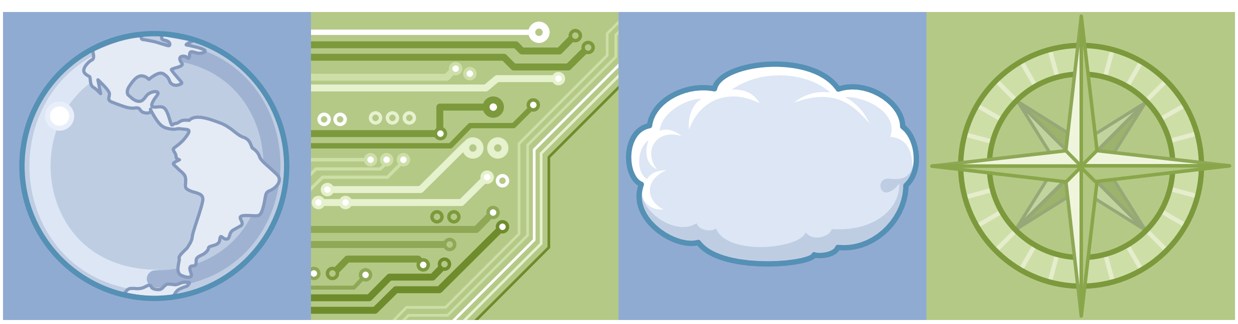
IBM Training

Student Notebook

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

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.

Course description

IBM MQ V8 Advanced System Administration for z/OS

Duration: 4 days

Purpose

This course focuses on advanced IBM MQ for z/OS skills. After a baseline of IBM MQ topics, the course continues with channel security, the need to use TLS ciphers in response to current security threats, and channel authentication rules. The course also covers queue-sharing groups, queue-sharing group updates, and distributed and intra-group queuing with queue-sharing groups. Other topics include the 8-byte relative byte address and buffers above the 2-GB bar features of IBM MQ V8, the use of IBM MQ accounting and statistics, how to capture and review information to resolve or mitigate performance and capacity problems, and how to create application profiles.Hands-on lab exercises throughout the course reinforce the lectures and provide familiarity with advanced tasks and updated capabilities.

Audience

This course is designed for z/OS IBM MQ administrators and other technical professionals who are responsible for IBM MQ security, architecture, availability, and performance-related tasks.

Prerequisites

Before taking this course, you should:

- Successfully complete course WM302, *IBM MQ V8 System Administration for z/OS*, or have equivalent experience
- Be proficient working with z/OS, or successfully complete course ES10G, *Fundamental System Skills in z/OS*

Objectives

After completing this course, you should be able to:

- Explain how SSL/TLS contributes to authentication and confidentiality
- Describe how to configure SSL/TLS in IBM MQ z/OS queue managers and IBM MQ clients
- Implement SSL/TLS with multiple certificate authority (CA) certificates
- Implement various types of channel authentication rules
- Describe how to configure and manage queue-sharing groups

- Explain how to incorporate shared message data sets and storage class memory to maximize coupling facility storage
- Describe how to add a queue-sharing group to an existing IBM MQ cluster
- Implement 8-byte relative byte address and buffers above the 2-GB line
- Describe how to use statistics and accounting capabilities for problem determination, tuning, and capacity planning activities

Agenda

Day 1

Course introduction
Unit 1. IBM MQ baseline
Exercise 1. Configuring the IBM MQ lab environment
Unit 2. Channel security with SSL/TLS
Exercise 2. Configuring SSL/TLS for queue managers and channels

Day 2

Unit 3. Channel security with channel authentication rules
Exercise 3. Configuring channel authentication rules
Unit 4. Queue-sharing groups
Exercise 4. Configuring queue-sharing groups and shared queues

Day 3

Unit 5. Intra-group, distributed queuing, and clustering with queue-sharing groups
Exercise 5. Intra-group and distributed queuing with queue-sharing groups
Unit 6. Eight-byte relative byte address (RBA) and buffers above the 2-GB line
Exercise 6. Implementing 8-byte relative byte address and buffers above the 2-GB line

Day 4

Unit 7. Introduction to IBM MQ for z/OS statistics and accounting
Exercise 7. Getting started with IBM MQ statistics and accounting records
Unit 8. Course summary

Unit 1. IBM MQ baseline

What this unit is about

This unit reviews the IBM MQ baseline requirements for this course and establishes the advanced topics for the course.

What you should be able to do

After completing this unit, you should be able to:

- Describe IBM MQ basic components
- Explain how to resolve common channel problems
- Distinguish between IBM MQ servers and IBM MQ clients
- Describe clusters and cluster channels
- Contrast the IBM MQ messaging styles
- Describe IBM MQ architectural options
- Summarize how IBM MQ is implemented in z/OS
- Identify the administrative options for IBM MQ on z/OS
- Describe security options for IBM MQ channels and clients

How you will check your progress

- Checkpoint questions
- Lab exercises

WebSphere Education

Unit objectives

- Describe IBM MQ basic components
- Explain how to resolve common channel problems
- Distinguish between IBM MQ servers and IBM MQ clients
- Describe clusters and cluster channels
- Contrast the IBM MQ messaging styles
- Describe IBM MQ architectural options
- Summarize how IBM MQ is implemented in z/OS
- Identify the administrative options for IBM MQ on z/OS
- Describe security options for IBM MQ channels and clients

Figure 1-1. Unit objectives WM3121.1

Notes:

WebSphere Education

IBM MQ components and baseline

- First part of this unit focuses on the basic IBM MQ components
  - Queue manager
  - Messages
  - Queues
  - Message queue interface (MQI)
  - Channels
- IBM MQ can be installed with IBM MQ server software, with IBM MQ client software (except z/OS), or with both
  - IBM MQ server and IBM MQ client are two distinct installed products
  - Unless a topic is identified as applying to IBM MQ client, the information pertains to installations of IBM MQ server software

Figure 1-2. IBM MQ components and baseline WM3121.1

Notes:

This unit discusses the basic IBM MQ concepts:

- The queue manager, or messaging server, owns IBM MQ resources and controls processes such as defining objects, trigger starting a channel or process, creating event messages, expiring messages, and message distribution by using clustering.
- Messages are the unit of data that the queue manager exchanges.
- Queues are where messages are kept.
- The Message Queue Interface, or MQI, is the application interface that is used to put and get messages in IBM MQ.
- Channels move messages from one queue manager to another queue manager, sometimes jumping across other queue managers.

Note

Do not confuse the MQI with IBM MQ client channels, which are also referred to as MQI channels.

WebSphere Education IBM

Queue manager

- System program that owns the resources it services
- Provides services that are needed for messaging
- A server can host more than one queue manager
- Queue managers that share a server need different TCP/IP port numbers

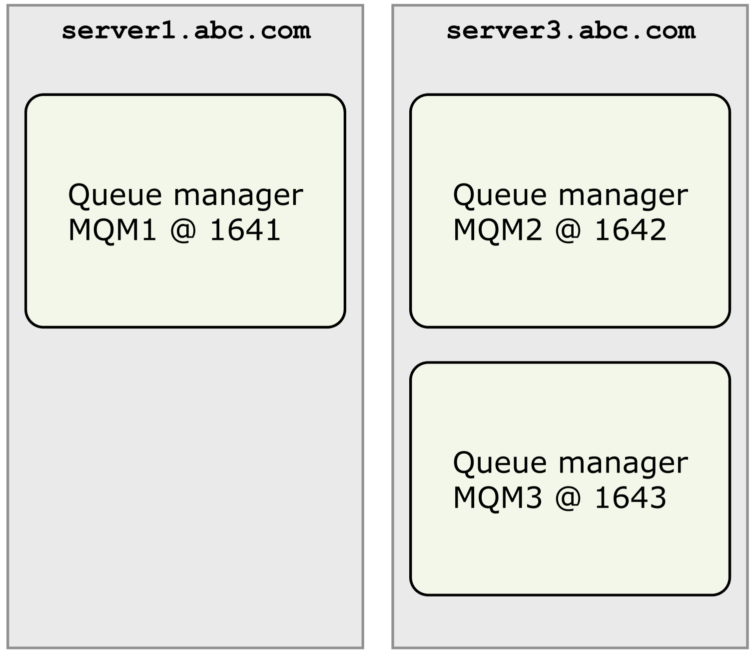


Figure 1-3. Queue manager WM3121.1

Notes:

There can be more than one queue manager in the same z/OS or distributed server. Each queue manager must listen on a different port.

You create queue managers in the same server, so when the channel initiator is configured, each of your queue managers has a specific assigned port number.

WebSphere Education

Messages

- Messages contain the business data or payload
- Messages are a distinct string of bytes exchanged between two programs
- All messages contain an IBM MQ message descriptor (MQMD) with control information
- Optionally, applications can add Message Properties to a message

Message descriptor (MQMD)	Message properties (optional)	Data: Payload

Figure 1-4. Messages WM3121.1

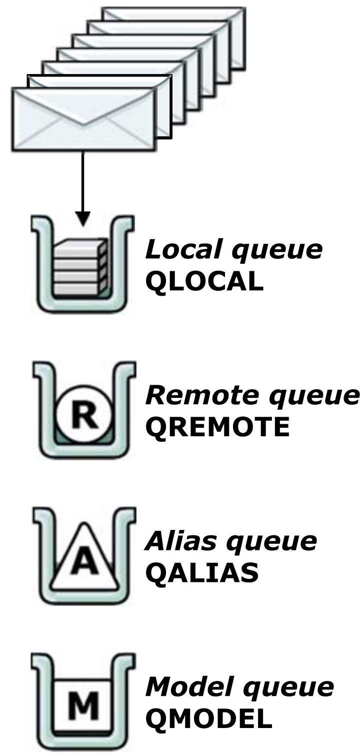
Notes:

Messages contain the message descriptor, or MQMD structure, and the message data or payload. Messages might contain other headers.

WebSphere Education

Queues

- A queue is a defined destination for messages
- Four types of queues can be created: QLOCAL, QREMOTE, QALIAS, QMODEL
- Some local queues are designated for special purposes in a queue manager
- Remote queues or QREMOTE are pointers to local queues in a remote queue manager
- An alias queue or QALIAS is a pointer to a local queue or a locally-owned remote queue
- A model queue or QMODEL is a template to create a dynamic local queue



Only local queues that are defined as QLOCAL queue type hold messages.

Figure 1-5. Queues WM3121.1

Notes:

Applications put and send messages by using queues. When you start working with IBM MQ, you might spend some time with developers or testers to determine “where is my message?”

