04 ended abnormally
 +CSQX777E MQ04 CSQXRESP Channel MQ00.SG04 from 10.31.188.54 325
  (10.31.188.54) has been blocked due to USERSRC(NOACCESS), Detail:
  QMNAME(MQ00)
 +CSQX599E MQ04 CSQXRESP Channel MQ00.SG04 ended abnormally 326
  connection 10.31.188.54.
```

Information

Do you remember the “back-stop rule” in the channel authentication exercise? You need to add a rule to allow the channel to connect. Copy down the IP address from your error message, as you need to use it in your rule. For brevity, a sample template of the rule is provided.

__ 25. Set an ADDRESSMAP channel authentication rule for this connection. Proceed to the IBM MQ ISPF panels by typing =m in the command input prompt. Press Enter (Ctrl). Check that you still have the IBM MQ ISPF panel set to your primary queue manager and Action = 8, and press Enter (Ctrl) to proceed to the MQSC editor.

__ 26. Type the rule from the template in the display, by replacing your ?? placeholders with the IP address you copied from the CSQX777E message. Replace all occurrences of the ## placeholder with your primary student number.

```
SET CHLAUTH(MQ00.SG##) TYPE(ADDRESSMAP) +
ADDRESS('??.??.??.??') MCAUSER(TSM00##)
```

__ 27. Press PF3 to create the rule.

__ 28. Return to the SDSF log view and start the channel by typing the command that is shown in this step. Replace the `##` placeholder with your primary student number:

```
    /MQ00 START CHL(MQ00.SG##)
```

__ 29. Check the results. **The channel should start.** If your channel did not start, resolve any problems before you continue. If after you review the messages you cannot resolve the problem, request assistance if needed.

__ 30. Display the channel status by typing the command that is shown in this step. Replace the `##` placeholder with your primary student number.

Reminder

Remember you must enter this command from `/MQ00`, which owns the sender part of this channel.

```
/MQ00 DIS CHS(MQ00.SG##)
```

__ 31. The results should resemble the display below which uses SG04 as an example. If the channel is in other than `RUNNING` status, correct any problems before you continue to the next step.

```
MQ00 DIS CHS(MQ00.SG04)
 CSQM293I MQ00 CSQMDRTC 1 CHSTATUS FOUND MATCH
 CSQM201I MQ00 CSQMDRTC  DIS CHSTATUS DETAILS
 CHSTATUS(MQ00.SG04)
 CHLDISP(PRIVATE)
 XMITQ(SG04)
 CONNAME(10.31.188.54)
 CURRENT
 CHLTYPE(SDR)
 STATUS(RUNNING)
 SUBSTATE(MQGET)
 STOPREQ(NO)
 RQMNAME(SG04)
```

__ 32. What is the channel disposition (CHLDISP) of this sender side of the channel?

__________________________.

Important

During the remainder of this exercise, ***do not explicitly start the channel again*** unless you explicitly stop it. The channel is triggered. Also, as part of the peer recovery capability of queue-sharing groups, the channel restarts in other available queue managers members of the queue-sharing group.

Section 3: Determine which of the queue managers in the queue-sharing group picked up the receiver side of the channel

Note

Make a special note of this section. You repeat it several times during the peer channel takeover sections of this exercise.

__ 33. Go to the system log by typing `=s;log` in the command input prompt and pressing the Enter (Ctrl) key.

__ 34. From your primary queue manager, display the channel status by typing the command that is shown. Replace the `##` placeholder with your primary student number:

```
/MQ## DIS CHS(MQ00.SG##)
```

__ a. If the command results in a `STATUS(RUNNING)` response, you can proceed to the next section.

__ b. If no status is available, continue to the next step.

Important

Log on to the second LPAR by using your secondary TSO ID.

__ 35. Display the channel status by typing the command that is shown. Replace the `@@` placeholder with your secondary student number:

```
/MQ@@ DIS CHS(MQ00.SG##)
```

If the receiver side of the channel is running in the secondary queue manager, then the sender started the receiver in the secondary queue manager.

Note

You should receive a `RUNNING` channel status when the command to display the status is entered from the MQ00 queue manager *(where the sender channel is)*. If you received the `RUNNING` channel status, it is expected that the shared receiver channel must be active ***in either*** your primary or your secondary queue manager.

If you did not find a status of `RUNNING` for the shared channel from either queue manager in your queue-sharing group, check the system log for error messages. ***Resolve the problem before continuing with subsequent steps***.

Regardless of which queue manager picked up the connection, the channel disposition (CHLDISP) status in the shared queue managers is SHARED.

```
MQ04 DIS CHS(MQ00.SG04)
 ...
 CHSTATUS(MQ00.SG04)
 CHLDISP(SHARED)
 CONNAME(10.31.188.54)
 CURRENT
 CHLTYPE(RCVR)
 STATUS(RUNNING)
 SUBSTATE(RECEIVE)
 STOPREQ(NO)
 RQMNAME(MQ00)
```

If you display the channel definition for both the primary and secondary queue manager, it shows as QSGDISP(COPY). You had to define the channel only one time in a shared queue manager, and a copy was sent to all queue manager members of the queue-sharing group.

Section 4: Clear all messages and display current depth of your TSM00##.SHAREDQ queue

In preparation for the channel takeover test, track the messages that arrive in your TSM00## queue. Clear all messages from the queue before starting the test.

Important

Log on, or switch to, your primary TSO session. Leave your secondary TSO session open.

__ 36. Proceed to the SDSF syslog by typing =s;log in the command input prompt and pressing the Enter (Ctrl) key.

__ 37. Display the number of messages in the TSM00##.SHAREDQ queue by typing the command that is shown. Replace the ## placeholder with your primary student number, and press the Enter (Ctrl) key.

```
/MQ## DIS Q(TSM00##.SHAREDQ) CURDEPTH
```

Your outcome should resemble the display that is shown, which uses MQ05 results as an example:

```
CSQM293I MQ05 CSQMDRTC 1 QUEUE FOUND
 CSQM201I MQ05 CSQMDRTC  DIS QUEUE DET
 QUEUE(TSM0005.SHAREDQ)
 TYPE(QLOCAL)
 QSGDISP(SHARED)
 CURDEPTH(29)
  END QUEUE DETAILS
```

__ 38. Clear the messages from TSM00##.SHAREDQ. Type the command that is shown in the SDSF command input prompt. Substitute your primary student number for the ## placeholder, and press the Enter (Ctrl) key:

```
/MQ## CLEAR QLOCAL(TSM00##.SHAREDQ)
```

Note

If the command was entered as directed, your results should resemble the output that is shown, which uses MQ05 as an example:

```
MQ05 CLEAR QL(TSM0005.SHAREDQ)
 CSQM125I MQ05 CSQMRQLC QLOCAL(TSM0005.SHAREDQ) QSGDISP(PRIVATE)
 NOT FOUND
 CSQM090E MQ05 CSQMRQLC FAILURE REASON CODE X'00D44002'
 CSQ9023E MQ05 CSQMRQLC ' CLEAR QL' ABNORMAL COMPLETION
```

Reminder

Normal MQSC commands are interpreted differently for queue-sharing group objects. When in doubt, check IBM Knowledge Center to determine the defaults and attributes that are required for the different MQSC commands. In the case of the CLEAR QLOCAL command, the default is QSGDISP(PRIVATE). You must provide attribute and value QSGDISP(SHARED) to obtain shared queue results for CLEAR QLOCAL.

__ 39. Repeat the CLEAR QLOCAL command, but this time supply the QSGDISP(GROUP) attribute-value combination. You might need to enter this command from the ISPF IBM MQ command panel to have enough space to include all needed attributes. The correct format for the MQSC command to provide to the ISPF MQSC utility is:

```
CLEAR QL(TSM00##.SHAREDQ) QSGDISP(SHARED)
```

__ 40. Confirm that queue TSM00##.SHAREDQ is empty by repeating the display queue command with the CURDEPTH attribute. You should see CURDEPTH(0).

Section 5: Start the shared channel peer recovery testing; first, with both queue managers running

Information

In the peer recovery sections, you put messages from queue manager MQ00, which is outside the queue-sharing group.

- Your sender channel should be started from the earlier section. `MQ00.SG##` is triggered, so it should start automatically in the future.
- Both queue managers and channel initiators in your queue-sharing group should be started.
- You should have one TSO session open for each of your TSO IDs.

Important

Switch to your primary TSO ID session if not already there.

__ 41. Proceed to edit the `TSM00##.WM31.SCSQPROC` PDS and open member `SGMQPUT1`.

__ 42. Use the remote queue that you displayed in the first section of this exercise to send three messages to the shared queue in your queue-sharing group. The **`PARM`** record in your JCL should resemble the **`PARM`** that is shown in the example for this step. Replace the ## placeholder with your primary student number. You must use MQ00 as the queue manager to handle the MQPUT.

STOP **Stop**

Ensure that you have MQ00 as the first queue manager in the **`PARM`** statement, and that you changed all other parameters, including the queue name. You need to send three smaller messages for this part of the exercise.

```
MQ00 TO.TSM00##.SHAREDQ 0003 A 0100 P
```

__ 43. Submit the `SGMQPUT1` job.

__ 44. Check that the channel is running and that TSM00##.SHAREDQ now contains three messages by running the commands that were used earlier in this exercise.

Reminder

You now should have the following setup:

- The channel initiators for each queue manager in your queue-sharing group are running
- The queue manager and generic listeners started in each channel initiator
- Three messages in the TSM00##.SHAREDQ queue

Section 6: Stop the channel initiator for your primary queue manager and send three more messages

__ 45. From your primary TSO ID session, proceed to the system log by typing `=s;log` in the command input prompt.

__ 46. Stop the channel initiator for your primary queue manager. Type the command that is shown but substitute your primary student ID for the `##` placeholder:

```
/MQ## STOP CHINIT
```

__ 47. Still from your primary TSO ID session, return to edit the `TSM00##.WM31.SCSQPROC` PDS and open member SGMQPUT1. Place three more messages to the remote queue from MQ00 by changing the `PARM` record as shown:

```
MQ00 TO.TSM00##.SHAREDQ 0003 B 0100 P
```

__ 48. Submit the job and check results.

Note

While you work through the steps that follow, wait a few moments for the channel takeover to take place. The start and stop messages might not display on the system logs immediately.

__ 49. Return to the system log on your primary TSO ID session by typing `=s;log` in the command input prompt and pressing the Enter (Ctrl) key.

__ 50. Either scroll or do a find in the system log for messages that are related to your channel. Look for group of messages that resemble the example that is shown in this step, which uses queue-sharing group SG05:

```
+CSQX208E MQ05 CSQXRESP Error receiving data, 42
                 426 00000090   channel MQ00.SG05
                 426 00000090   connection 10.31.185.46
                 426 00000090   (queue manager MQ00)
                426 00000090   TRPTYPE=TCP RC=00000480  reason=00000000
17:44:41.74 STC04049 00000090  +CSQX599E MQ05 CSQXRESP Channel MQ00.SG05
ended abnormally
                  427 00000090   connection 10.31.185.46 …
17:45:22.19 STC03744 00000090  +CSQX500I MQ00 CSQXRCTL Channel MQ00.SG05
started
```

__ 51. Examine your results. You should see:

__ a. A CSQX280E message from your `MQ##` queue manager, followed by a CSQX599E message, which confirms that the channel ended abnormally.

__ b. A CSQX500I from the MQ00 queue manager, which provides information that the channel restarted.

__ c. If your `MQ00.SG##` channel in MQ00 did not restart, look for any errors in the system log for both LPARs, where you are logged on with your primary TSO ID and with your secondary TSO ID. You might also check that your generic listeners started.

Troubleshooting

If you get a RETRYING status on the MQ00 side:

- Check that all listeners start without errors.
- Check your MQ00 sender channel to ensure that the CONNAME is correct. If there are any errors, make corrections before you continue these steps.
- Issue a stop for the MQ00.SG## channel from your primary TSO session.
- Stop your primary queue manager.
- Stop your secondary queue manager.
- Start both queue managers by making sure that you use your **`MQ##ZQSG`** parameter module.
- Start channel MQ00.SG## from your primary queue manager.

If you cannot determine the problem and your channels do not start, contact the instructor for assistance.

Section 7: Check details in the secondary TSO ID session

Important

Switch to the secondary TSO ID session.

__ 52. Confirm that the shared channel in your `MQ@@` queue manager picked up the connection. Display the channel status by typing the command in the display. Replace the `##` placeholder with your primary student number, and the `@@` placeholder with your secondary student number.