A basic concept is that only local queues that are defined as QLOCAL can hold messages. Keeping this concept in mind is helpful when looking for a message.

Other queue types are pointers to other queues:

- Remote queues point to a transmit queue, and a remote queue manager and target queue. QREMOTE is the queue type only. These queues are “local QREMOTEs”.
- Alias queues, or QALIAS, point to another local queue or a topic.

Model queues, or QMODEL, can be thought of as a template to create queues, but also do not hold messages.

You also hear about shared queues, which are exclusive to z/OS. These queues belong to a queue-sharing group and might also be QLOCALs.

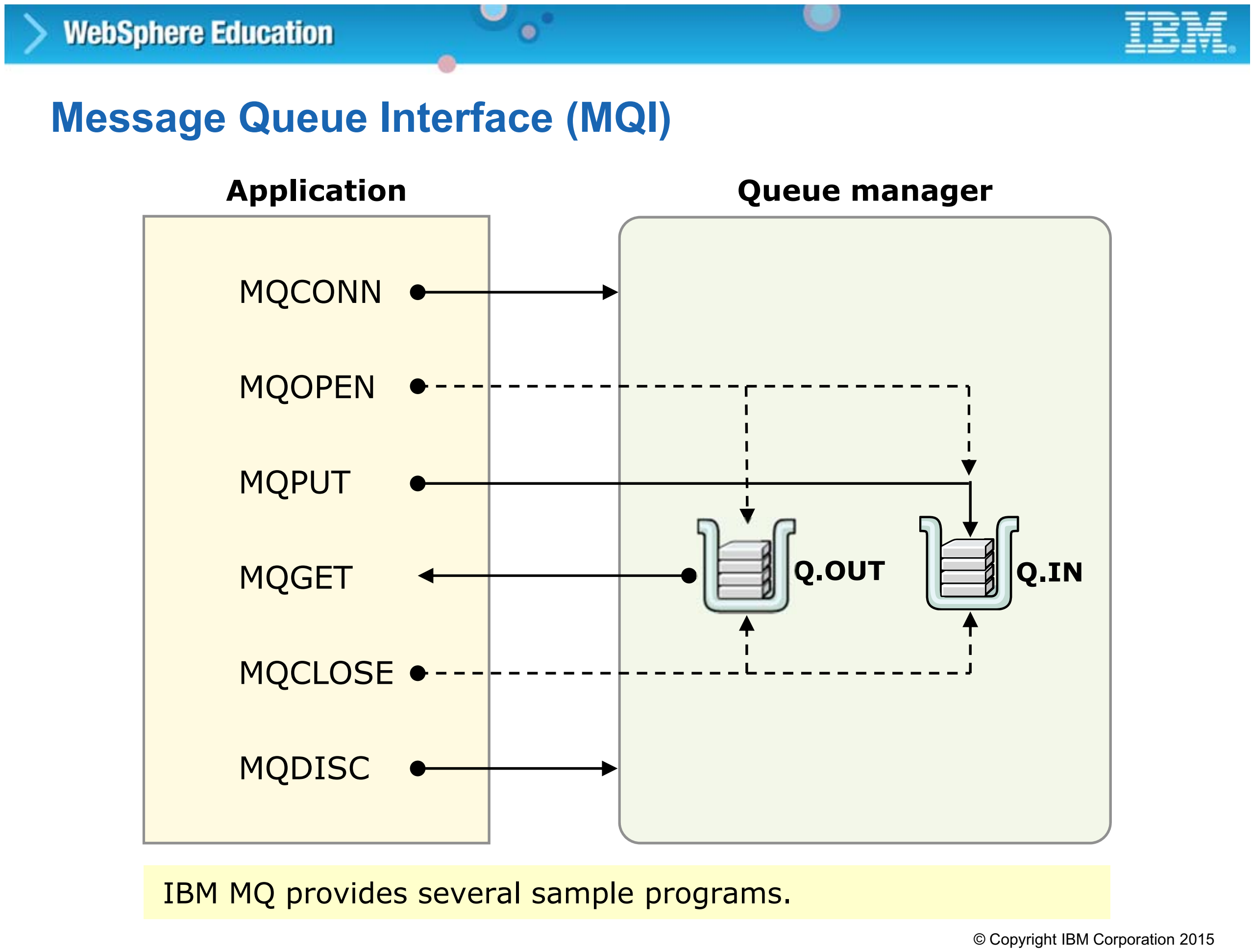


Figure 1-6. Message Queue Interface (MQI) WM3121.1

Notes:

Queue managers, messages, queues, and channels are explained. But how are messages placed into queues?

Applications connect to a queue manager and open a queue. If the MQCONNECT and MQOPEN calls are successful, then messages are put to the queue. If many messages are expected to be placed in the queue, there can be a loop of puts to accomplish this purpose. It would be more efficient than connecting and opening the queue for every put.

It is important to always close the queues and disconnect from the queue manager when finished placing messages in the queue.

WebSphere MQ contains numerous code samples and compiled sample applications that can be used to test putting and getting messages from a queue.

WebSphere Education IBM

Channel categories and types

- Two main categories of channels
  - Message channels: Distributed or clustered
  - Client (or MQI) channels
- A message channel is a combination of a sending MCA, the network connection, and the receiving MCA
- Message channels are defined in a caller and responder pair
- Distributed message channel types:
  - Sender
  - Server
  - Receiver
  - Requester

Valid distributed message channel pair combinations

Caller	Flow direction	Responder
Sender	→	Receiver
Server	→	Receiver
Server	→	Requester
Requester	←	Server
Requester	←	Sender ***Callback***

Figure 1-7. Channel categories and types WM3121.1

Notes:

A sending MCA, the communication link, and the receiving MCA form a one-way message path that is called a channel. There are several types of distributed message channels, and the main difference is the way that they get started. The sender-receiver pair is the most prevalent.

MQI channels are not IBM MQ server channels. They are IBM MQ client channels. They are not one-way but used with two-way communication.

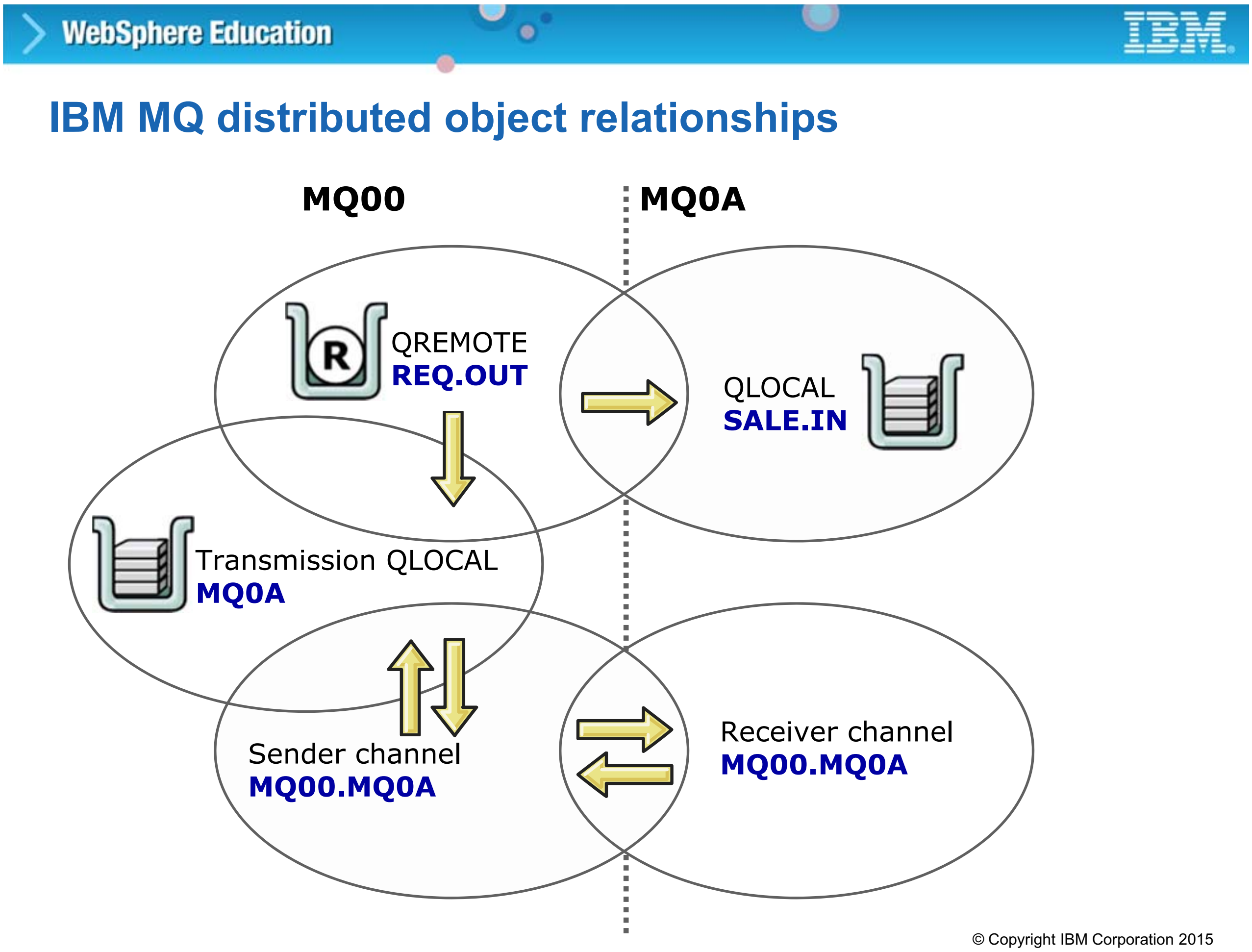


Figure 1-8. IBM MQ distributed object relationships WM3121.1

Notes:

The same transmit queue can be used for several remote queues, and for one sender channel.

Looking at object relationships, when a message is put in the MQM1 local remote queue REQ.OUT, this message is placed in the MQM3 transmit queue.

If channel MQM1.MQM3 is running, the MCA consumes the message and sends the message to its remote target local queue at MQM3, SALE.IN.

Designating MQM1 as the local queue manager, you can see how the QREMOTE is related to the transmission queue on the local queue manager, and also to the target queue in the remote queue manager. The sender channel is related to the transmission queue in the local queue manager and the receiver channel in the remote queue manager.