```
/MQ@@ DIS CHS(MQ00.SG##)
```

__ 53. What is the channel disposition for this receiving side of the channel? ______________.

The shared receiver channel status is expected to be `RUNNING`. If you do not have an available status, or the status does not show `RUNNING`, resolve any problems before you continue.

__ 54. Confirm the number of messages in the shared queue by entering the command that is shown in this step. Substitute your primary student ID for the `##` placeholder, and your secondary student ID for the `@@` placeholder.

```
/MQ@@ DIS Q(TSM00##.SHAREDQ) CURDEPTH
```

__ 55. The count of messages in the queue should be 6: three messages with both channel initiators up, and three with the `MQ##` channel initiator down. If you do not have six messages, stop and determine where the missing messages are before you continue.

__ 56. Leave your secondary TSO ID session open.

Reminder

You now should have:

- Channel initiator `MQ##` stopped
- Shared receiver channel at `MQ@@` picked up the connection for the `MQ00.SG##` channel
- Six messages in the `TSM00##.SHAREDQ`

Section 8: Restart the channel initiator for the primary queue manager

Important

Switch to the primary TSO ID session.

__ 57. Start the channel initiator for the primary queue manager by entering the `START` command from the system log panel. Replace the `##` placeholder with your primary student number.

```
/MQ## START CHINIT
```

__ 58. Check the system log to confirm:

__ a. The channel initiator started without errors.

__ b. The following set of informational messages were written to the system log for the channel initiator. The example output in the display uses SG05. Your output might not show the messages contiguously; there might be other messages in between.

```
+CSQX120I MQ05 CSQXSPRI Shared channel recovery started for channels
 owned by this queue manager
+CSQX121I MQ05 CSQXSPRI Shared channel recovery started for channels
 owned by other queue managers in the queue-sharing group
+CSQX020I MQ05 CSQXSPRI Shared channel recovery completed
```

__ c. The queue manager and generic listeners started successfully.

__ d. If there are any problems with starting the listeners, resolve them before you continue.

__ 59. Is sender channel MQ00.SG## still running on the MQ00 queue manager? Issue command:

```
/MQ00 DIS CHS(MQ00.SG##)
```

Ensure that you enter the command from queue manager MQ00.

__ 60. What is the MQ00 sender channel status? ____________________.

If you did not take a long break after the last messages were sent, it is expected that the channel is running.

__ 61. If the sender channel is still running, determine which of the shared queue managers is hosting the shared receiver channel after queue manager `MQ##` restarted. This determination is made by entering the `/MQ## DIS CHS(MQ00.SG##)` command from your primary TSO session, or by entering the `/MQ@@ DIS CHS(MQ00.SG##)` command from your secondary TSO session.

Which of the queue managers is hosting the shared receiver channel? __________.

__ 62. If the channel is not running, look at the system log messages and ensure that the channel stopped normally due to inactivity. If the channel stopped due to an error, make corrections before you continue to the next step.

__ 63. Keep your primary TSO session open.

Section 9: Stop the channel initiator for your secondary queue manager

Important

Switch to the secondary TSO ID session.

__ 64. Return to the system log SDSF panel by typing `=s;log` in the command input prompt and pressing the Enter (Ctrl) key.

__ 65. Stop the channel initiator for your secondary queue manager by typing the command that is displayed. Replace the `@@` placeholder with your secondary student number.

```
/MQ@@ STOP CHINIT
```

Important

Return to the primary TSO ID session.

__ 66. Return to edit the `TSM00##.WM31.SCSQPROC` PDS and open member SGMQPUT1. Place three more messages to the remote queue from MQ00 by changing the `PARM` record as shown:

```
MQ00 TO.TSM00##.SHAREDQ 0003 C 0100 P
```

Check that the job completed without errors.

__ 67. Proceed to the system log on the primary TSO ID session.

__ 68. Examine messages for your shared group channel, `MQ00.SG##`. You should see:

__ a. Messages CSQ209E, then CSQ599E from `MQ00`, stating that the connection ended unexpectedly and then ended abnormally.

__ b. Message CSQX500I from queue manager `MQ00`, announcing that the `MQ00.SG##` (sender) channel started.

__ c. Message CSQX500I from queue manager `MQ##`, announcing that the `MQ00.SG##` (shared receiver) channel picked up the channel connection from the “failed” `MQ@@` queue manager.

__ 69. If your results were not as expected, review the system log in the primary TSO session for any possible error messages. Make any needed corrections before you continue.

__ 70. Display the messages in your `TSM00##.SHAREDQ` file by using the `DIS Q` command from the `MQ##` queue manager. Type: `/MQ## DIS Q(TSM00##.SHAREDQ) CURDEPTH`

__ 71. How many messages are in the queue? __________.

__ 72. If you found nine messages, the process completed as expected. If you have fewer than nine messages, stop to determine where the messages are, and resolve any channel problems if needed.

Reminder

You should now have:

- The channel initiator for the secondary queue manager stopped
- Shared receiver channel at `MQ##` picked up the connection for the `MQ00.SG##` channel
- Nine messages in the TSM00##.SHAREDQ

Section 10:Restart the channel initiator for the secondary queue manager

Important

Switch to the secondary TSO ID session.

__ 73. Start the channel initiator for the secondary queue manager by entering the `START` command from the system log panel. Replace the `@@` placeholder with your secondary student number.

```
/MQ@@ START CHINIT
```

Information

Leave your primary TSO ID session open.

You completed Part 1 of exercise *Intra-group and distributed queuing with queue-sharing groups*.

Part 2: Incorporate a queue-sharing group into an existing cluster

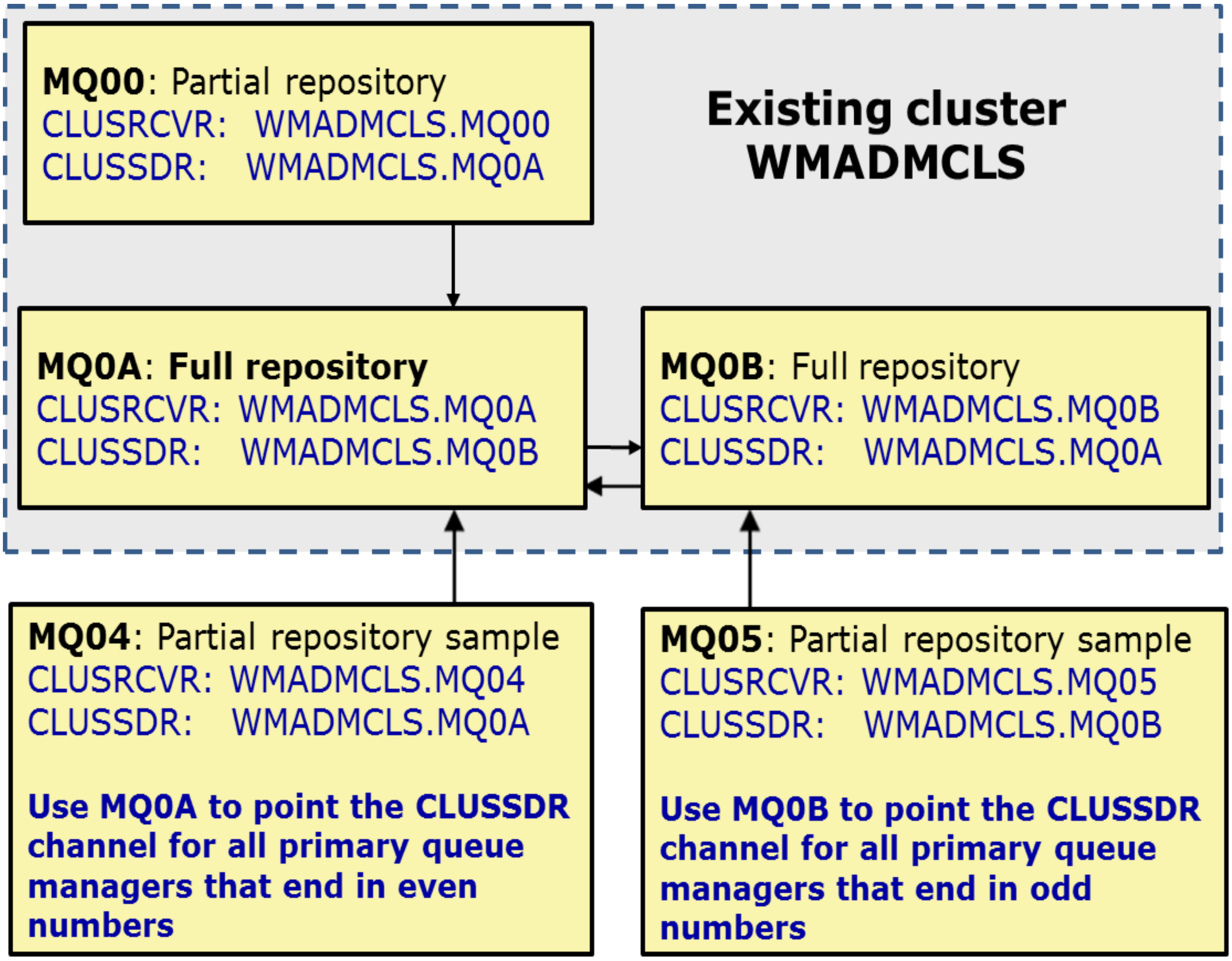


Figure 5-2. WM312 course cluster WM312

The diagram depicts the WM312 lab cluster WMADMCLS. It provides an example of the channel names and full cluster repositories to use when you add your queue-sharing group to the lab cluster. The arrows indicate the full repositories that the different CLUSSDR channels point to.

The two full repositories, MQ0A and MQ0B, point to each other. ***Do not confuse the MQ0A or MQ0B definitions with the directions as to which queue manager you point to.*** Your directions are in the yellow box that applies to your queue-sharing group (even, or odd), not in the full repository yellow boxes.

There are three parts to the definitions required to add a queue-sharing group to an existing cluster.

Important

Review these descriptions carefully as they contain details on the use and naming conventions that are required for the IBM MQ definitions for this part of the exercise.

- Define one CLUSRCVR channel for **each** queue manager member of the queue-sharing group. Follow the naming standards to use the cluster name followed by the queue manager name. By using queue-sharing group SG05 as an example:

  `WMADMCLS.MQ05` is the CLUSRCVR channel name for MQ05.

  `WMADMCLS.MQ25` is the CLUSRCVR channel name for MQ25.

- Define a total of ONE CLUSSDR channel for the queue-sharing group. Use your primary queue manager for the CLUSSDR definition. Use MQ0A as your full repository if your primary queue manager name ends in an even number. Use MQ0B as your full repository if your primary queue manager name ends in an odd number. By looking at even numbered ending and odd numbered ending queue managers, you see that:

  `WMADMCLS.MQ0A` is the ***shared*** CLUSSDR channel name queue managers that end in even numbers such as MQ02, MQ06, and MQ10.

  `WMADMCLS.MQ0B` is the ***shared*** CLUSSDR channel name queue managers that end in odd numbers such as MQ01, MQ05, and MQ09.

- Define one shared local queue (or more later as needed). You define one queue that is named `TSM00##.CLUSSHRQ` from your primary queue manager.

Section 11:Display details of the WMADMCLS cluster and capture information for the IBM MQ object definitions

The diagram at the start of this part of the exercise depicted the architecture of the WMADMCLS cluster as follows:

- MQ0A and MQ0B are the full cluster repository queue managers.
- MQ00 is a partial repository member of the cluster.
- Students use either MQ0A or MQ0B to direct the CLUSSDR channel for their queue-sharing group as follows:
  - Primary, even-numbered queue manager names use MQ0A.
  - Primary, odd-numbered queue manager names use MQ0B.

You display the corresponding MQ0A or MQ0B receiver connection name and record this information to create the CLUSSDR for your queue-sharing group.

Important

Use your primary TSO session for the next set of instructions.

__ 74. ***Regardless whether you have an even or odd queue manager name***, type the command as shown in this step to obtain information about the cluster:

```
/MQ0A DIS CLUSQMGR(*) QMTYPE CONNAME
```

__ 75. Review the resulting display to confirm the correct CONNAME value for your CLUSSDR channel.

Warning

Do not copy the CONNAME information from the sample display because this connection information might be different in your class. Use the LPAR names that your instructor provided.

__ 76. If your primary queue manager name ends in an odd number, copy the information that displays in the CONNAME attribute directly below the CLUSQMGR(MQ0A) header.

__ 77. If your primary queue manager name ends in an even number, copy the information that displays in the CONNAME attribute directly below the CLUSQMGR(MQ0B) header.