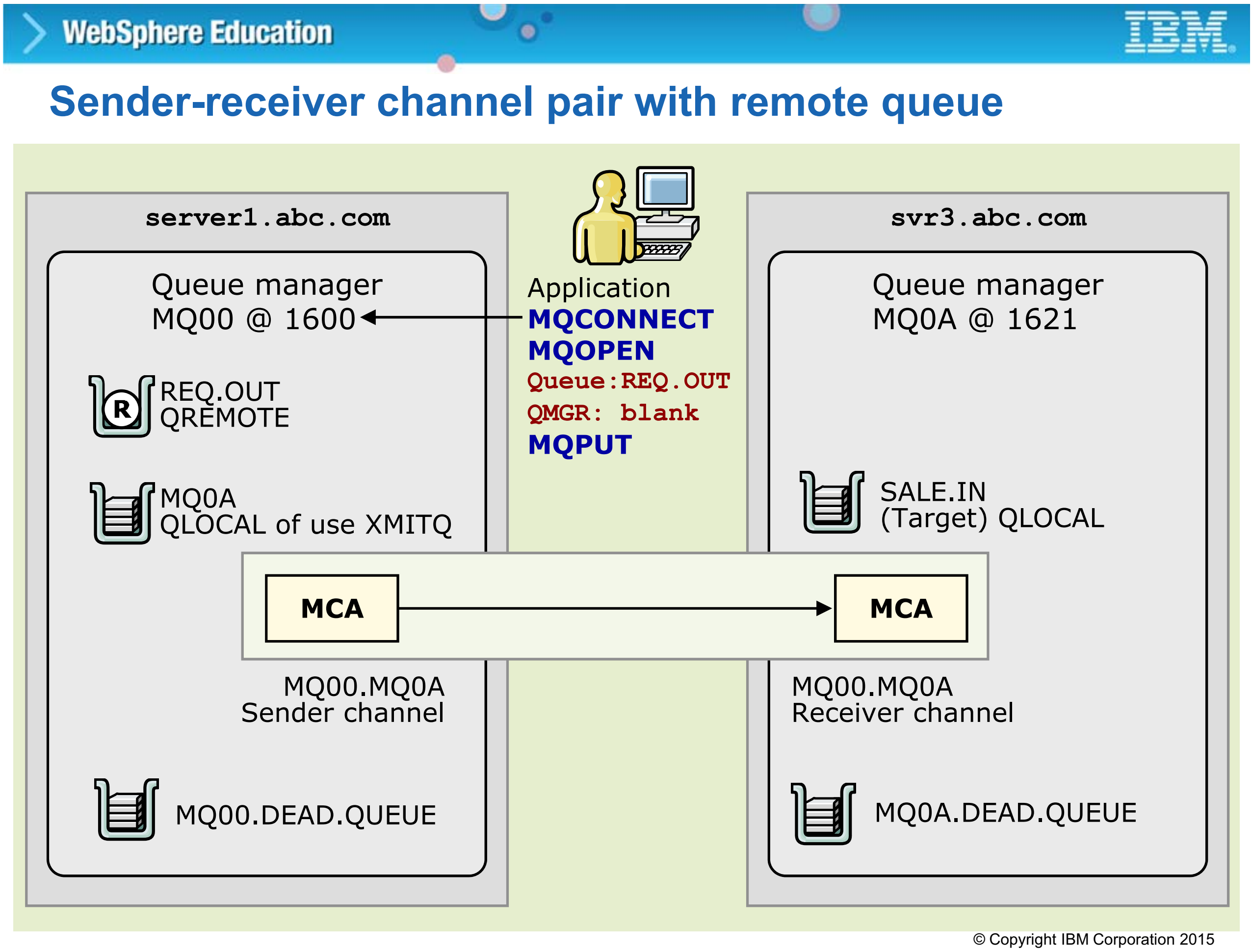


Figure 1-9. Sender-receiver channel pair with remote queue WM3121.1

Notes:

Channels are used to transmit messages between queue managers. IBM WebSphere MQ interfaces to networks are called message channel agents (MCA).

- The SENDing message channel agent MQGETs messages from the transmit queue under syncpoint.
- It transmits the message to the receiving MCA, which MQPUTs it to the destination queue, under syncpoint, first MQOPENing it if necessary.
- At the end of a batch of messages, the two MCAs agree to the batch status and MQCMIT their queues.
- All of this work might be logged for recovery. It is also possible not to use syncpoint for nonpersistent messages only. The NPMSPEED parameter on the SENDER (or SERVER) channel definition can be used to allow nonpersistent messages to flow immediately.

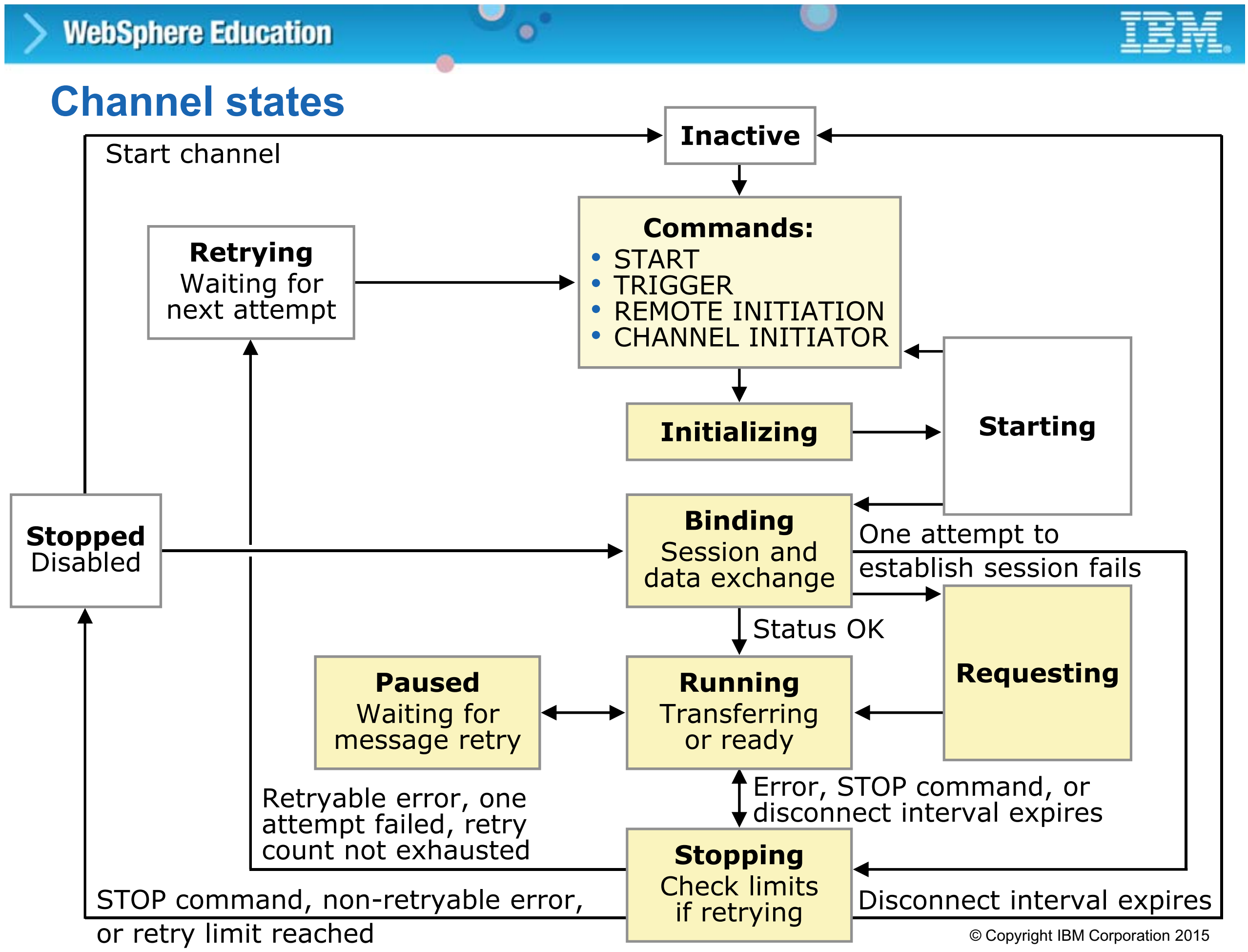


Figure 1-10. Channel states WM3121.1

Notes:

Channels go across several states before they start. The most common error you find when first starting a channel is “retrying.” Retrying is often the result of a problem with the definition, channel pairs unable to find each other, possibly a wrong port, or a misconfigured host name or IP address.

When the problem is rectified, the channel normally starts on its own. If the channel does not automatically start after the correction, then stop the sender (assuming that you have a sender-receiver pair), and then restart it.

If a channel is manually stopped or is in the STOPPED state, it must be manually restarted, even if it is triggered.

A brief description of the states in the figure follows. The figure is a simplified representation of what happens. The flows between the states, in particular, should not be interpreted too literally.

START is not a state. It indicates a start point for when a command was entered to start a channel, or when the arrival of a message on the transmission queue triggers the start of a channel. It can also indicate a start point when a channel initiator decides it is time for the next attempt to start a channel, or when there is an incoming request to start a channel.

- INITIALIZING: Before a channel initiator starts an MCA, it creates an entry for the channel in the channel status table, if an entry does not yet exist, and sets the state of the channel to INITIALIZING. This entry acts as a “placeholder” in case the MCA fails before it has a chance to put an entry in the channel status table itself. If the MCA fails without such an entry, the channel Initiator knows nothing about the channel and so there would be no further attempts to start it.
- STARTING: A channel waits in this state if no active slot is available. However, if there is no active slot for the requester end of a channel, or the responder end of a channel, the channel fails to start and does not enter the STARTING state. To start a channel that is in the STARTING state when an active slot does become available, a channel initiator must be running.
- BINDING: A channel is establishing a communications connection and is doing the initial data negotiation. However, a requester that is also a caller remains in this state only while it is establishing a communications connection.
- REQUESTING: A requester that is also a caller is doing the initial data negotiation.
- RUNNING: A channel is transferring messages, or waiting for messages to arrive on the transmission queue.
- PAUSED: A channel is waiting for the message-retry interval to elapse before attempting to open a destination queue or place a message on it. This state does not occur on IBM WebSphere MQ for Windows as message-retry is not supported.
- STOPPING: A channel encounters a failure, or a command is entered to stop the channel, or the disconnect interval elapses.

  If the channel encounters a failure, it checks whether its retry limits were reached while it is in this state. If the limits were not reached, the channel enters the RETRYING state. If the limits were reached, or the retry counts were set to zero in the channel definition, the channel enters the STOPPED state. However, on a version 5 queue manager, a channel becomes inactive if no channel initiator is running, whether the retry limits were reached or not.