```
MQ0A DIS CLUSQMGR(*) QMTYPE CONNAME
 CSQM293I MQ0A CSQMDRTC 3 CLUSQMGR FOUN
 CSQM201I MQ0A CSQMDRTC  DIS CLUSQMGR D
 CLUSQMGR(MQ0A)
 CLUSTER(WMADMCLS)
 CHANNEL(WMADMCLS.MQ0A)
 QMTYPE(REPOS)
 CONNAME(mvsmb11.ilsvpn.ibm.com(1621))
  END CLUSQMGR DETAILS
 CSQM201I MQ0A CSQMDRTC  DIS CLUSQMGR D
 CLUSQMGR(MQ0B)
 CLUSTER(WMADMCLS)
 CHANNEL(WMADMCLS.MQ0B)
 QMTYPE(REPOS)
 CONNAME(mvsmb11.ilsvpn.ibm.com(1622))
  END CLUSQMGR DETAILS
 CSQM201I MQ0A CSQMDRTC  DIS CLUSQMGR D
 CLUSQMGR(MQ00)
 CLUSTER(WMADMCLS)
CHANNEL(WMADMCLS.MQ00)
 QMTYPE(NORMAL)
 CONNAME(mvsmb11.ilsvpn.ibm.com(1600))
  END CLUSQMGR DETAILS
```

__ 78. Write down the CONNAME information that you copied from your command output for use in your CLUSSDR definition:

____.

__ 79. Confirm that the following queues exist in two of the cluster queue managers. You use these queues in a later part of this exercise. You enter the commands to two different queue managers as shown by substituting your primary student ID for the ## placeholder:

```
/MQ0A DIS Q(TSM00##.SHIP) CLUSTER
/MQ0B DIS Q(TSM00##.SHIP) CLUSTER
```

__ 80. Examine results. Queues TSM00##.SHIP should display for each of the MQ0A and MQ0B queue managers as being in the WMADMCLS cluster. If these queues are not there, notify your instructor for resolution.

Section 12:Create a channel authentication rule to allow the cluster channels to work with your primary queue manager

Information

You create a cluster authorization channel of type QMGRMAP to prevent blocked cluster channels to your MQ## queue manager.

Take special care to type all apostrophes for generic values as displayed.

__ 81. Proceed to the IBM MQ ISPF panel by typing =m at a command input prompt and pressing Enter (Ctrl) twice.

__ 82. Discard any definitions from the MQSC edit panel.

__ 83. Type a QMGRMAP type rule by using the template in this step by taking extra care to include all apostrophes as shown. Replace the ## placeholder with your primary student number:

```
SET CHLAUTH('WMADMCLS.*') TYPE(QMGRMAP) ADDRESS('10.31.*') +
DESCR('Access for WMADMCLS cluster channels') +
MCAUSER(TSM00##) QMNAME('MQ*')
```

__ 84. Check that the rule was created successfully. Complete any needed corrections before you continue to the next step.

Section 13:Create the CLUSSDR, CLUSRCVR, and shared queue definition in your shared primary queue manager

A template with the definitions required to add your primary shared queue manager to the WMADMCLS cluster is available to avoid errors in the definition. You copy this file to your ISPF MQSC command file.

__ 85. Proceed to edit the TSM00##.WM31.SCSQPROC PDS and open member SGCLSDF1.

__ 86. Type CAPS OFF at the command input prompt. Although it should not make a difference, type the host names in lowercase.

__ 87. Review the contents of SGCLSDF1 and make the indicated changes by using the notes below the template display. Take particular care:

```
* -------------------------------------------------------
* CLUSTER RECEIVER DEFINITION WHICH HOLDS INFORMATION ABOUT
*    YOUR SHARED PRIMARY QUEUE MANAGER
-------------------------------------------------------
 DEF CHANNEL(WMADMCLS.MQ##) CHLTYPE(CLUSRCVR) TRPTYPE(TCP) +
     CLUSTER(WMADMCLS) CONNAME('+++++++.++++++.+++.+++(31##)')
 * -------------------------------------------------------
 * CLUSTER SENDER DEFINITION. MQ0? SHOULD BE MQ0A FOR EVEN NUMBERED
 *    QUEUE MANAGERS OR MQ0B FOR ODD NUMBERED QUEUE MANAGERS
 * USE THE CONNAME YOU OBTAINED FROM THE DIS CLUSQMGR FOR THIS CHANNEL
 * -------------------------------------------------------
 DEF CHANNEL(WMADMCLS.MQ0?) CHLTYPE(CLUSSDR) TRPTYPE(TCP) +
    CLUSTER(WMADMCLS) CONNAME('+++++++.++++++.+++.+++(++++)') +
    QSGDISP(GROUP)
* -------------------------------------------------------
 * CLUSTERED SHARED QUEUE DEFINITION
 * -------------------------------------------------------
 DEF QLOCAL(TSM00##.CLUSSHRQ) CLUSTER(WMADMCLS) +
    QSGDISP(SHARED) CFSTRUCT(APPL1)
 * -------------------------------------------------------
```

__ a. Replace all occurrences of the `##` placeholder with your primary student number.

__ b. Replace the CLUSRCVR CONNAME attribute value with the **host name and port** for your primary queue manager. (There is no DVIPA; it is `mvsmb++.ilsvpn.ibm.com` where `++` depends on the LPAR1 name that your instructor provided. It is 11 ***or*** 13.

__ c. Replace the `?` in the CLUSSDR name with:

- The letter **`A`** if your primary queue manager name ends with an even number, so channel name is `WMADMCLS.MQ0A`.
- The letter **`B`** if your primary queue manager name ends with an odd number, so channel name is`-***--------------`.

__ d. Substitute the CONNAME attribute value for the CLUSSDR channel with the h**ost name and port that you captured** from the DIS CLUSQMGR command.

__ 88. Save and close member SGCLSDF1 but keep `TSM00##.WM31.SCSQPROC` open for edit.

__ 89. Open PDS member `CSQUTIL`.

__ a. Replace all occurrences of the `##` placeholder with your primary student number.

__ b. Scroll down to the CSQUCMD DD statement and, if needed, change the member name to contain `SGCLSDF1` as the input to the MQSC COMMAND utility.

__ c. Capture the job name for the `CSQUTIL` JCL member: `MQ##UTIL` where `##` is your primary student number.

__ d. Save and submit the `CSQUTIL` job.

__ 90. Proceed to SDSF to review the results by typing `=s;st` in the command input prompt. Press the Enter (Ctrl) key and scroll to locate ***your*** `MQ##UTIL` job output.

__ 91. Review you results:

- __ a. Confirm that the three objects that are defined in your primary shared queue manager were successfully created.
- __ b. Observe the additional work that is generated for the shared objects. Your results should resemble the results that are displayed, which use MQ05 as an example:

```
DEF CHANNEL(WMADMCLS.MQ0B) CHLTYPE(CLUSSDR) TRPTYPE(TCP) +
   CLUSTER(WMADMCLS) CONNAME('MVSMB11.ILSVPN.IBM.COM(1622)') +
   QSGDISP(GROUP) REPLACE
CSQN205I   COUNT=       3, RETURN=00000000, REASON=00000004
CSQM122I MQ05 CSQMACHL ' DEF CHANNEL' COMPLETED FOR QSGDISP(GROUP)
CSQN138I MQ05 'DEFINE CHANNEL' command generated for CMDSCOPE(*), sent to 2
CSQN205I   COUNT=       4, RETURN=00000000, REASON=00000004
CSQN121I MQ05 'DEFINE CHANNEL' command responses from MQ05
CSQM130I MQ05 CSQMACHL CLUSTER REQUEST QUEUED
CSQ9022I MQ05 CSQMACHL ' DEFINE CHANNEL' NORMAL COMPLETION
CSQN205I   COUNT=       4, RETURN=00000000, REASON=00000004
CSQN121I MQ05 'DEFINE CHANNEL' command responses from MQ25
CSQM130I MQ25 CSQMACHL CLUSTER REQUEST QUEUED
CSQ9022I MQ25 CSQMACHL ' DEFINE CHANNEL' NORMAL COMPLETION
CSQN205I   COUNT=       2, RETURN=00000000, REASON=00000000 DEF
QLOCAL(TSM0005.CLUSSHRQ) CLUSTER(WMADMCLS) +
    QSGDISP(SHARED) CFSTRUCT(APPL1)
 CSQN205I   COUNT=       3, RETURN=00000000, REASON=00000004
 CSQM122I MQ05 CSQMAQLC ' DEF QLOCAL' COMPLETED FOR QSGDISP(SHARED)
 CSQN138I MQ05 'ALTER QLOCAL' command generated for CMDSCOPE(*), sent to 2
 CSQN205I   COUNT=       4, RETURN=00000000, REASON=00000004
 CSQN121I MQ05 'ALTER QLOCAL' command responses from MQ05
 CSQM130I MQ05 CSQMAQLC CLUSTER REQUEST QUEUED
 CSQ9022I MQ05 CSQMAQLC ' ALTER QLOCAL' NORMAL COMPLETION
 CSQN205I   COUNT=       4, RETURN=00000000, REASON=00000004
 CSQN121I MQ05 'ALTER QLOCAL' command responses from MQ25
 CSQM130I MQ25 CSQMAQLC CLUSTER REQUEST QUEUED
 CSQ9022I MQ25 CSQMAQLC ' ALTER QLOCAL' NORMAL COMPLETION
 CSQN205I   COUNT=       2, RETURN=00000000, REASON=00000000
```

Information

Message `CSQN138I` reflects that after the shared queue definition was generated in queue manager MQ05, when the object was created on the shared repository, another command was generated. This command was sent to any other active queue manager in the same queue-sharing group.

Section 14:Confirm that the primary shared queue manager successfully joined the WMADMCLS cluster

In this section, you enter a `DIS CLUSQMGR` and a `DIS CHSTATUS` to confirm successful connectivity from your primary shared queue manager to the WMADMCLS cluster.

Take a moment to review the information that reviews what constitutes a valid `DIS CLUSQMGR` command included in the next information box.

Information

When you enter a DIS CLUSQMGR:

- There should ***not*** be any DEFTYPE(CLUSSDR) displayed. CLUSSDR in the DIS CLUSQMGR command indicates a connectivity error. Valid **DEFTYPEs** for DIS CLUSQMGR are:
  - CLUSRCVR
  - CLUSSDRB: If the queue manager from which the command was entered has an explicitly defined CLUSSDR to the full repository queue manager in the corresponding CLUSQMGR value of the display
  - CLUSSDRA: If the queue manager from which the command was entered does not have an explicit CLUSSDR defined to the queue manager in the CLUSQMGR value of the display

When you enter a DIS CHS: CLUSSDR ***is*** a valid CHLTYPE value for the display.

__ 92. Display information about the cluster by entering the `DIS CLUSQMGR` command. Type the command that is displayed by replacing the `##` placeholder with your primary student number.

```
/MQ## DIS CLUSQMGR(*) DEFTYPE STATUS
```

__ 93. Review the results. Your display should resemble the output that is shown *(from the perspective of the queue manager where the command was entered)* by using MQ05 as an example. If you have any errors, stop and make corrections before you continue.

```
MQ05 DIS CLUSQMGR(*) DEFTYPE STATUS XMITQ    <=== Command issued by MQ05
CSQM293I MQ05 CSQMDRTC 3 CLUSQMGR FOUND MATCH
CLUSQMGR(MQ0A)
CLUSTER(WMADMCLS)
CHANNEL(WMADMCLS.MQ0A)
DEFTYPE(CLUSSDRA) <====== MQ05 does not have an explicit CLUSSDR to MQ0A
STATUS(RUNNING) <=====
END CLUSQMGR DETAILS
CLUSQMGR(MQ0B)
CLUSTER(WMADMCLS)
CHANNEL(WMADMCLS.MQ0B)
DEFTYPE(CLUSSDRB) <======  MQ05 has an explicit CLUSSDR to MQ0B
STATUS(RUNNING) <======
END CLUSQMGR DETAILS
CLUSQMGR(MQ05)
 CLUSTER(WMADMCLS)
 CHANNEL(WMADMCLS.MQ05)
 DEFTYPE(CLUSRCVR) <====== This is the CLUSRCVR defined for MQ05 - there
might be multiple connections to this channel, so the status is shown in the
DIS CHS command.
  END CLUSQMGR DETAILS
```

__ 94. Display channel status information for `MQ##` for all channels whose name starts with a `W`, which narrows the display to the cluster channels. Type the command as shown; replace the `##` placeholder with your primary student number.

```
/MQ## DIS CHS(W*)
```

__ 95. Confirm the results of the `DIS CHS` command. A **partial** expected output is displayed by using MQ05 as an example. If your channels are not running or you see any other problem, correct it before you continue.

```
MQ05 DIS CHS(W*) STATUS CHLTYPE RQMNAME
 CSQM293I MQ05 CSQMDRTC 4 CHSTATUS FOUND MATCH
 CHSTATUS(WMADMCLS.MQ0A)
 CHLDISP(PRIVATE)
 CHLTYPE(CLUSSDR)
 STATUS(RUNNING)
 RQMNAME(MQ0A)
  END CHSTATUS DETAILS
 CHSTATUS(WMADMCLS.MQ0B)
 CHLDISP(PRIVATE)
 CHLTYPE(CLUSSDR)
 STATUS(RUNNING)
 RQMNAME(MQ0B)
  END CHSTATUS DETAILS
 CHSTATUS(WMADMCLS.MQ05)
 CHLDISP(PRIVATE)
 CHLTYPE(CLUSRCVR) <=== Two CLUSRCVR for this DIS CHS - one from MQ0A ...
 STATUS(RUNNING)
 RQMNAME(MQ0A)
  END CHSTATUS DETAILS
 CHSTATUS(WMADMCLS.MQ05)
 CHLDISP(PRIVATE)
 CHLTYPE(CLUSRCVR) <=== ... and second CLUSRCVR from MQ0B
 STATUS(RUNNING)
RQMNAME(MQ0B)
  END CHSTATUS DETAILS
```

Note

There might be more CLUSRCVR channels, depending on how many students joined the cluster.

Troubleshooting

If you are having problems with the definitions, such as you see SYSTEM.TEMPQMGR in a display, or see a channel with `RETRYING` status, you can try these steps:

- Issue command `/MQ## DIS CLUSQMGR(*) DEFTYPE CONNAME` to determine the connection attributes you defined.
  - `DEFTYPE CLUSSDRB` display shows the CONNAME from your CLUSSDR definition.
  - `DEFTYPE CLUSRCVR` display shows the CONNAME from your CLUSRCVR definition.
  - **Disregard `DEFTYPE CLUSSDRA`** in this check. **`CLUSSDRA`** is a dynamic, or automatic, channel to your CLUSRCVR from another queue manager member of the cluster. CLUSSDRA channels usually indicate that your CLUSRCVR definition is correct.
- If any definitions are incorrect:
  - Return to your SGCLSDF1 member (or use the ISPF MQSC panel).
  - Make corrections and add a REPLACE attribute.
  - Submit the CSQUTIL job if you used the SGCLSDF1 file.
  - If you get errors that state the channel is running, you might need to stop the channel, then retry the CSQUTIL job or ISPF MQSC definition.
- If you find that the definitions are correct, restart your channel initiator for your primary queue manager, and repeat your tests.

If these steps do not resolve the problem, obtain assistance from your instructor.

Section 15:Put messages to the TSM00##.CLUSSHRQ clustered shared queue from outside the queue-sharing group

Reminder

__ 96. Return to edit the `TSM00##.WM31.SCSQPROC` PDS and open member SGMQPUT1 for edit.

__ a. Set up to send three 100-byte persistent messages to queue `TSM00##.CLUSSHRQ` from queue manager MQ00, which is a member of the WMADMCLS cluster.

__ b. Replace the `##` placeholder with your primary student number.

__ c. Leave the running queue manager name as `MQ00`.

__ d. Ensure that your queue name is `TSM00##.CLUSSHRQ` and that you removed the “`TO.`” prefix from the queue name the last time you used this JCL for the remote queue.

__ e. Check that the `PARM` record in your JCL looks similar to the example displayed, but substitute your primary student ID for the `##` placeholder:

```
PARM=('MQ00 TSM00##.CLUSSHRQ 0003 Z 0100 P')
```

__ 97. Submit the job and check results.

__ 98. Proceed to the system log by typing `=s;log` in the command input prompt and pressing the Enter (Ctrl) key.

__ 99. Display the number of messages in the `TSM00##.CLUSSHRQ` by typing the `DIS Q` command as shown in the display. Replace the `##` placeholder with your primary student number.

```
/MQ## dis q(TSM00##.CLUSSHRQ) CURDEPTH
```

__ 100.The results should display three messages. If you do not have any messages in the queue, resolve any errors before you proceed.

Section 16:Add the second queue manager in your queue-sharing group to the WMADMCLS cluster

Important

Switch to your secondary TSO session.

__ 101.Proceed to edit `TSM00##.WM31.SCSQPROC` PDS and open member SGCLSDF2.

__ 102.A display of SGCLSDF2 is presented. Review the contents of SGCLSDF1 and change as indicated in this step:

```
* -----------------------------------------------------
 * CLUSTER RECEIVER DEFINITION WHICH HOLDS INFORMATION ABOUT
 *    YOUR SHARED SECONDARY QUEUE MANAGER
 * -----------------------------------------------------
 DEF CHANNEL(WMADMCLS.MQ@@) CHLTYPE(CLUSRCVR) TRPTYPE(TCP) +
     CLUSTER(WMADMCLS) CONNAME('+++++++.++++++.+++.+++(31##)')
 * -----------------------------------------------------
```

__ a. Replace the `@@` placeholder with your secondary student ID.

__ b. Replace the `##` placeholder with your primary student ID.

__ c. Replace the `++++` placeholders for the CONNAME attribute host name with the correct host name for your secondary queue manager (this host name is ***not*** a DVIPA IP; it is `mvsmb++. ilsvpn.ibm.com(31##)`, where `++` denotes the LPAR2 number). Depending on the information that the instructor provided, `++` is 12 ***or*** 14.

__ 103.Save and close SGCLSDF2 but keep `TSM00##.WM31.SCSQPROC` open for edit. It is assumed that you have two screens open in each TSO session by using PF2.

__ 104.Open PDS member `CSQUTIL`.

__ a. Replace all occurrences of the `@@` placeholder with your secondary student number.

__ b. Scroll down to the `CSQUCMD DD` statement and change the member name to contain `SGCLSDF2` as the input to the `MQSC COMMAND` utility.

__ c. Capture the job name for the `CSQUTILC` JCL member: `MQ@@UTIL` where `@@` is your secondary student number.

__ d. Save and submit the `CSQUTIL` job.

__ 105. Proceed to SDSF to review the results by typing `=s;st` in the command input prompt. Press the Enter (Ctrl) key and scroll to locate your `MQ@@UTIL` job output. Ensure that your CLUSRCVR channel was successfully created. Make any corrections if needed before you continue.

__ 106. Repeat the `DIS CLUSQMGR` from your secondary queue manager. Type the command as shown, replacing the `@@` placeholder with your secondary student number.

```
/MQ@@ DIS CLUSQMGR(*) DEFTYPE STATUS
```

__ 107. Check the output. Your results might vary depending on activity.

__ a. A status of `INACTIVE` is OK, not a problem.

__ b. **If there are no channels with RETRYING status,** *although not a good practice in a busy queue manager*, you can enter a `/MQ@@ REFRESH CLUSTER(*)`; then, repeat the `DIS CLUSQMGR` to see extra channel activity.

__ c. Look for a CLUSSDRB channel, such as the example shown for MQ25. The CLUSSDR channel was defined as a shared channel in **`MQ##`**. ***For lab environment purposes***, the CLUSSDR points to the MQ0A full repository if it ends in an even number. The CLUSSR points to the MQ0B full repository if it ends in an odd number. The display shows MQ25, which ends in an odd number. Channel `WMADMCLS.MQ0B` shows up as a `CLUSSDRB`, which indicates an explicitly defined CLUSSDR channel to the MQ0B full repository exists for the queue-sharing group:

```
... ... ... ... ...
CSQM201I MQ25 CSQMDRTC  D... ...
 CLUSQMGR(MQ0A)
 CLUSTER(WMADMCLS)
 CHANNEL(WMADMCLS.MQ0A)
 DEFTYPE(CLUSSDRA)
 STATUS(RUNNING)
  END CLUSQMGR DETAILS
 CSQM201I MQ25 CSQMDRTC  D
 CLUSQMGR(MQ0B)
 CLUSTER(WMADMCLS)
 CHANNEL(WMADMCLS.MQ0B)
 DEFTYPE(CLUSSDRB)
 STATUS(RUNNING)
  END CLUSQMGR DETAILS
... ... ...
```

__ 108. Repeat the `DIS CHS(WMADM*)` command. You should see a display similar to `MQ##`.

Section 17:Put messages to clustered queues outside the queue-sharing group

Important

Stay in your secondary TSO session for this section.

In an earlier section of this exercise, you displayed clustered queues `TSM00##.SHIP`, which were defined in clustered queue managers MQ0A and MQ0B. You now put messages to one of the `TSM00##.SHIP` queues from your secondary shared queue manager, `MQ@@`.

__ 109. Proceed to edit the `TSM00@@.WM31.SCSQPROC` PDS and open member `SGMQPUT2`.

__ 110. Put 10 100-byte non-persistent messages to `TSM00##.SHIP`. Update the `PARM` record that is shown by replacing the `@@` placeholder with your secondary student number, and by replacing the `##` placeholder with your primary student number:

```
PARM='MQ@@ TSM00##.SHIP 0010 T 0100 N'
```

The following example uses SG05:

```
PARM='MQ25 TSM0005.SHIP 0010 T 0100 N'
```

__ 111. Submit the job and check that the job completed successfully.

Important

Switch to your primary TSO session.

__ 112. Proceed to the system log by typing `=s;log` in the command input prompt and pressing the Enter (Ctrl) key.

__ 113. Determine where the messages were delivered. The messages are not expected to be balanced between the two queue managers that host `TSM00##`. Issue the `DIS Q` command on both MQ0A and MQ0B by replacing the `##` placeholder with your primary student number:

```
/MQ0A DIS Q(TSM00##.SHIP) CURDEPTH
```

__ 114. If all messages are in the MQ0A instance of this queue, you can proceed to the next section.

__ 115. If the messages were not in the MQ0A queue manager, repeat the display for the MQ0B queue manager:

```
/MQ0B DIS Q(TSM00##.SHIP) CURDEPTH
```

__ 116. If you found your 10 messages, proceed to the next section. If you did not see your messages, determine and correct the problem before you continue.

Section 18:Enable queue monitoring statistics in your primary shared queue manager and system transmission queues

Important

Switch to your primary TSO session. Leave your secondary TSO session open.

__ 117.Enable queue monitoring statistics. Proceed to the ISPF IBM MQ MQSC command processor by typing `=M` in the command input prompt and pressing the Enter (Ctrl) key.

__ 118.Ensure that all three queue manager entries indicate `MQ##`, where `##` is your primary student ID, and the action is 8 for the MQSC command processor.

__ 119.Delete or comment out any remaining commands; then, type the following commands to enable queue statistics:

```
alter ql(SYSTEM.CLUSTER.TRANSMIT.QUEUE) MONQ(HIGH)
alter ql(SYSTEM.QSG.TRANSMIT.QUEUE) QSGDISP(SHARED) MONQ(HIGH)
```

__ 120.Press PF3 to process the commands and check the results in the command output panel. Make any corrections if needed before you continue.

__ 121.Edit member `CSQ4INYG` in your `TSM00##.WM31.SCSQPROC` PDS.

__ a. Scroll to find the `ALTER QMGR` statement.

__ b. Change the value in the `MONQ` attribute to `HIGH`.

__ c. Save and close `CSQ4INYG`.

__ 122.Restart your queue primary queue manager.

__ a. Issue the `/MQ## STOP QMGR` command by replacing the `##` placeholder with your primary student number.

__ b. Wait until you see the `CSQ3100I "READY TO START"` message for your primary queue manager.

__ c. Start your queue manage. Ensure that you use the parameter module that is customized for queue-sharing groups. Type the command per the display, replacing the `##` with your primary student number:

```
/MQ## START QMGR PARM(MQ##ZQSG)
```

Section 19:Enable queue monitoring in your secondary shared queue manager and selected queues

Important

Switch to your secondary TSO session.

__ 123.Repeat all steps from the previous section by substituting all references to your primary student ID to your secondary student ID.

__ 124.Make sure that you:

__ a. Alter `SYSTEM.CLUSTER.TRANSMIT.QUEUE` to have `MONQ(HIGH)`.

__ b. Alter `SYSTEM.QSG.TRANSMIT.QUEUE QSGDISP(SHARED)` to have `MONQ(HIGH)`.

__ c. Update `CSQ4INYG` in your `TSM00@@.WM31.SCSQPROC` PDS.

__ d. Restart your `MQ@@` queue manager by taking care to use the parameter module that is customized for queue-sharing groups in the `MQ@@` start command.

__ 125.Create a new clustered, but not shared, local queue that is called `TSM00@@.CL`, where `@@` is your secondary student ID. You can create this queue from the SDSF syslog panel as displayed (or go to the ISPF IBM MQ panel and include all attributes in this section):

```
/MQ@@ DEF QL(TSM00@@.CL) CLUSTER(WMADMCLS)
```

__ 126.Confirm that the queue got created by entering the `DIS Q` command as indicated in the display:

```
/MQ@@ DIS Q(TSM00@@.CL) CLUSTER
```

You should get confirmation that the queue got created as part of the WMADMCLS cluster.

__ 127.Enable monitoring of `TSM00@@.CL` by typing the `ALTER` command as shown in the display:

```
/MQ@@ ALTER QL(TSM00@@.CL) MONQ(HIGH)
```

Confirm that the change completed successfully.

Section 20:Determine the route that is taken when a message is placed to a clustered queue outside the queue-sharing group

Information

`TSM00@@.CL` is a clustered queue, but not a shared queue.

In this section, you confirm whether IBM MQ uses the `SYSTEM.CLUSTER.TRANSMIT.QUEUE` or the `SYSTEM.QSG.TRANSMIT.QUEUE` to deliver a small message. The message is put in a different queue manager in the cluster, to the clustered queue that you defined in your secondary queue manager.

You use your `MQ##` queue manager to put the message.

Important

Switch to your primary queue manager. Keep your secondary TSO session open.

__ 128.Open your `TSM00##.WM31.SCSQPROC` for edit and select member `SGMQPUT1`.

__ 129.From your primary queue manager, MQ##, place three 100-byte persistent messages to your TSM00@@.CL queue by using a PARM record similar to the display. Replace the ## placeholder with your primary student ID, and the @@ placeholder with your secondary student ID:

```
PARM=('MQ## TSM00@@.CL 0003 C 0100 P')
```

__ 130.Submit the JCL and confirm that it completed successfully.

Important

Switch to your secondary queue manager.

__ 131.Obtain the time stamp when the message was put to the TSM00@@.CL queue by typing the command in the display. Replace the @@ placeholder with your secondary student number:

```
/MQ@@ DIS QS(TSM00@@.CL) MONITOR
```

__ 132.Check your results. Your output should resemble the display, which uses queue-sharing group SG05 as an example:

```
MQ25 DIS QS(TSM0025.CL) MONITOR
 CSQM293I MQ25 CSQMDRTC 1 QSTAT
 CSQM201I MQ25 CSQMDRTC  DIS QS
 QSTATUS(TSM0025.CL)
 TYPE(QUEUE)
 MONQ(HIGH)
 QTIME(999999999,999999999)
 MSGAGE(14323)
 LPUTDATE(2015-08-11)
 LPUTTIME(13.07.38)
 LGETDATE(2015-08-11)
 LGETTIME(10.06.55)
 QSGDISP(QMGR)
  END QSTATUS DETAILS
```

__ 133.Record the ***exact*** time that the messages were put to the TSM00@@.CL queue:

____.

__ 134.Issue the DIS QS command for SYSTEM.QSG.TRANSMIT.QUEUE and for SYSTEM.CLUSTER.TRANSMIT.QUEUE:

```
/MQ@@ DIS QS(SYSTEM.QSG.TR*) MONITOR
/MQ@@ DIS QS(SYSTEM.CLUSTER.TR*) MONITOR
```

__ 135. Check the LGETTIME for these displays. Sample output for queue-sharing group SG05 is displayed:

```
MQ25 DIS QS(SYSTEM.QSG.T*) MONITOR
  CSQM293I MQ25 CSQMDRTC 1 QSTATUS FOUND MATC
  CSQM201I MQ25 CSQMDRTC  DIS QSTATUS DETAILS
  QSTATUS(SYSTEM.QSG.TRANSMIT.QUEUE)
  TYPE(QUEUE)
  MONQ(HIGH)
  QTIME(5492,5486)
  MSGAGE(0)
  LPUTDATE()
  LPUTTIME()
  LGETDATE(2015-08-11)
  LGETTIME(13.07.38) <=== matches the time stamp of the QSG transmit queue.
  QSGDISP(SHARED)
 ... ... ...
  MQ25 DIS QS(SYSTEM.CLUSTER.TR*) MONITOR
  CSQM293I MQ25 CSQMDRTC 1 QSTATUS FOUND MA
  CSQM201I MQ25 CSQMDRTC  DIS QSTATUS DETAI
  QSTATUS(SYSTEM.CLUSTER.TRANSMIT.QUEUE)
  TYPE(QUEUE)
  MONQ(HIGH)
  QTIME(1710,2764)
  MSGAGE(0)
  LPUTDATE(2015-08-11)
  LPUTTIME(11.07.53)
  LGETDATE(2015-08-11)
  LGETTIME(11.07.53)
  QSGDISP(QMGR)
  END QSTATUS DETAILS
```

__ 136. Which transmission queue shows a queue status `LGETTIME` time stamp that matches the `LPUTTIME` you captured from your `TSM00@@.CL` queue? Write the transmission queue name and the `LGETTIME`.____.

__ 137. If you did not obtain any matching statistics, go back through the MONQ setup; ***do not continue with subsequent steps.*** Correct any details, and repeat the exercise. If needed, request help from your instructor.

__ 138. Display the number of messages in your `TSM00@@.CL` queue by typing:

```
/MQ@@ dis q(TSM00@@.CL) CURDEPTH
```

Press the Enter (Ctrl) key, and write down the number of messages: ___________.

There should be three messages.

__ 139. Stop the channel initiator on your secondary queue manager by typing `/MQ@@ STOP CHINIT` and pressing the Enter (Ctrl) key.

__ 140.Open your `TSM00##.WM31.SCSQPROC` for edit and select member `SGMQPUT1`. ***Do not change any details***. Submit the JCL to put three more messages to the `TSM00@@.CL` queue.

__ 141.Display the number of messages in your `TSM00@@.CL` queue by typing:

```
/MQ@@ dis q(TSM00@@.CL) CURDEPTH
```

Press the Enter (Ctrl) key, and write down the number of messages: __________.

There should be six messages.

Note

It is expected that the messages were routed by using the SYSTEM.QSG.TRANSMIT.QUEUE.

This section of the exercise demonstrated how IBM MQ chooses the most efficient route to deliver messages. It also demonstrated how messages reached their destination without the use of the channel initiator.

End of exercise

Exercise review and wrap-up

In this exercise, you obtained hands-on practice with activities related to queue-sharing groups, which included how to:

- Configure a peer recovery channel scenario from a queue manager outside the queue-sharing group to the queue-sharing group
- Start the generic port listener for the shared channel
- Start the shared channel and check its status
- Test a channel recovery configuration
- Incorporate a queue-sharing group into an existing cluster
- Use intra-group queuing to put messages to a clustered queue
- Determine the route that a message takes in a combined cluster-queue-sharing group environment

Exercise 6. Implementing 8-byte relative byte address and buffers above the 2-GB line

What this exercise is about

In this exercise, you implement the IBM MQ V8 for z/OS features.

What you should be able to do

After completing this exercise, you should be able to:

- Implement 8-byte RBA for queue managers that are members of a queue-sharing group
- Configure buffers above the 2-GB line

Introduction

In this exercise, you implement 8-byte RBA for the two queue managers in your queue-sharing group.

Requirements

This exercise uses:

- The two queue managers members of the queue-sharing group that is configured in this course.
- The existing, corresponding BSDS data sets used in the exercises.

Part 1: Implementing 8-byte RBA

This exercise uses IDCAMS JCL and the CSQJUCNV utility. A summary of the JCL members that are used, in the order that is used, is included in the table.

Table 5: JCL members that are used in Exercise 6.

JCL name and utility	Description
CVTBSDS1 - IDCAMS	DEFINE CLUSTER for BSDS V1BK backup
CVTBSDS2 - IDCAMS	DEFINE CLUSTER for BSDS NEW to contain converted, version 2 BSDS
CVTBSDS3 - IDCAMS	REPRO of active (current) to V1BK (backup)
CSQ4BCNV - CSQJUCNV utility	Convert active version 1 BSDS to NEW version 2 BSDS
CVTBSDS5 - IDCAMS	ALTER, to rename active BSDS to OLD BSDS
CVTBSDS6 - IDCAMS	ALTER, to rename NEW BSDS to active BSDS

Important

Log on to your primary TSO session.

Section 1: Define the BKP and the NEW BSDS data sets for the primary queue manager

__ 1. Proceed to your TSM00##.WM31.SCSQPROC data set, where ## represents your primary student number, and open member CVTBSDS1 for edit.

Note

Although you might see a TSM00##.MQ##.BSDS1 and TSM00##.MQ##.BSDS2 defined in the course environment, the queue manager does not use the second copy.

For this exercise, the required conversion is for TSM00##.MQ##.BSDS1. You ***do not*** convert the second BSDS (TSM00##.MQ##.BSDS2). The queue manager does not use it.

__ 2. Replace the ## placeholder with your primary student number.

__ 3. Check that all .DATA and .INDEX components were changed.

__ 4. Submit the job and check results. Make any corrections before you proceed.

__ 5. Open member CVTBSDS1 for edit.

__ 6. Replace the ## placeholder with your primary student number.

__ 7. Check that all .DATA and .INDEX components were changed.

__ a. Type =3.4 in the command input prompt and press the Enter (Ctrl) key.

__ b. Type TSM00##.MQ##.BSDS* in the Dsname level field by replacing the ## placeholder with your primary TSO ID and press the Enter (Ctrl) key.

__ c. You should see the BSDS data sets you defined in the display. An example that shows the expected data sets is shown, which uses MQ04:

Example

```
TSM0004.MQ04.BSDS01
 TSM0004.MQ04.BSDS01.BKP
 TSM0004.MQ04.BSDS01.BKP.DATA
 TSM0004.MQ04.BSDS01.BKP.INDEX
 TSM0004.MQ04.BSDS01.DATA
 TSM0004.MQ04.BSDS01.INDEX
 TSM0004.MQ04.BSDS01.NEW
 TSM0004.MQ04.BSDS01.NEW.DATA
 TSM0004.MQ04.BSDS01.NEW.INDEX
 TSM0004.MQ04.BSDS01.V1BK
 TSM0004.MQ04.BSDS01.V1BK.DATA
 TSM0004.MQ04.BSDS01.V1BK.INDEX
```

Section 2: Stop all the queue managers in the queue-sharing group

__ 8. Proceed to the SDSF log view by typing =s;log in the command input prompt and pressing the Enter (Ctrl) key.

__ 9. Type the command to stop the queue manager as shown in this step by replacing the ## placeholder with your primary student number and pressing the Enter (Ctrl) key.