  If the disconnect interval elapses, the channel becomes inactive. If a command is entered to stop the channel, the channel enters the STOPPED state.
- RETRYING: A channel is waiting until it is time for the channel initiator to make the next attempt to start the channel.
- STOPPED: In this state, the channel is disabled and needs manual intervention to start it again.
- INACTIVE: An inactive channel is one that was never started, or one for which the disconnect interval elapsed, or one that is considered to have ended “normally.”

The current state of a channel can be determined by using the DISPLAY CHSTATUS command.

Current and active channels

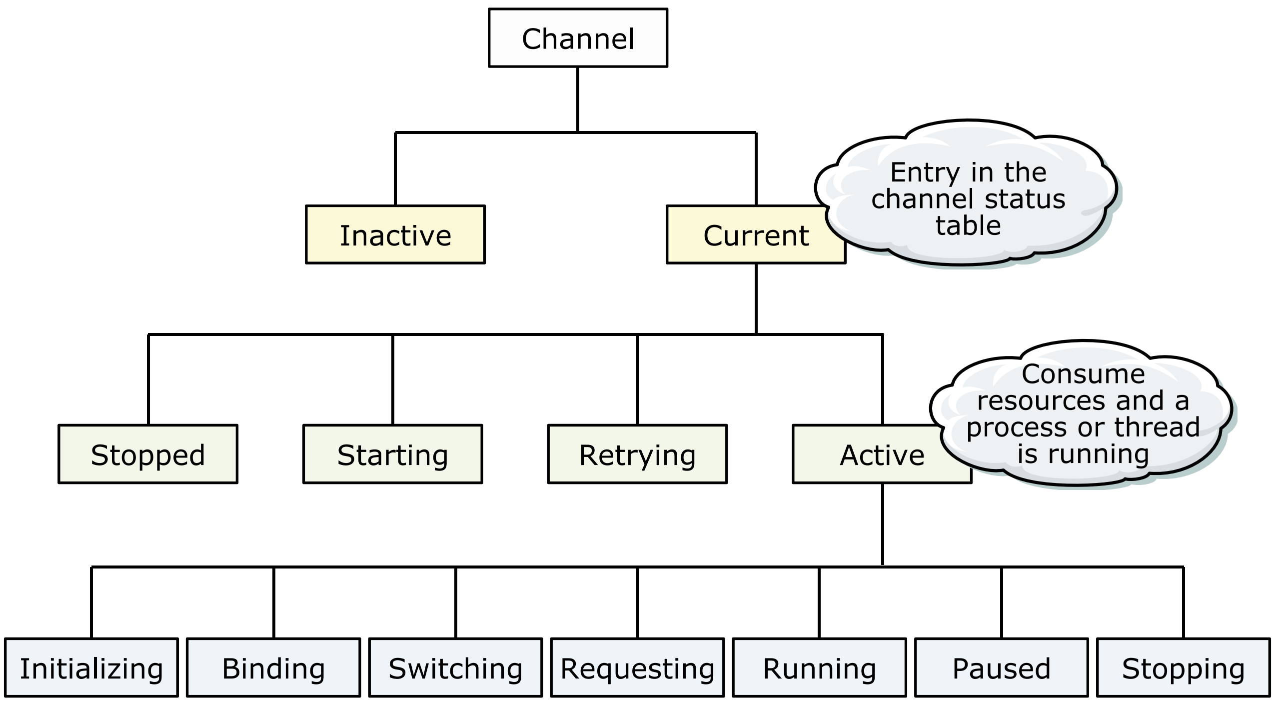


Figure 1-11. Current and active channels WM3121.1

Notes:

WebSphere Education IBM

Current and saved status data

SAVED status

```
/MQ00 DIS CHS(MQ00.MQ0A)+
   SAVED ALL
 CHSTATUS(MQ00.MQ0A)
 CHLDISP(PRIVATE)
 XMITQ(MQ0A)
 CONNAME(MQ0A)
 SAVED
 CHLTYPE(SDR)
 INDOUBT(NO)
 LSTSEQNO(12)
 LSTLUWID(CDB054432EB…
 CURMSGS(0)
 CURSEQNO(12)
 CURLUWID(CDB054432EB…
  END CHSTATUS DETAILS
```

CURRENT status

```
/MQ00 DIS CHS(MQ00.MQ0A)+
   ALL
 CHLDISP(PRIVATE)
 … … …
 CURRENT
 CHLTYPE(SDR)
 STATUS(RUNNING)
 SUBSTATE(MQGET)
 INDOUBT(NO)
 LSTSEQNO(20)
 LSTLUWID(CDB15283A9…
 CURMSGS(0)
 CURSEQNO(20)
 CURLUWID(CDB152E0B1…
 LSTMSGTI(13.08.53)
 LSTMSGDA(2014-09-01)
 MSGS(8)
 BYTSSENT(5144)
 BYTSSENT(5144)
 BYTSRCVD(592)
 BATCHES(2)
```

```
  current continued
 CHSTATI(13.07.15)
 CHSTADA(2014-09-01)
 BUFSSENT(10)
 BUFSRCVD(4)
 LONGRTS(999999999)
 SHORTRTS(10)
 MONCHL(OFF)
 …
 KAINT(360)
 QMNAME(MQ00)
 RQMNAME(MQ0A)
 RQMNAME(MQ0A)
 SSLCERTI()
 … … …
 RVERSION(08000000)
 STATCHL(OFF)
 LOCLADDR(10.31.87 … … …
 BATCHSZ(50)
 MAXMSGL(4194304)
 … … … … …
```


Figure 1-12. Current and saved status data WM3121.1

Notes:

When you use the `DISPLAY CHSTATUS` command (or the equivalent ISPF panel option) to display the current status of one or more channels, you specify whether you want the current status or the saved status. Current status data is returned if it is available.

The current status data for a channel is data that is derived from the entry for the channel in the channel status table. One consequence of this is that an inactive channel does not have a current status.

The saved status data for a channel is derived from the status message for the channel on the synchronization queue, or from the in-doubt status message if the channel is in doubt. Thus, an inactive channel might contain saved status data.

The `LOCALADDR` attribute is truncated to fit in the page.

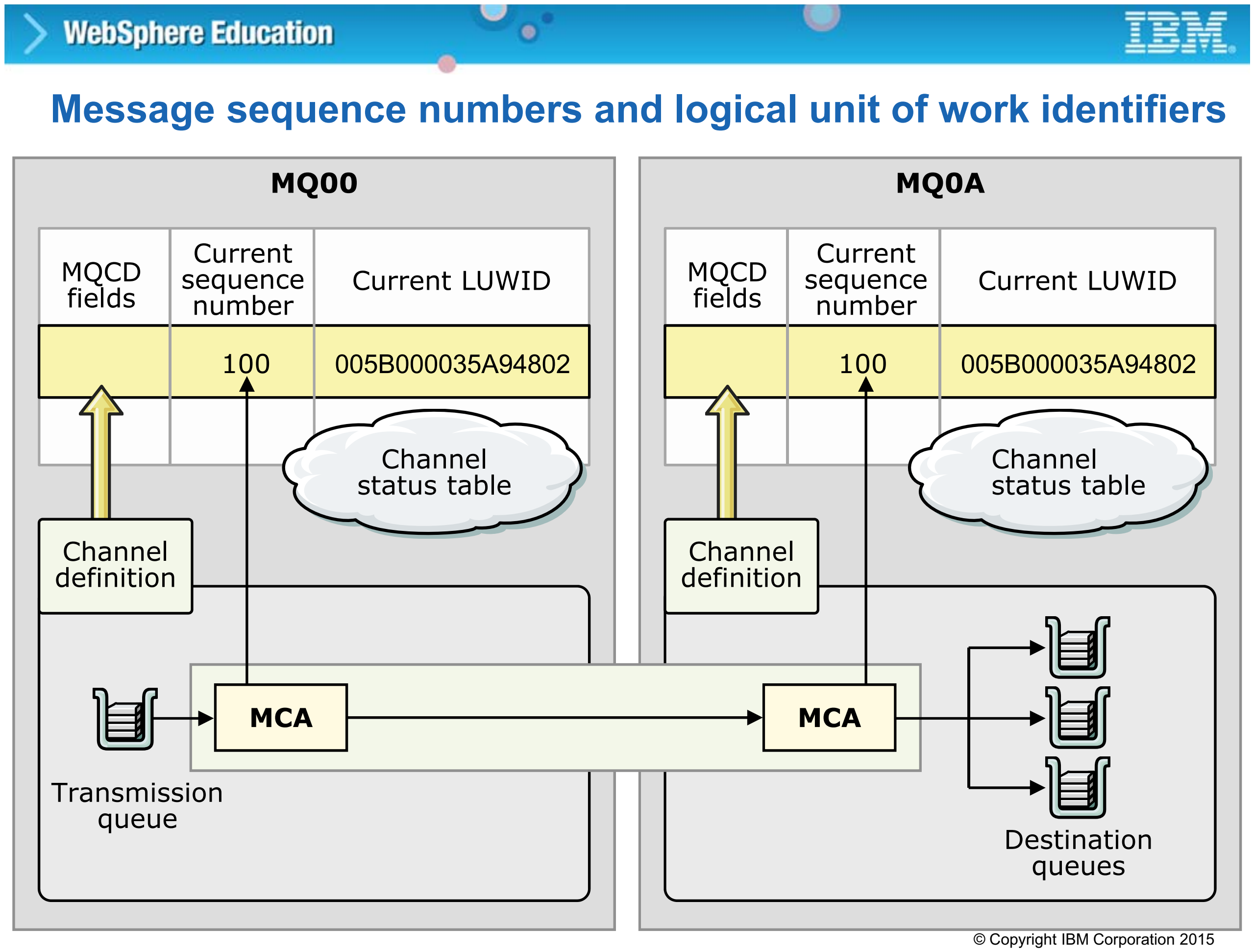


Figure 1-13. Message sequence numbers and logical unit of work identifiers WM3121.1

Notes:

After an MCA is started and connects to the queue manager, it reads the channel definition and uses it to initialize the fields in the channel data structure, MQCD. These fields constitute the initial entry for the channel in the channel status table.

Each queue manager has its own channel status table. On IBM WebSphere MQ for z/OS, the channel status table is stored in the channel initiator address space. Therefore, there is a channel status table at each end of a channel, and there is an entry for the channel in each table. The table is stored in main memory. If there is a failure, the queue manager ends, the information that is held in the table is lost.

The channel status table is used by the various components of distributed queuing to manage channels and for communication among the components. These components include MCAs, channel initiators, IBM WebSphere MQ listeners, and the channel management commands.

Each message that is sent across a channel has an associated message sequence number (MSN) which the sending MCA assigns. The current sequence number for a channel is stored in its entry in the channel status table at each end of the channel. At the sending end of the channel, the current sequence number is the sequence number of the last message sent. The sending MCA increments it by one just before sending a message. At the receiving end of the channel, the current sequence number is the sequence number of the last message received. Thus, a receiving MCA knows which sequence number to expect on the next message it receives. If the sequence number is incorrect, it is assumed that some failure occurred and the channel terminated.

The usual cause of a sequence number mismatch is when an administrator deletes a channel at one end and then redefines it. Also, if a queue manager is deleted, then re-created with all of its objects, but the sequence number no longer matches the partner channel. The **RESET CHANNEL** command is used to reset the sequence number at each or either end of a channel. If the command is entered at the sender or server end of a channel, the sequence number at the receiver or requester end is automatically reset to the same value the next time the channel starts.

One consequence of continually incrementing a sequence number by one is that, at some time, the sequence number must wrap. The highest value that the sequence number reaches before it restarts at 1 is called the *sequence number wrap value*. The **SEQWRAP** parameter on the **DEFINE CHANNEL** command specifies the s*equence number wrap value*. A channel does not start if the sequence number wrap value at one end of the channel is different from the wrap value at the other end.

Sequence numbers are used by administrators for monitoring the health of channels, and for calculating how many messages are sent across a channel within a certain period. In a later unit, you see how to use statistics and accounting to obtain a use history on the channel.

For efficiency, messages are sent across a channel in batches. At the end of a batch, the receiving MCA confirms to the sending MCA that it committed all the messages in the batch. Now the sending MCA commits the removal of the messages from the transmission queue. This procedure is part of the protocol that ensures that a message is delivered once.

Each batch of messages is assigned a unique identifier by the sending MCA. This identifier is called a *logical unit of work identifier (LUWID)*. At the start of a new batch, the sending MCA communicates the LUWID for the batch to the receiving MCA. The LUWID for the current batch is also stored in the channel status table at each end of the channel.

An MCA also uses the sequence number of the last message in the batch. You will see how both identifiers are used on the next visual. From the point of view of the administrator, LUWIDs are possibly more convenient as batch identifiers because they are a little more constant than sequence numbers.

WebSphere Education

Message sequence number error

```
CSQX500I MQ0A CSQXRESP Channel MQ00.MQ0A started 117
connection 10.31.187.59
… … …
CSQX526E MQ00 CSQXRCTL Message sequence error for channel MQ00.MQ0A,
sent=13 expected=1
… … …
CSQX526E MQ0A CSQXRESP Message sequence error for channel MQ00.MQ0A,
sent=13 expected=1
… … …
CSQX599E MQ0A CSQXRESP Channel MQ00.MQ0A ended abnormally 120
connection 10.31.187.59
… … …
CSQX558E MQ00 CSQXRCTL Remote channel MQ00.MQ0A not available
```


Figure 1-14. Message sequence number error WM3121.1

Notes:

This screen display is an example of messages that are observed in the system log when there is a message sequence error condition. When you see such an error, the **RESET** command must be used to set the correct sequence number.

WebSphere Education

Transactions: Terminology

- A **resource manager** is a system that owns and controls its components such as
  - A database manager owns its tables
  - A queue manager owns its queues
- A **transaction** or **unit of work** is a set of changes that must be completed in their entirety (**committed**) or restored to a previous consistent state (**backed out**)
- A **transaction manager** is a subsystem that coordinates units of work
- An IBM MQ queue manager can:
  - Act as a transaction manager over its own resources
  - Manage updates to DB2 tables
  - Run under a compatible external transaction manager
- An ATM withdrawal is a common scenario of a process that needs to be coordinated with a transaction manager

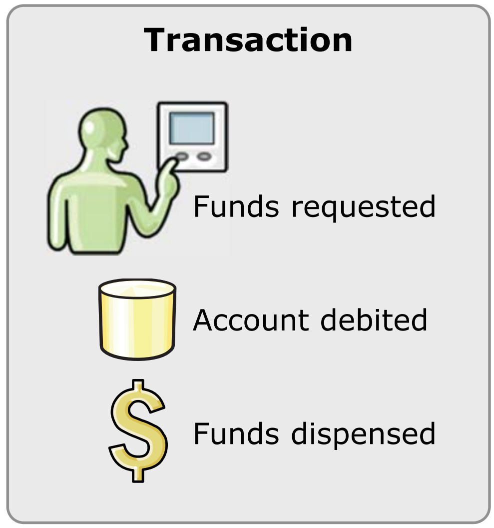


Figure 1-15. Transactions: Terminology WM3121.1

Notes:

This slide baselines transaction-related terminology.

WebSphere Education

Transactions: Local units of work

- Local units of work occur when all the puts and gets are coordinated by the same queue manager as a **single phase commit** process
- The MQBEGIN and MQCMIT or MQBACK verbs are used to start and complete the unit of work
- The channel process is an example of a local unit of work
  - A channel process defaults to a batch of 50 messages
  - The channel commits after it sends all the messages in the transmission queue or reaches the configured batch size
  - Messages are not visible by the receiving application until the MQCMIT is issued

```
MQCONN
MQOPEN

MQBEGIN

MQPUT
MQPUT
MQGET
MQGET

MQCMIT or
MQBACK

MQCLOSE
MQDISC
```


Figure 1-16. Transactions: Local units of work WM3121.1

Notes:

Local units of work involve only the queue manager. Coordination is done by the local queue manager.

Applications start a local unit of work with one of the MQI functions, such as MQGET, MQPUT, specifying the appropriate syncpoint option.

When the work is completed, the function MQCMIT is used to commit the work, or MQBACK is used to roll it back.

WebSphere Education

Transactions: Global units of work

- Global units of work are transactions that need to be coordinated across two or more resource managers by using a two-phase commit process
- Global units of work might involve different queue managers and database tables

Using an IBM MQ queue manager as the transaction manager

- Supported on Linux, UNIX, and Windows environments only
- Unit of work that is started with the MQBEGIN verb
- Unit of work that is completed with the MQCMIT or MQBACK verb, for example:

```
MQCONNECT
MQOPEN
MQBEGIN
MQGET
MQPUT
SQL INSERT
MQCMIT or MQBACK
MQCLOSE
MQDISC
```

Using an external transaction manager

- The external transaction manager must be compatible (X/Open XA compliant)
- IBM MQ queue manager is a participant but does not manage the transaction

- An application requests the external transaction manager (TM) to start the unit of work
- Transaction is controlled by TM API
- No MQBEGIN, MQCMIT, or MQBACK
- Examples of transaction managers are CICS and Microsoft Transaction Server
- Requirements for each TM must be confirmed

Figure 1-17. Transactions: Global units of work WM3121.1

Notes:

This slide presents global units of work, which involve coordinating transactions over two or more resources.

MQBEGIN is available on distributed platforms. z/OS always has a transaction manager.

Unless running under an external transaction manager, MQCMIT and MQBACK are available in all platforms.

Syncpoint options should always be specified on MQI calls.

- z/OS MQI assumes `SYNCPOINT`
- Distributed MQI assumes `NO_SYNCPOINT`

Two-phase transactions on z/OS can use RRS, CICS, and IMS for transaction management. On distributed platforms, XA is a standard two-phase interface, and IBM MQ can act as a transaction manager.

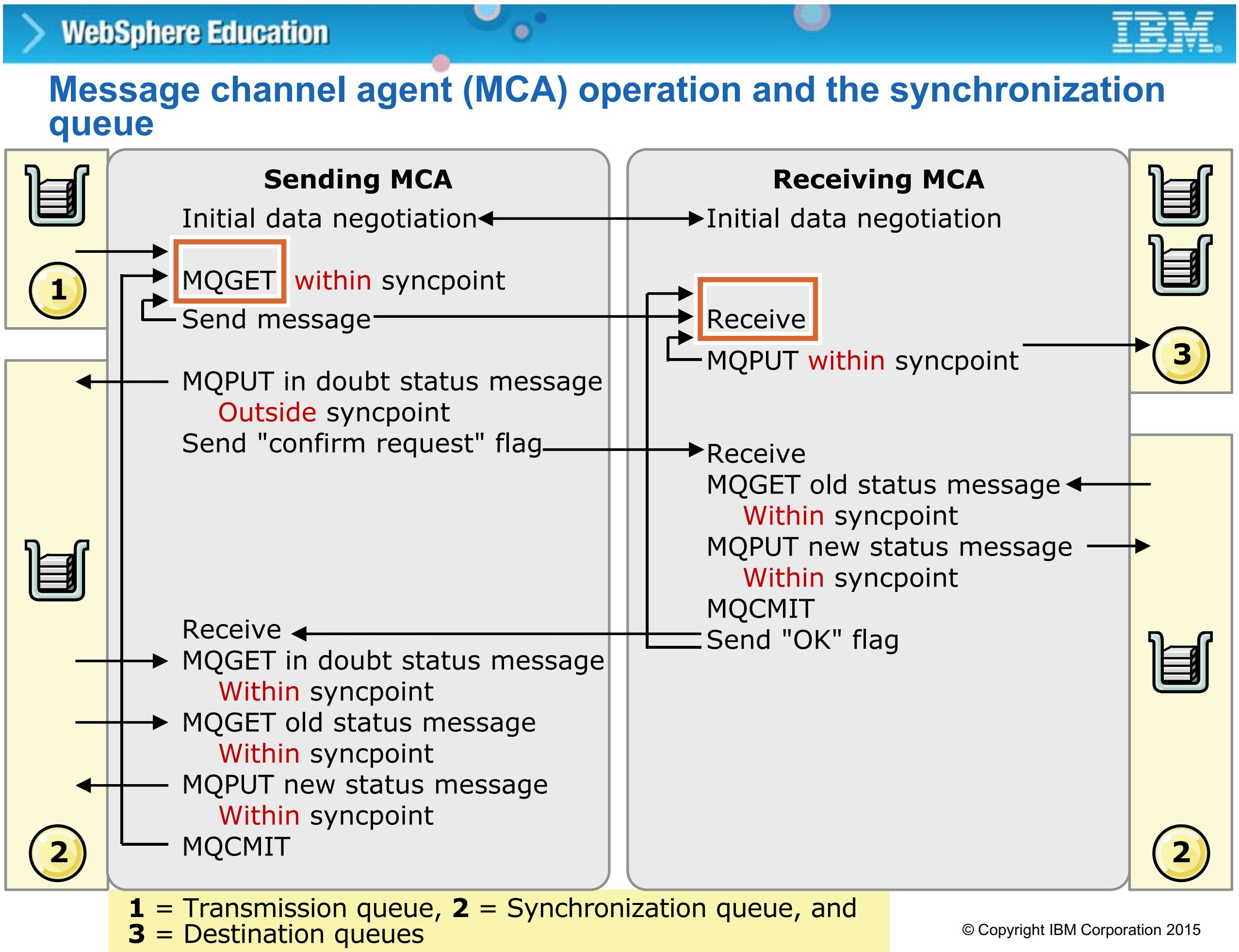


Figure 1-18. Message channel agent (MCA) operation and the synchronization queue WM3121.1

Notes:

When a channel starts and a communications connection is established, the two MCAs enter a phase that is called the *initial data negotiation* before messages start to flow. During the initial data negotiation, the MCAs exchange certain information about their respective ends of the channel.