```
/MQ## STOP QMGR
```

Important

Switch to your secondary TSO session. Leave your primary TSO session open.

The secondary queue manager might stay active while the primary queue manager is converted. Both queue manager stops were included in the same exercise section for convenience.

__ 10. From your secondary TSO ID, stop the secondary queue manager. Proceed to the SDSF log view by typing =s;log in the command input prompt and pressing the Enter (Ctrl) key.

__ 11. Type the command to stop the queue manager as shown in this step by replacing the @@ placeholder with your secondary student number and pressing the Enter (Ctrl) key.

```
/MQ@@ STOP QMGR
```

Section 3: Back up the active BSDS for the primary queue manager

Important

Return to your primary TSO session.

*In this lab, the term **active** BSDS refers to the BSDS used in the `MQ##MSTR` procedure.*

__ 12. Proceed to your `TSM00##.WM31.SCSQPROC` data set, where `##` represents your primary student number, and open member `CVTBSDS3` for edit.

__ 13. Change all occurrences of the `##` placeholder to your primary TSO number.

__ 14. Submit the job and check results. If any corrections are needed, make them before you proceed.

Section 4: Convert the primary queue manager BSDS to version 2

__ 15. Return to your `TSM00##.WM31.SCSQPROC` data set, where `##` represents your primary student number, and open member `CSQ4BCNV` for edit.

__ a. Change all occurrences of the `##` placeholder to your primary student number.

__ b. Change the `?` placeholder to 1 to make the `DB2 SSID DBA1`.

__ c. Check that your `PARM` record looks similar to the example shown, which uses MQ04:

Example

```
PARM=('INQSG,SG04,DSNDBA0,DBA1')
```

__ d. Check that the `SYSUT1` and `SYSUT3` statements are as expected; `SYSUT1` shows the active BSDS, and `SYSUT3` shows the `TSM00##.MQ##.BSDS01.NEW` BSDS.

__ e. Check that STEPLIB includes the DB2 load library, in this case `DSNDBA0.DB2B10.SDSNLOAD`, in the concatenation *(because the queue managers are in a queue-sharing group.)*

__ 16. Submit the job and check results. Make any corrections before you proceed. Your results should resemble the example. Check how the utility connected to the DBA1 SSID:

Example

```
CSQJ445I CSQJUCNV BSDS CONVERSION UTILITY - 2015-10-03 14:49:32
CSQU526I CSQJUCNV Connected to DB2 DBA1
CSQU528I CSQJUCNV Disconnected from DB2 DBA1
CSQJ200I CSQJUCNV UTILITY PROCESSING COMPLETED SUCCESSFULLY
```

Section 5: For the primary queue manager, rename the active BSDS to OLD, and the NEW BSDS to the active BSDS

__ 17. Return to your `TSM00##.WM31.SCSQPROC` data set, where `##` represents your primary student number, and open member `CVTBSDS5` for edit.

__ 18. Change all occurrences of the `##` placeholder to your primary student number.

__ 19. This JCL should rename the active BSDS to the `OLD` BSDS. Check your changes to ensure that all components were correctly changed, and the JCL contains the expected change.

__ 20. Submit the job and check results. Make any corrections before you proceed.

__ 21. Return to your `TSM00##.WM31.SCSQPROC` data set, where `##` represents your primary student number, and open member `CVTBSDS6` for edit.

__ 22. Change all occurrences of the `##` placeholder to your primary student number.

__ 23. This JCL should rename the `OLD` BSDS to the active BSDS. Check your changes to ensure that all components were correctly changed, and the JCL contains the expected change.

__ 24. Submit the job and check results. Make any corrections before you proceed

Section 6: Start the (converted) primary queue manager

Important

Complete the next steps from the primary TSO session.

__ 25. If you completed the SSL unit before this exercise, the `OPMODE` in your primary queue manager should be set to (`NEWFUNC,800`). If you did not follow the order of the labs for this course, check that `OPMODE` is set correctly ***before you start the queue manager***, by entering the `/MQ## DIS SYSTEM` command. Then, find the value reported for `OPMODE`.

__ 26. Enter the command to start the queue manager. Ensure that you use the `MQ##ZQSG` parameter module. The command is shown in the example; replace the `##` placeholder with your primary student number.

```
/MQ## START QMGR PARM(MQ##ZQSG)
```

__ 27. Check the messages as your primary queue manager starts, and look for the `CSQJ034I`. If your queue manager started and you found the `CSQJ034I` message, you successfully converted the primary queue manager BSDS to a version 2 BSDS.

Example

```
CSQJ034I MQ04 CSQJW007 END OF LOG RBA RANGE IS FFFFFFFFFFFFFF
```

__ 28. If your queue manager did not start and resulted in a memory dump, check your trail of work and the queue manager parameter module. Some checks that you can do if the queue manager fails to start are as follows:

__ a. In your queue manager failed started task, do a find for string `OPMODE`. Ensure that it is set to `(NEWFUNC,800)`. If it is not, then edit member `SGQ4ZPRM`, change the `OPMODE` taking care to keep all Xs in line, and reassemble. ***Do not use CSQ4ZPRM***; you must use **`SGQ4ZPRM`**, which contains the queue-sharing group configuration.

__ b. Examine each step, in particular the BSDS data set names that are used in each job.

__ c. Recheck the results of the conversion and renames.

If after you complete the first few checks you still have problems, contact the instructor for assistance.

Section 7: Define the BKP and the NEW BSDS data sets for the secondary queue manager

STOP **Stop**

There is an exception to the `##` placeholder substitution conventions that are used in the course. After Section 7, `##` is replaced with the ***secondary*** student number, as noted in the steps.

Take special care to replace `##` as noted, with the ***secondary*** student number, ***not*** the primary student number.

Important

Log on or switch to your secondary TSO session.

__ 29. Proceed to your `TSM00@@.WM31.SCSQPROC` data set, where `@@` represents your secondary student number, and open member `CVTBSDS1` for edit.

__ 30. Replace the `##` placeholder with your ***secondary*** student number.

__ 31. Check that all `.DATA` and `.INDEX` components were changed.

__ 32. Submit the job and check results. Make any corrections before you proceed.

__ 33. Open member `CVTBSDS1` for edit.

__ 34. Replace the `##` placeholder with your ***secondary*** student number.

__ 35. Check that all `.DATA` and `.INDEX` components were changed:

- __ a. Type `=3.4` in the command input prompt and press the Enter (Ctrl) key.
- __ b. Type `TSM00##.MQ##.BSDS*` in the `Dsname` level field by replacing the `##` placeholder with your ***secondary*** student number and press the Enter (Ctrl) key.

__ 36. You should see the `V1BK` and `NEW` BSDS data sets that you defined in the display. You can safely disregard any BSDS2 data set; it is not used.

Section 8: Back up the active BSDS for the secondary queue manager

Important

Return to your secondary TSO session

*In this lab, the term **active** BSDS refers to the BSDS used in the `MQ@@MSTR` procedure.*

__ 37. Proceed to your `TSM00@@.WM31.SCSQPROC` data set, where `@@` represents your secondary student number, and open member `CVTBSDS3` for edit.

__ 38. Change all occurrences of the `##` placeholder to your ***secondary*** TSO number.

__ 39. Submit the job and check results. If any corrections are needed, make them before you proceed.

Section 9: Convert the secondary queue manager BSDS to version 2

__ 40. Return to your `TSM00@@.WM31.SCSQPROC` data set, where `@@` represents your secondary student number, and open member `CSQ4BCNV` for edit.

- __ a. Change all occurrences of the `##` placeholder to your ***secondary*** student number.
- __ b. Change the `?` placeholder to 2 to make the `DB2 SSID` **`DBA2`**.
- __ ***c. Return to the `PARM` statement and ensure that the queue-sharing group name shows your primary student number, instead of the secondary student number. Remember that the queue-sharing group name uses the primary student number.***
- __ d. Check that your **`PARM`** record looks similar to the example shown, which uses MQ24:

Example

```
PARM=('INQSG,SG04,DSNDBA0,DBA2')
```

__ e. Check that the `SYSUT1` and `SYSUT3` statements are as expected; `SYSUT1` shows the active BSDS, and `SYSUT3` shows the `TSM00@@.MQ@@.BSDS01.NEW` BSDS.

__ f. Check that STEPLIB includes the DB2 load library, in this case `DSNDBA0.DB2B10.SDSNLOAD`, in the concatenation *(because the queue managers are in a queue-sharing group.)*

__ 41. Submit the job and check results. Make any corrections before you proceed. A possible common error is listed in this step. A second display with a successful completion is shown in the next step.

Troubleshooting

If you find the message:

```
CSQU514E CSQJUCNV RRSAF function IDENTIFY failed, RC=00F30006
```

It means that your DB2 SSID is incorrect. Ensure the `db2ssid` value in the `PARM` record of the `CSQ4BCNV` JCL member must contain value `DBA2`.

__ 42. The outcome should look similar to the example shown, which uses MQ24. Check how the utility connected to the DBA2 SSID for this member of the queue-sharing group.

Example

```
CSQJ445I CSQJUCNV BSDS CONVERSION UTILITY - 2015-10-03 18:53:
CSQU526I CSQJUCNV Connected to DB2 DBA2
CSQU528I CSQJUCNV Disconnected from DB2 DBA2
CSQJ200I CSQJUCNV UTILITY PROCESSING COMPLETED SUCCESSFULLY
```

Section 10:For the secondary queue manager, rename the active BSDS to OLD, and the NEW BSDS to the active BSDS

__ 43. Return to your `TSM00@@.WM31.SCSQPROC` data set, where `@@` represents your secondary student number, and open member `CVTBSDS5` for edit.

__ 44. Change all occurrences of the `##` placeholder to your ***secondary*** student number.

__ 45. This JCL should rename the active BSDS to the `OLD` BSDS. Check your changes to ensure that all components were correctly changed, and the JCL contains the expected change.

__ 46. Submit the job and check results. Make any corrections before you proceed.

__ 47. Return to your `TSM00@@.WM31.SCSQPROC` data set, where `@@` represents your secondary student number, and open member `CVTBSDS6` for edit

__ 48. Change all occurrences of the `##` placeholder to your ***secondary*** student number.

__ 49. This JCL should rename the `OLD` BSDS to the active BSDS. Check your changes to ensure that all components were correctly changed, and the JCL contains the expected change.

__ 50. Submit the job and check results. Make any corrections before you proceed.

__ 51. Stay logged on to the secondary TSO session.

Section 11:Start both queue managers and check the results of the BSDS conversion

Important

Complete the next steps from the secondary TSO session.

__ 52. **Before you proceed, ensure that you changed the secondary queue manager `OPMODE` parameter to `NEWFUNC,800`.**

__ a. From your secondary TSO session, proceed to the SDSF log panel by typing `=s;log` and pressing the Enter (Ctrl) key.

__ b. Type the command `/MQ@@ DIS SYSTEM` replacing the `@@` placeholder with your secondary student number, and press the Enter (Ctrl) key.

__ c. In the output, locate the `OPMODE` value and ensure that it is set to `(NEWFUNC,800)`. If it is not, then edit member `SGQ4ZPRM`, change the `OPMODE` taking care to keep all Xs in line, and reassemble. ***Do not use `CSQ4ZPRM`***; you must use **`SGQ4ZPRM`**, which contains the queue-sharing group configuration.

__ 53. Enter the command to start the queue manager. Ensure that you use the `MQ@@ZQSG` parameter module. The command is shown in the example; replace the `@@` placeholder with your secondary student number.