Certain parameters that control the operation of the channel are negotiated, and the MCAs agree on which one does certain tasks. Here are some examples of what transpires during this phase.

- Checks are made that a channel with the same name is defined at both ends, and that the type of the channel at one end is compatible with the type at the other.
- The two MCAs compare the value of the **SEQWRAP** parameter at each end of the channel. If the two values are not the same, the channel ends.
- The MCAs agree on which one converts the data in the transmission queue header that accompanies each message that is transferred on the channel. The usual rule is “receiver makes good.”
- The MCAs agree on the maximum size of a batch of messages. The agreed size is the lower of the values of the **BATCHSZ** parameter at each end of the channel. The agreed size is reduced

even further if the maximum number of uncommitted messages that are allowed within a single unit of work at either queue manager is lower.

- The MCAs agree on the maximum length of a message that can be transferred on the channel. The agreed size is the lower of the values of the **MAXMSGL** parameter at each end of the channel.
- Each MCA determines whether its partner supports heartbeat flows. If both MCAs support them, then the agreed size of the heartbeat interval is the larger of the values of the **HBINT** parameter at each end of the channel. The use of heartbeat flows is discussed later in this topic.
- The MCAs agree on the class of service for nonpersistent messages on the channel, as determined by the value of the **NPMSPEED** parameter at each end of the channel. The value can be either **FAST** or **NORMAL**. If the values at the two ends of the channel do not match, or if one end of the channel does not support fast nonpersistent messages, **NORMAL** is used. Fast nonpersistent messages are discussed later in the course.
- The sending MCA determines whether the partner queue manager supports distribution lists and sets the *DistLists* attribute of the transmission queue to **MQDL_SUPPORTED** or **MQDL_NOT_SUPPORTED**.

Transferring a batch of messages

After the initial data negotiation is complete and the two MCAs agreed to proceed with the transfer of messages, the sending MCA gets a message from the transmission queue within syncpoint control and sends the message to the receiving MCA. The receiving MCA puts the message on its intended destination queue, also within syncpoint control. The sending MCA then continues to get messages from the transmission queue and send them to the receiving MCA until one of the following conditions is met:

- The agreed maximum number of messages in a batch is reached.
- There are no more messages on the transmission queue, and an interval, specified by the **BATCHINT** parameter, elapsed while waiting for a message.

End of batch commitment processing

After the sending MCA closed a batch, it then needs to determine whether the receiving MCA safely committed all the messages in the batch on their respective destination queues. Until the sending MCA receives such confirmation, it does not commit the removal of the messages from the transmission queue, and the channel is said to be *in doubt*.

The sending MCA puts an in-doubt status message, outside of syncpoint control, on the synchronization queue, **SYSTEM.CHANNEL.SYNCQ**, at its end of the channel. The in-doubt status message contains the following information:

- The `LUWID` of the in-doubt batch
- The sequence number of the last message in the in-doubt batch
- The number of messages in the in-doubt batch
- The message identifier of each message in the in-doubt batch
- The `LUWID` of the last committed batch, that is, the previous batch
- The sequence number of the last message in the last committed batch

The sending MCA then sends a confirm request flag on an IBM WebSphere MQ flow to the receiving MCA. If the batch is closed because the maximum agreed batch size is reached, the sending MCA can include the flag on the flow for the last message in the batch. Otherwise, it requires a separate flow.

On each queue manager, the synchronization queue contains a status message for each channel that transferred at least one batch of messages. The status message for a channel contains the following information:

- The LUWID of the last committed batch, that is, the previous batch
- The sequence number of the last message in the last committed batch

When the receiving MCA receives the confirm request flag, it gets the status message for the channel from the synchronization queue within syncpoint control, which marks it for removal from the queue. It then puts a new status message for the channel on the synchronization queue, also within syncpoint control. This status message contains the LUWID for the current batch and the sequence number of the last message in the current batch. The receiving MCA then calls **`MQCMIT`** to commit all the messages on their respective destination queues, the new status message for the channel on the synchronization queue, and the removal of the old status message. Finally, the receiving MCA sends the OK flag on an IBM WebSphere MQ flow to the sending MCA. The receiving MCA then reverts to waiting on a communications receive.

When the sending MCA receives the OK flag, it gets the in-doubt status message for the channel from the synchronization queue within syncpoint control. In addition, it gets the old status message and puts a new status message, also within syncpoint control. It calls **`MQCMIT`** to commit the new status message on the synchronization queue, the removal of the in-doubt status message and the old status message from the synchronization queue, and the removal of the messages in the batch from the transmission queue.

After the batch of messages is processed, the sending MCA starts the processing for the next batch.

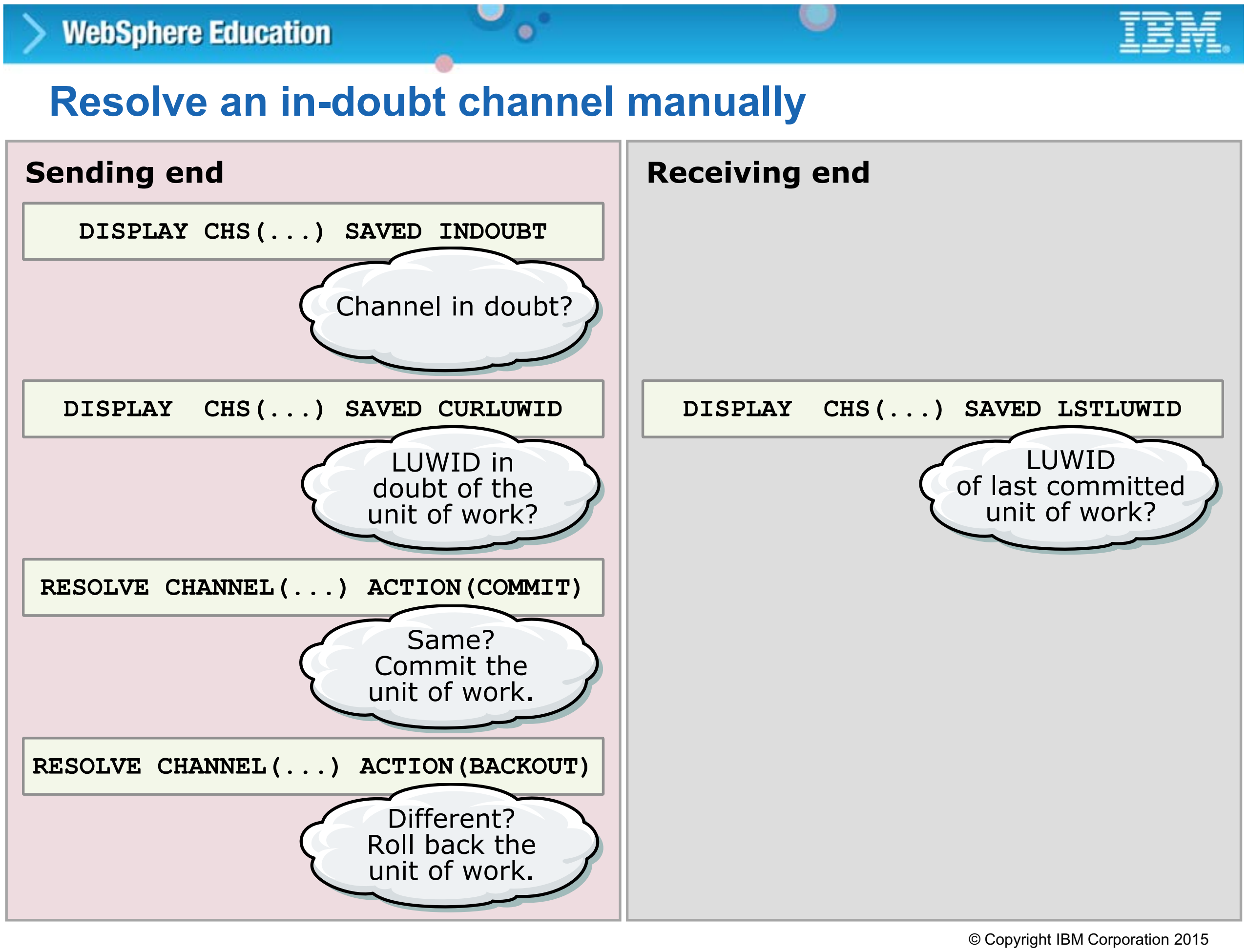


Figure 1-19. Resolve an in-doubt channel manually WM3121.1

Notes:

The procedure to resynchronize a channel manually is:

1. At the sending end of the channel, enter the following command to determine whether the channel is in doubt:

   `DISPLAY CHSTATUS(...) SAVED INDOUBT`

2. If the channel is in doubt, change the following command at the sending end of the channel to determine the LUWID of the in-doubt batch:

   `DISPLAY CHSTATUS(...) SAVED CURLUWID`

   At the receiving end of the channel, change the following command to determine the LUWID of the last committed batch:

   `DISPLAY CHSTATUS(...) SAVED LSTLUWID`

3. If the two LUWIDs are the same, the receiving end already committed the in-doubt messages on their respective destination queues and so the messages must be removed from the transmission queue at the sending end. Removal is achieved by entering the following command:

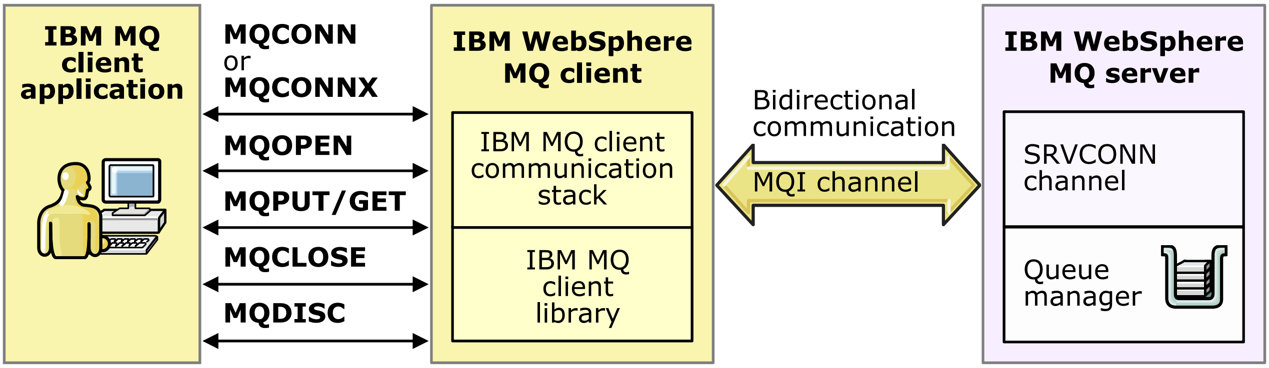

```
RESOLVE CHANNEL(...) ACTION(COMMIT)
```

4. If the two LUWIDs are not the same, the receiving end did not commit the in-doubt messages and the messages are retained on the transmission queue at the sending end. Messages are retrained by entering the command:

```
RESOLVE CHANNEL(...) ACTION(BACKOUT)
```

WebSphere Education

IBM MQ clients



- An IBM MQ client has IBM MQ client libraries installed
- No queue managers or queues are defined
- Uses the same MQI calls as the IBM MQ server
- Flow of messages in an MQI channel is bidirectional
- Transactional capabilities
- No client for z/OS but clients can connect to a z/OS queue manager

Figure 1-20. IBM MQ clients WM3121.1

Notes:

An IBM MQ Client is an option that allows installation of a different set of IBM MQ libraries that run in client mode. The client does not have queues or queue managers. An IBM MQ client application and a server queue manager communicate with each other by using an MQI channel. An MQI channel does not use a channel pair, but rather an MQI connection.

An MQI channel starts when the client application enters an MQCONN or MQCONNX call to connect to the queue manager, and ends when the client application enters an MQDISC call to disconnect from the queue manager.

Note

Do not confuse the term client as used in the general architectural mode, with an IBM MQ Client. An application that is compiled with IBM MQ server libraries might serve as a client application, but uses distributed channel pairs. An IBM MQ Client is compiled with IBM MQ client libraries, uses MQI channels, and must connect to a queue manager SVRCONN channel.

For IBM MQ for z/OS, before IBM MQ V8:

- Separate software had to be installed to enable IBM MQ Client connections for IBM MQ clients. With IBM MQ V8 for z/OS, client connections are included in the IBM MQ z/OS server.
- There was no option to create client channel definition tables (CCDT, covered later in this unit) in the IBM MQ client side. Client definitions in the CCDT had to be downloaded from an IBM MQ server to the IBM MQ client.
- IBM MQ for z/OS does not provide IBM MQ clients, but an IBM MQ client in other platforms can connect to an IBM MQ for z/OS queue manager.

Important

This course focuses on IBM MQ client behavior for IBM MQ V8 only; however, the topic on creating a CCDT on the z/OS side and downloading to a client is applicable to pre-V8 IBM MQ.

WebSphere Education IBM

MQI channel objects

- SVRCONN
  - Channel type that is used by all clients
  - MCAUSER security implications, SVRCONN attributes, and performance considerations
  - Use standard facilities to define and display status on IBM MQ server
- CLNTCONN with runmqsc –n option
  - Correct method to use to create CCDT for IBM MQ for z/OS V8 and later
  - CCDT created at the client side, no CCDT download necessary
- CLNTCONN with CSQUTIL MAKECLNT option
  - MAKECLNT last update before IBM MQ V8
  - CCDT had to be downloaded to client
  - Do not use MAKECLNT for IBM MQ V8 for z/OS and later

Figure 1-21. MQI channel objects WM3121.1

Notes:

On the z/OS queue manager side, the SVRCONN channel is defined for use by clients. It is not necessary to have a CLNTCONN definition to connect to an SVRCONN channel. However, use of a CLNTCONN and a CCDT provide more capabilities, such SSL, or being able to define an alternative queue manager to connect to, perhaps for a failover situation.

The setting of the SVRCONN SHARECNV attribute can be tuned for performance. Having a single socket host multiple connections is called multiplexing, and might enhance the performance of an application. The following sections of the IBM Knowledge Center explains the settings and application considerations for SHARECONV:

- *Tuning client and server connection channels*.
- *MQI client: Default behavior of client-connection and server-connection channels*.

When displaying channel status, SVRCONN-specific attribute CURSHCNV can be used to determine whether applications are using multiplexed connections.

Always use **`client-side`** ***`runmqsc -n`*** with IBM MQ V8 queue managers when use of functions that are introduced by IBM MQ V8 functions is required.

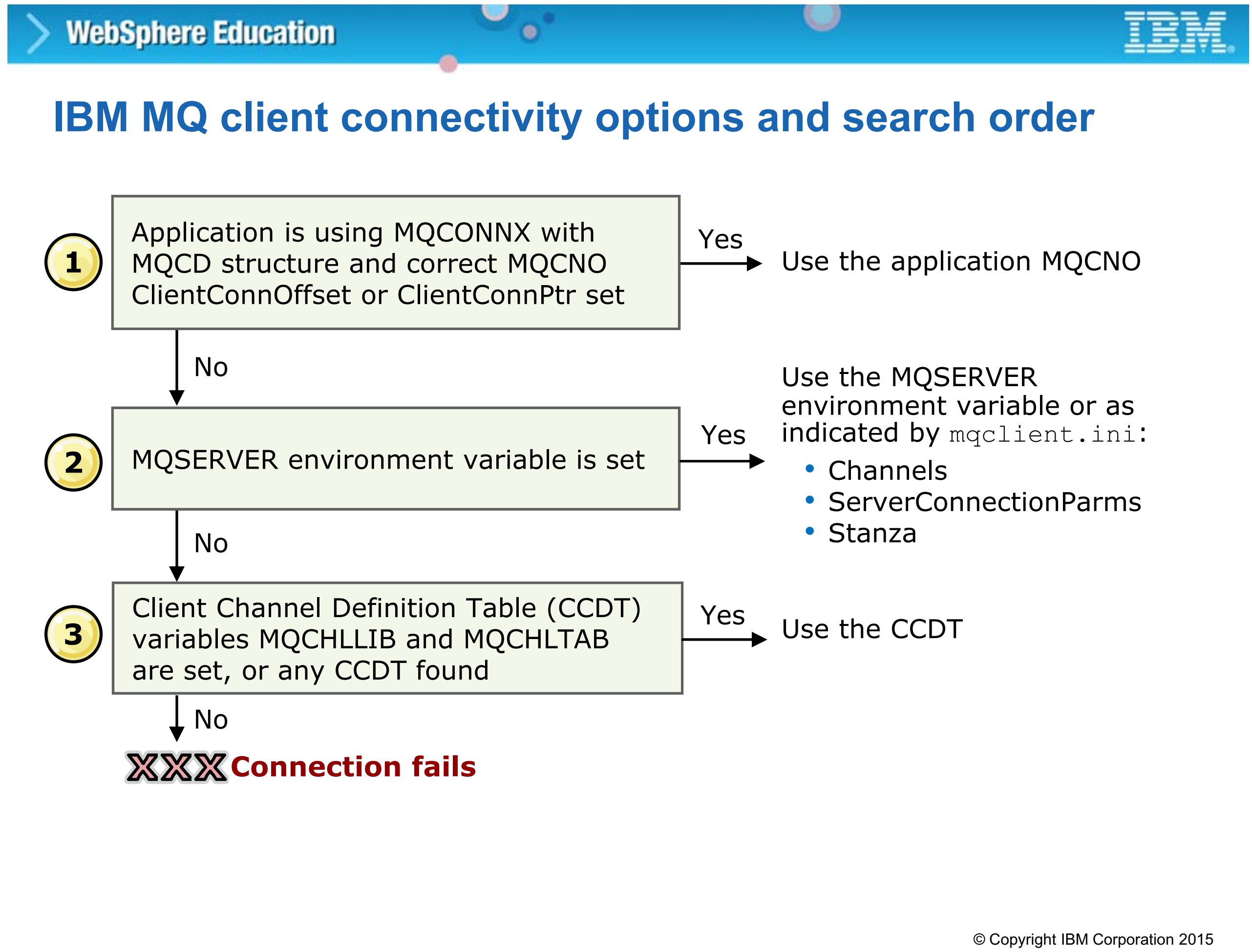


Figure 1-22. IBM MQ client connectivity options and search order WM3121.1

Notes:

IBM MQ client applications have several options to connect to a queue manager.

1. If MQCONNX is used in the application, the connectivity fields of the MQCNO structure are examined. If present, these fields identify structure to be used as the definition of the CCDT. If MQCONN is used, it follows the next options.
2. If the MQSERVER environment variable is set, the channel information is used to connect.
3. If the MQCHLLIB and MQCHLTAB environment variables are set, the CCDT referred to by these variables are used to connect.
4. If no information is found, the client looks for an `mqs.ini` file for the location of the CCDT.

If no information is found, the client looks for the CCDT in the default location according to the platform-specific convention for IBM MQ CCDT locations on the client side.