```
/MQ@@ START QMGR PARM(MQ@@ZQSG)
```

__ 54. Check the queue manager start messages, and ensure that you find the `CSQJ034I` for your queue manager, followed by the message with the “16 foxes.” An example that uses `MQ24` is shown. If your queue manager started and you found the `CSQJ034I` message, you successfully converted the BSDS to a version 2.

Example

```
CSQJ034I MQ24 CSQJW007 END OF LOG RBA RANGE IS FFFFFFFFFFFFFFFF
```

Section 12:Put persistent messages from each shared queue manager

__ 55. Proceed to your `TSM00##.WM31.SCSQPROC` data set and open member `SGMQPUT1` for edit.

__ 56. Change the `PARM` record with the number of messages and persistence as shown. Replace the `##` placeholder with your primary student number, and submit the job.

```
// PARM=('MQ## TSM00##.SHAREDQ 0010 R 0100 P')
```

__ 57. Check that the job completed correctly and that no errors reported for your job in the system log.

Important

Switch to your secondary TSO session.

__ 58. Proceed to your `TSM00@@.WM31.SCSQPROC` data set and open member `SGMQPUT1` for edit.

__ 59. Change the `PARM` record as shown with the number of messages and persistence as shown. Replace the `@@` placeholder with your secondary student number, and submit the job.

```
// PARM=('MQ@@ TSM00@@.SHAREDQ 0010 R 0100 P')
```

__ 60. Check that the job completed correctly and that no errors reported for your job in the system log.

__ 61. If your queue manager did not start, check your trail of work and the queue manager parameter module. Review the troubleshooting tips documented earlier in this exercise.

You completed the first part of this exercise, implementation of 8-byte RBA. You now configure buffers above the 2 GB bar.

Note

The work with buffers above the bar uses the primary queue manager. The rest of the exercises do not use the secondary queue manager.

You can close your secondary TSO session.

Part 2: Implement buffers above the 2 GB bar

Important

Log on or switch to your primary TSO session.

Note

Information for this part of the exercise is as follows:

- `OPMODE` is at (`NEWFUNC,800`). There is no need to change the parameters.
- `DEFAULT STGCLASS` uses page set 4; mapped to `BUFFPOOL 3`.

Section 1: Display buffer pool use

__ 1. Proceed to the system log by typing `=s;log` in the command input prompt and typing the Enter (Ctrl) key.

__ 2. You can disregard the top section of the output for page sets. Review the buffer pools for your queue manager. The display should look similar to the example shown, which uses `MQ04`. Stay in the system log view.

```
End of page set report
 CSQI065I MQ04 Buffer pool attributes ... 747
   Buffer  Available  Stealable   Stealable  Page      Location
     pool    buffers    buffers  percentage  class
 _      0      50000      49983          99  4KB       BELOW
 _      1      20000      19999          99  4KB       BELOW
 _      2      50000      49995          99  4KB       BELOW
 _      3      20000      18391          91  4KB       BELOW
End of buffer pool attributes
```

Section 2: Move buffer pool 3 above the 2 GB bar

Stop

In consideration of fellow students, refrain from increasing the size of your buffer pools. Also, ensure that you decrease the size of the buffer pool when requested to do so, by typing all commands with the **exact attributes shown**.

__ 3. Move the fourth buffer pool that is **buffer pool ID 3** to be located above the bar, by typing the command that is shown. ***Do not change*** the `PAGECLAS` attribute. Replace the `##` placeholder with your primary student number:

```
MQ## ALTER BUFFPOOL(3) LOCATION(ABOVE)
```

__ 4. Check the results in the log. You should see the message: `CSQP054I MQ04 Buffer pool 4 is now located above the bar.`

__ 5. Confirm the change by using the `DIS USAGE` command. Your response should be similar to the example shown, which uses MQ04. This change took place immediately. In a working environment, ensure that you repeat the change in `CSQINP1` so the change reflects in subsequent queue manager restarts.

Example

```
End of page set report
 CSQI065I MQ04 Buffer pool attributes ... 747
   Buffer  Available  Stealable   Stealable  Page      Location
     pool    buffers    buffers  percentage  class
 _      0      50000      49983          99  4KB       BELOW
 _      1      20000      19999          99  4KB       BELOW
 _      2      50000      49995          99  4KB       BELOW
 _      3      20000      18391          91  4KB       ABOVE
End of buffer pool attributes
```

Section 3: Change a buffer pool LOCATION and PAGECLAS attributes

__ 6. Proceed to the ISPF MQSC panel by typing `=M` and pressing the Enter (Ctrl) key.

__ 7. Ensure that the `Action` field is set to 8, and the connect, target, and action queue managers are set to `MQ##`, where `##` is your primary student number.

__ 8. Press Ctrl (Enter) to proceed to the `TSM00##.CSQUTIL.COMMANDS` edit panel.

__ 9. Delete any leftover commands.

__ 10. Enter the command to alter buffer pool 1, exactly as shown in this step:

```
ALTER BUFFPOOL(1) BUFFERS(100) LOCATION(ABOVE) PAGECLAS(FIXED4KB).
```

__ 11. Check to ensure that the command is correct; then, press PF3 to run the MQSC processor.

__ 12. In the resulting `TSM00##.CSQUTIL.OUTPUT` data set, check the outcome of your command. Make any corrections if necessary before you continue.

__ 13. Proceed to the system log by typing `=s;log` in the `Command` prompt at the bottom of your `TSM00##.CSQUTIL.OUTPUT` screen, and press the Enter (Ctrl) key.

__ 14. Display the buffer pools by typing `/MQ## DIS USAGE` in the system log and pressing the Enter (Ctrl) key.

__ 15. Change the second buffer pool that is BUFFPOOL(1), to a location above the bar, with fixed pages by typing the commands that are shown.

__ 16. Did your Page class change to `PAGECLAS(FIXED4KB)`? ________________________.

__ 17. If your answer is no (expected):

__ a. Find message `CSQP062I` for your queue manager by typing:

`'F CSQP062I MQ##'`

Make sure that you enclose the find in quotation marks. You might need to do a find previous or scroll through the log to find it.

__ b. When you find the message, follow the action that is required, which is to **restart the queue manager**. Remember to always use your `MQ##ZQSG` parameter module.

__ c. Repeat the `/MQ## DIS USAGE` command. Your display should resemble the output that is shown, which uses `MQ04` as an example.

Example

```
End of page set report
CSQI065I MQ04 Buffer pool attributes ... 747
   Buffer  Available  Stealable   Stealable  Page      Location
     pool    buffers    buffers  percentage  class
 _      0      50000      49983          99  4KB       BELOW
 _      1        100         99          99  FIXED4KB  ABOVE
 _      2      50000      49995          99  4KB       BELOW
 _      3      20000      18391          91  4KB       ABOVE
  End of buffer pool attributes
```

You completed Part 2, which concludes this exercise.

End of exercise

Exercise review and wrap-up

In this exercise, you:

- Converted two queue manager members of a queue-sharing group to use the 8-byte RBA
- Moved existing buffers to be above the 2 GB bar without the use of fixed memory pages
- Moved existing buffers to be above the 2 GB bar and use fixed memory pages

Exercise 7. Getting started with IBM MQ statistics and accounting records

What this exercise is about

In this exercise, you learn how to format statistics and accounting information. You then explore how to use this information to check the health of the queue manager and create profiles for capacity planning.

What you should be able to do

After completing this exercise, you should be able to:

- Format an SMF accounting and statistics trace
- Review the information in the various JES output files that MP1B generates
- Identify key fields of information for selected MP1B JES output files
- Use the MP1B and MP16 documentation as a guide to continue your work with accounting and statistics
- Use the MP1B SYSIN data set options to format a trace with extra details
- Format an SMF accounting and statistics trace with the CSQ4SMFD utility

Introduction

In this exercise, you format the SMF information by using MQ1B and use the support pac documentation to check selected fields per output file. You also format the data at a higher detail level for contrast. At the end of the exercise, you format the data with the CSQ4SMFD and compare the difference in the output.

Requirements

Previously collected SMF 215 and 216 records in file `TSM00##.SMF0007`, where ## is your primary student number.

Exercise instructions

Section 1: Use the MP1B utility to format the accounting and statistics records

In this part of the exercise, you format an SMF file that contains previously collected IBM MQ accounting and statistics records.

Important

Log on with your primary TSO ID. Split your screen by using PF2, so you can use PF9 to switch ISPF views.

__ 1. Proceed to your `TSM00##.WM31.SCSQPROC` PDS and open member STATS003 for edit. The `##` placeholder represents your primary student number.

__ 2. This JCL should not need any changes; check that the JOB name includes your primary student number.

Stop

Do not change or add any //SYSIN statements.

Leave //SYSIN with the `* comment *` record. You change //SYSIN later in this exercise.

__ 3. Submit the JCL.

__ 4. Select your job from the SDSF job status panel ***by using a question mark (?)*** so that all JES output files are listed. Press the Enter (Ctrl) key.

__ 5. You should see several JES output files. If your job had any errors, make the necessary corrections before you proceed. Leave this JES output file screen list open.

__ 6. From your Windows VMware image, go to `C:\WM312\documents` and open the MP1B documentation named `mp1b_mqsmf.pdf`. Use the PDF zoom option if necessary.

__ 7. From the PDF document, find the header *`Control what is displayed`* by right-clicking and selecting **Find**; then, type `Control what is displayed` in the find box.

__ 8. Note the first two entries in the *`Control what is displayed`* table:

__ a. `Debug` is the first entry. Note how in the `meaning` column, it is labeled `for IBM use only`. Some of the information that MP1B displays is for the use of the support team.

__ b. `Detail` is the second entry. What is the initial default value for Detail? What is the highest value possible?

____.

__ 9. Return to your TSO session and scroll down towards the end in the list of JES output files and select the SYSPRINT file. The first piece of information that is displayed is the MP1B compile date.

__ a. Does the second SYSPRINT line resemble the value in your STATS001 JCL //SYSIN statement? ____.

__ b. Do a find for string `Detail` by typing `f detail` in the command input prompt and pressing the Enter (Ctrl) key. This record represents the `Detail` value at which this iteration of the MP1B formatting ran. ***Do not use these fields in your first MP1B run***.

__ c. Keep the SYSPRINT file open.

__ 10. By using either the SYSPRINT file or the MP1B documentation, review some of the fields that are of interest to you. These fields can be used to filter the output of MP1B. ***Keep the MP1B documentation open***.

__ 11. Now scroll to the end of the SYSPRINT file by typing an `M` in the command input prompt and pressing the PF8 key.

__ 12. You should see a list of SMF record types that the MP1B utility processes.

Use this information to confirm that the trace collected the expected data.

Warning

You should see a report similar to the display. For this course, a trace was precollected and copied to each student TSO ID. If you do not see information similar to the following sample, contact your instructor to have the trace file correctly copied.

```
Summary of MQ SMF records and subtypes found
============================================
SMF type 115 subtype 1, record count 14 System statistics(1)
SMF type 115 subtype 2, record count 14 System statistics(2)
SMF type 115 subtype 5, record count 7 Storage statistics
SMF type 115 subtype 6, record count 7 Storage detail statistics
SMF type 115 subtype 7, record count 10 Storage summary statistics
SMF type 115 subtype 215, record count 14 Buffer manager extension
SMF type 115 subtype 231, record count 27 Chinit statistics
SMF type 116 subtype 0, record count 31 Accounting class(1)
SMF type 116 subtype 1, record count 89 Accounting class(3)
SMF type 116 subtype 10, record count 6 Channel accounting data
```

Section 2: Check information on number of API calls

__ 13. Close the SYSPRINT file; stay in the file list for MP1B.

__ 14. Find and select file MSGM.

__ 15. In your MP1B documentation, find the section that describes the contents of the MSGM file. You should end up under heading *Message Manager*. The documentation has information on the fields that are formatted for the message manager information.

__ 16. Close the MSGM file, and open MSGMCSV. Contrast with the information for MSGM. MSGMCSV is the CSV file that you can use to build a profile of API calls.

Section 3: Check the logging statistics

__ 17. **Scroll to the start** of your MP1B documentation, and do a find for *log statistics*.

__ a. First, you see a *table of contents* entry. Review the areas where the documentation for MP1B provides guidance.

__ b. From the *table of contents*, click *Log statistics,* which should take you to the section by the same name. Review the documentation until you come to a section titled *Regular checks*. Keep the documentation open to the *Regular checks* page.

__ 18. Close the MSGM file and open the LOG file.

__ 19. Review the LOG file by following the information in the *Regular checks* page; ***skip looking at the LOGCSV data set for now***.

__ 20. Complete working with the MP1B documentation through the end of the section titled *`You think you may have a problem with logging`*.

__ 21. A couple of paragraphs down, find section *`Records produced if detail >= 10`*. The output in the documentation is for IBM MQ V710, but you can see the added detail, such as:

- `BSDS_Reqs`
- `LLCheckpoints`

Later in this lab you use `Detail 20` to rerun MP1B, so you can see the output with the highest level of detail for IBM MQ V8.

__ 22. Still in the logging section of the MP1B documentation, scroll until you find a section that is called *`Additional messages produced in MESSAGE output file`*. These messages contain analysis that MP1B produced from the SMF records.

__ 23. Review the MP1B-generated logger messages in the MP1B documentation. Note the distinction on the `Detail` level that is required in order for some of the messages to be generated.

__ 24. Close the LOG file and open the MESSAGE file. This file contains MP1B-generated messages. It helps answer the question *"Is the queue manager setup OK?"* Review the contents:

__ a. For the collection period used for the course material, the MESSAGE file might not have messages for the logger. However, look at the data reported.

__ b. Notice that the information in the MESSAGE file includes analytical information for other areas, such as SMDS and coupling facility resources.

Note

Support pacs MP16 provides extensive guidance on your logs and buffer pools.

__ 25. Proceed to `C:\WM312\documents` and open the *MP16: Capacity Planning and Tuning for IBM MQ for z/OS*, which is file `mp16-1.pdf`.

__ 26. Search the PDF file for string `Log data set` until you reach the *`Log data set definition`* header.

__ 27. Review the various sections of the log data set guidance in the MP16 document.

__ 28. For a current snapshot, you can enter the `/MQ## DIS LOG` command from your SDSF panel.

Section 4: Review the buffer pool statistics

__ 29. In your MP1B documentation, proceed to the top of page where you find header *Buffer pool statistics*, or proceed to page 53. (*As of the date this course was written, a find did not work for the buffer pool heading due to special characters in the PDF file*).

__ 30. Capture the SCSQMACS member name that contains field information for buffer manager unformatted records: `CSQD`**`QPST`**

You use this name in a later step. Keep the documentation open on this page.

__ 31. Close the MESSAGE file, and open the BUFF file.

__ 32. By using the MP1B documentation, review the records in the BUFF file. Leave this file open.

__ 33. Review the MP1B-generated messages that are documented at the end of the buffer display information.

__ 34. Return to the MP16 document. In the PDF document, look for section *`Definition of buffer pool statistics`*, which should be near page 154 (find does not work for this string as it has special characters).

__ 35. There are three tables in the section selected. If you remove the **`QPST`** prefix from each field, you find that the fields match the labels for the information that is presented in the BUFF file.

__ 36. Review section *`Definition of buffer pool statistics`*, and the next section, *`Interpretation of MQ statistics`*. Review the information in the BUFF file as needed when you read these sections.

__ 37. In the BUFF file, find the first occurrence of the word above by typing `f above` in the ISPF command input prompt. Press the Enter (Ctrl) key. You should find information similar to the example display:

```
03 Above the bar   PAGECLAS 4KB
```

You notice that the PAGECLAS is 4 KB. What is the guidance for PAGECLAS?

__ 38. From the top of the MP16 document, use `find` in the PDF document for string `PAGECLAS`. You should now be around page 21. What relevant information about the PAGECLAS setting is found in MP16?

____

____

__ 39. For a quick snapshot, you can enter the `/MQ## DIS USAGE` command from your SDSF panel.

Section 5: Determine what takes place with the queues and applications

__ 40. Close the BUFF file in your MP1B output; keep the MP1B job open to the list of JES output files.

__ 41. In the MP1B documentation, locate the start of section titled *`Accounting data describing the task`*. You can do a find in the PDF file for this section; or scroll to the top of page 64.

__ a. You are at the start of the documentation for the TASK file. The layout of information reported in this file can vary significantly. Variations can be caused by type of task, and the level of `Detail` that MP1B was run with. When this course was created, the documentation for this task was at VRM 7.0. This information in the TASK file is captured in the accounting trace.

__ b. Scroll through this information, and notice the additional sections that pertain to the TASK definition, which spans several sections up to the `QALL` heading.

__ 42. Open the TASK file.

__ a. Notice that the information in this file is organized according to the queue that is used.

__ b. As documented in the different MP1B sections, information that is displayed in the different entries of this file might by vary by the type of task. For example, mover (channel initiator), or a put or get to a local queue.

__ c. Some of the information that is documented in MP1B might surface at a higher `Detail` level. You use a higher `Detail` level later in this exercise.

__ 43. Close the TASK file and open the TASKCSV file. This file contains a summary of the transactions, jobs, and channels that are obtained from the accounting class(03) trace.

__ 44. Close the TASKCSV and open the TASKSUM file.

__ a. The TASKSUM file might be a good place to start when finding a problem, depending on the information captured.

__ b. Note how some of the MP1B-generated messages embed the four-character identifier or “eye-catcher” for the IBM MQ component.

__ c. Other messages include the string “TASK” embedded in the message number.

__ 45. Find the section that is named *`Investigate potential problems`* in the MP1B documentation; around the middle of page 9.

__ 46. Review the suggestions on how to follow up with TASKSUM entries. Not all messages reported in this file are suitable for follow-up. With practice, you learn which type of message in this file you should use for more investigation.

__ 47. Close the TASKSUM file, and open the QALL file.

__ a. Look at how the information is organized, by queue.

__ b. Check field `Total` queue `CPU` used. You might elect to create your application profile by extracting all records for this field from this file.

Section 6: Review channel initiator information

__ 48. Close the QALL file, and open the CHINIT file.

__ 49. Review the contents. The information in this file is helpful to build an application profile. For example, how close is the application to a channel limit, and how much storage does the channel initiator use?

__ 50. Close the CHINIT file, and open the CHINCSV file. Here, you find the data in the CHINIT file, in CSV format ready for import into a spreadsheet to build a profile.

__ 51. In your MP1B documentation, find heading `Channel Accounting`; it is at the top of page 25. The information in this file is similar to the output of a DIS CHSTATUS command.

__ 52. Close the CHINCSV file, and open the DCHS file.

__ a. Notice that you can see how many persistent or non-persistent messages used the channel.

__ b. Browse through this file and use the MP1B documentation to review the fields that are explained. The data in MP1B might be for an older version, at a higher level of detail.

Note

Disregard any `CHST record length mismatch` messages in the DCHS file.

Keep the first MP1B output open; you compare selected output files with the same file obtained at a higher MP1B detail level in the next section.

Section 7: Rerun MP1B at the highest detail level

__ 53. Leave the panel with the original MP1B results open in the first screen.

__ 54. Switch to your second TSO screen by pressing PF9.

__ 55. Return to your `TSM00##.WM31.SCSQPROC` and select PDS member `STATS003` for edit.

__ 56. Overlay the record in the //SYSIN file, which now contains `* comment *`, with the highest level of detail, `Detail 20`, as shown in the example.

Example

```
//S1 EXEC PGM=MQSMF,REGION=0M
 //* change userid.MP1B.LOAD to
 //* SMFIN to the dataset with
 //STEPLIB DD DISP=SHR,DSN=D62W
 //SMFIN  DD DISP=SHR,DSN=TSM00
 //SYSIN DD *
 Detail 20
 //IGNORED DD *
```

__ 57. Submit the JCL.

__ 58. Select your last MP1B job output from the SDSF job status panel ***by using a question mark (?)*** so that all output files are listed. Press the Enter (Ctrl) key.

__ 59. Scroll down to select the SYSPRINT file. Look at the value of the `Detail` parameter. The second line of the SYSPRINT output should display `buffer:Detail 20`

Note

In the rest of this section, the original MP1B results are referred to as Original. The second run with Detail 20 is referred to as Detailed.

__ 60. In the Original MP1B job, open file QALL.

__ 61. Open the QALL file in the Detailed MP1B job, and contrast the results with the original output. The Detailed output shows extra fields such as the queue indexing information, when the queue was first opened and last closed, and the number of times the queue was opened or closed. Check what other information is in the detailed view.

__ 62. Close both Original and Detailed QALL output files.

__ 63. Open both DCHS Original and Detailed output files. Note how the Detailed contains some extra information such as compression, DNS resolution, and exit times.

__ 64. Close both DCHS output files.

__ 65. Open both Original and Detailed LOG output files.

  __ a. Take time to review the additional information that is provided in the Detailed file, including the `LLCheckpoints` field.

  __ b. Look at the added detail on the numbers of `CI` writes.

__ 66. Close both LOG output files.

__ 67. Open the Original and Detailed TASK output files.

- __ a. In the Detailed file, find string `CF activity` by typing `F 'CF activity'` in the command input prompt and pressing the Enter (Ctrl) key. Ensure that you use quotation marks.
- __ b. Are you able to find `'CF activity'` in the original output?
- __ c. Do you find other pieces of information in the Detailed output that are not shown in the Original output? ____.

__ 68. Close both TASK output files.

__ 69. Open both TASKSUM output files. Do you see additional information in the Detailed file that did not appear in the Original run? ____________________________________.

__ 70. Leave the Original results open in one of the screens. Use the other screen to submit the job in the next section.

__ 71. Open the BUFF file and proceed to the next section.

Section 8: Use the CSQ4SMFD utility to format the accounting and statistics records

__ 72. Return to your `TSM00##.WM31.SCSQPROC` PDS and open member CSQ4SMFJ for edit.

__ 73. Do not edit any of this JCL. Submit the job.

__ 74. Select your CSQ4SMFJ results from the SDSF job status panel ***by using a question mark (?)*** so that all output files are listed. Press the Enter (Ctrl) key.

__ 75. Scroll through the list of output files. Notice how the file names correspond to the identifier abbreviations used for the different IBM MQ process managers.

__ 76. Select the file that contains the buffer manager information. The identifier is QPST.

__ 77. Scroll through the CSQ4SMFD formatted output. Try to find some of the buffer key fields such as SOS and RIO.

__ 78. Contrast with the MP1B formatted BUFF file. You should find the MP1B formatting easier to work with.

You reached the official end of the exercise. If you have extra time, you can select one of the output files that you did not check in the course or lab, look up its documentation in MP1B or MP16, and look through the file.

When you return to your administrative activities, incorporate accounting and statistics work as a regular part of the administration tasks.

End of exercise

Exercise review and wrap-up

In this exercise, you:

- Formatted accounting and statistics traces and gained familiarity with the job information produced by the MP1 utility
- Reviewed the information in the different output files that MP1B generated
- Identified key fields of information for selected MP1B JES output files
- Used the MP1B and MP16 documentation as a guide to continue you work with accounting and statistics
- Used the MP1B SYSIN data set options to format a trace with extra details
- Formatted an SMF accounting and statistics trace with the CSQ4SMFD utility and contrasted the results with MP1B-formatted output

IBM®