WebSphere Education

IBM MQ clusters

- Simplified administration
  - Reduce number of remote queues, transmit queues, and channel definitions
- Workload balancing
  - Same queue can be hosted in several queue managers
  - Route around failures
- Scalable
- Contribute to high availability
- MQPUT to local queue manager if queue exists locally
  - If the queue is not found locally, an algorithm is used for target queue selection
- MQGET done to local queue manager
- Publish/subscribe can use cluster

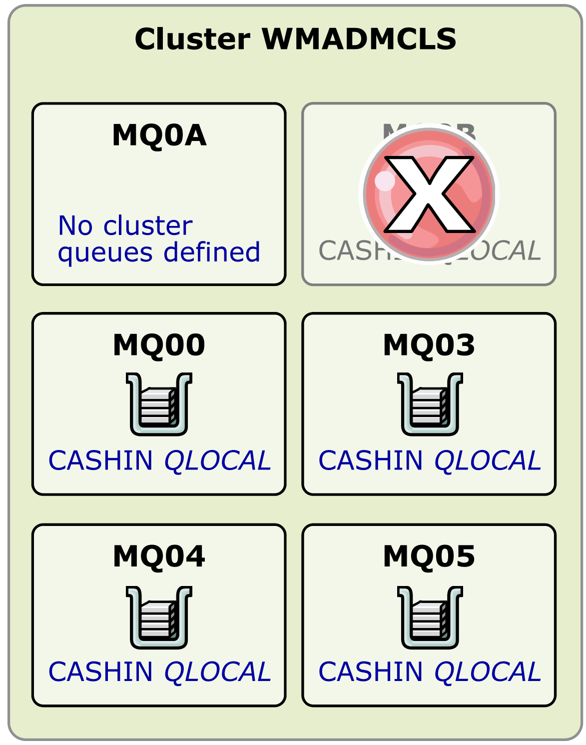


Figure 1-23. IBM MQ clusters WM3121.1

Notes:

An IBM MQ cluster is a grouping of queue managers where queue managers share knowledge of each other's queues and can use automatic channels to reach these queues. After a queue manager "joins" a cluster, it learns about the queues in other clusters.

If an application needs to put a message to the CASHIN queue from MQ0A, this queue would be available in queue managers MQ00, MQ03, MQ04, and MQ05. IBM MQ recognizes the member queue manager that is not available and exclude that queue manager from the potential target queue managers.

As you look at how the clusters are defined, the reduction in the number of IBM MQ object definitions that are required becomes apparent.

The slide qualifies the statement "`contribute to high availability`" because a cluster is not a high availability solution. Messages in cluster queues for a cluster queue manager that is not available are not accessible unless the queue manager is part of a queue-sharing group or configured for high availability. However, clustering does contribute to high availability, scalability, and workload balancing for new incoming requests.

Queue-sharing groups are discussed in a later unit. A queue part of a queue-sharing group (called a shared queue) can be clustered.

When working with clusters, the significant part is the MQOPEN and the MQPUT, which is where cluster workload balancing occurs, particularly the MQOPEN.

MQGET calls are always done to the local queue manager.

WebSphere Education

IBM MQ cluster components

- Repositories
  - Collection of information about the cluster
  - Cluster queue managers hold a full or a partial repository
- Queue manager system support objects
  - SYSTEM.CLUSTER.COMMAND.QUEUE
  - SYSTEM.CLUSTER.REPOSITORY.QUEUE
- CLUSRCVR channels
- CLUSSDR channels
  - Defined channel should point to only one full repository
  - Communicate any cluster-related changes to the full repository
- Transmission queues
  - SYSTEM.CLUSTER.TRANSMIT.QUEUE
  - Alternative cluster transmit queue
- Clustered queues

Figure 1-24. IBM MQ cluster components WM3121.1

Notes:

IBM MQ clusters rely on several components. This slide omits a most important one, which is the cluster or repository manager. In distributed systems. The repository is a process that is called `amqrrmfa`. In z/OS, the repository manager is part of the channel initiator.

Repositories:

- The repository is where all information about the cluster is stored. There are two types of repositories.
  - A full repository, which is held by a queue manager with the REPOS queue manager attribute set to the cluster name.
  - A partial repository, which is held by a member queue manager. Full repositories are critical. There should always be two queue managers that are designated as full cluster repositories and the CLUSSDR channel for these two queue managers should point to each other.
- Full repositories have information about the entire cluster, while partial repositories keep information about those objects that are of interest to what they need to process.
- If one full repository is available, it is possible for a cluster to continue functioning. That is, although the optimal is to have the two full repository queue managers available, should one

queue manager be brought down for maintenance, the cluster is able to operate with the second full cluster repository queue manager active.

- When a member queue manager defines a queue, it tells the repository that it connects to this new queue.

Support objects:

The queue managers have two SYSTEM.* queues that support the cluster.

- The `SYSTEM.CLUSTER.REPOSITORY` queue, as the name implies, keeps the cluster repository. This queue should never be empty for an active cluster.
- The `SYSTEM.CLUSTER.COMMAND.QUEUE`. This queue receives commands and information processed by the repository manager and should normally be empty. When messages accumulate in this queue, it is a good indication that the repository manager is not picking them up. If there are any cluster problems, you know to check whether the channel initiator started a task for any errors. If you are working with a distributed administrator, you can also check whether the `amqrrmfa` process is running, or is not running.

Cluster channels:

Understanding the channels is key to learning about clusters. There are two cluster-related channel definitions:

- CLUSRCVR channels. Unlike distributed RCVR channels, the CLUSRCVR has the connection information. This CLUSRCVR connection information is what cluster member queue managers use to create a dynamic `CLUSSDRx` channel to the queue manager owning the CLUSRCVR. The dynamic CLUSSDR channel is of the type CLUSSDRA or CLUSSDRB. When using clusters, a member queue manager that just joined a cluster checks the repository for the information on a queue manager that it needs to send messages to, and uses the connection name in the CLUSRCVR to create a dynamic channel to the required queue manager.
- CLUSSDR channels. A queue manager that joins the cluster should have one, and only one CLUSSDR channel that points to *one* of the two full cluster repository queue managers.
- Each queue manager that joins a cluster has one CLUSRCVR and one CLUSSDR channel, and has the ability to create dynamic **`CLUSSDRx`** channels to other cluster member queue managers.

What is the “x” in `CLUSSDRx`?

- If a local queue manager creates a dynamic channel to a remote queue manager, and the CLUSSDR in that local queue manager is *not* pointing to the target remote queue manager, the CLUSSDR channel displays as a CLUSSDRA.
- When this local queue manager has a CLUSSDR channel pointing to the target remote queue manager, the dynamic channel that is created to that remote queue manager where CLUSSDR is pointing displays as CLUSSDRB.

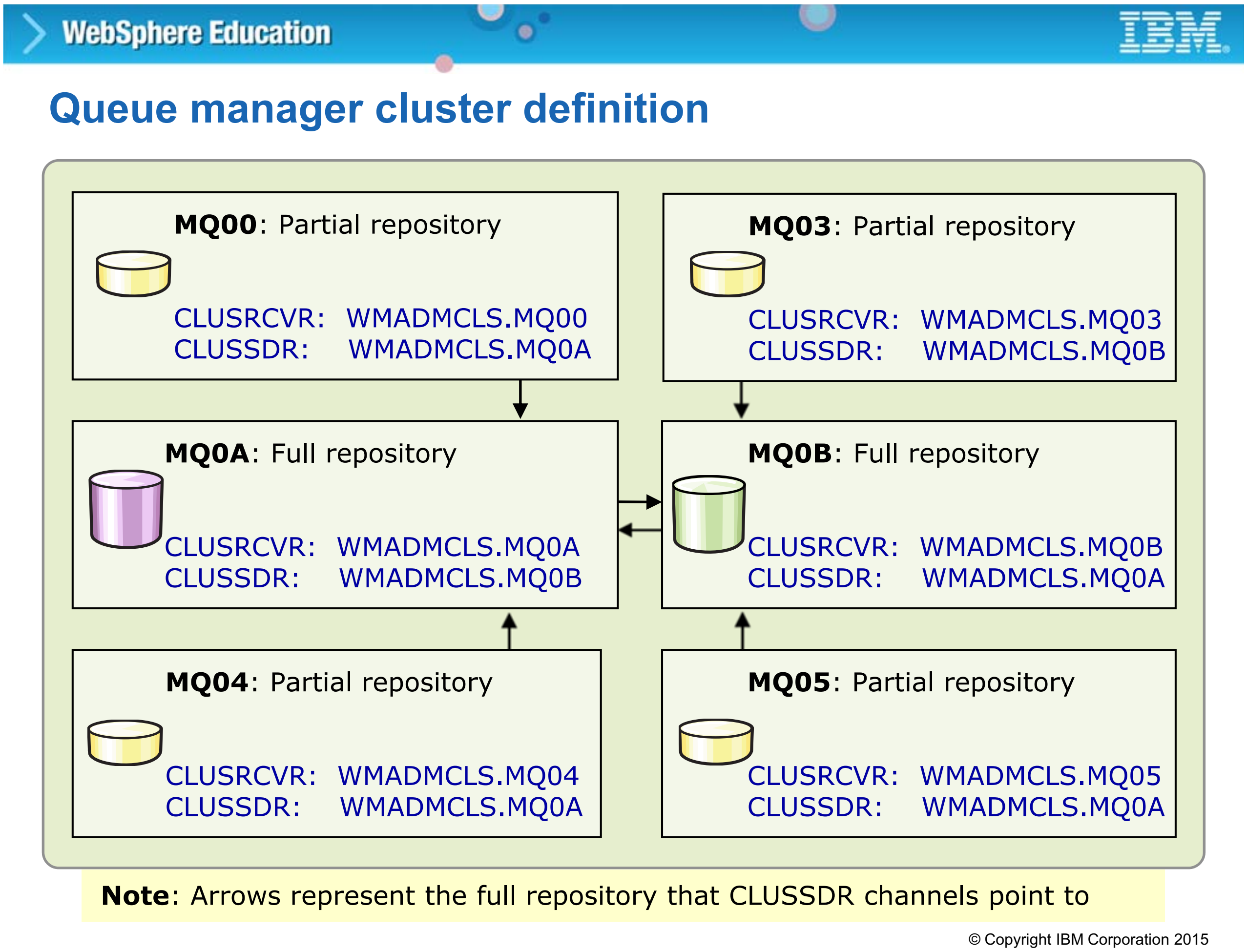


Figure 1-25. Queue manager cluster definition WM3121.1

Notes:

Before defining a cluster, it is important to draw a plan and determine what queue managers host the full cluster repositories. For the WMADMCLS cluster, MQ0A and MQ0B are designated as full cluster repositories. MQ0A and MQ0B has:

- The queue manager REPOS attribute set to the cluster name, WMADMCLS
- The CLUSRCVR channel pointing to their own connection information
- *One* CLUSSDR channel pointing to each other

The rest of the queue managers joining the cluster have:

- The CLUSRCVR channel pointing to their own connection information
- One CLUSSDR channel pointing to *one* of the full repository queue managers
- No changes to the queue manager object

The naming convention that is used for the channels includes the cluster name as the first node, which represents one of the cluster "best practices".

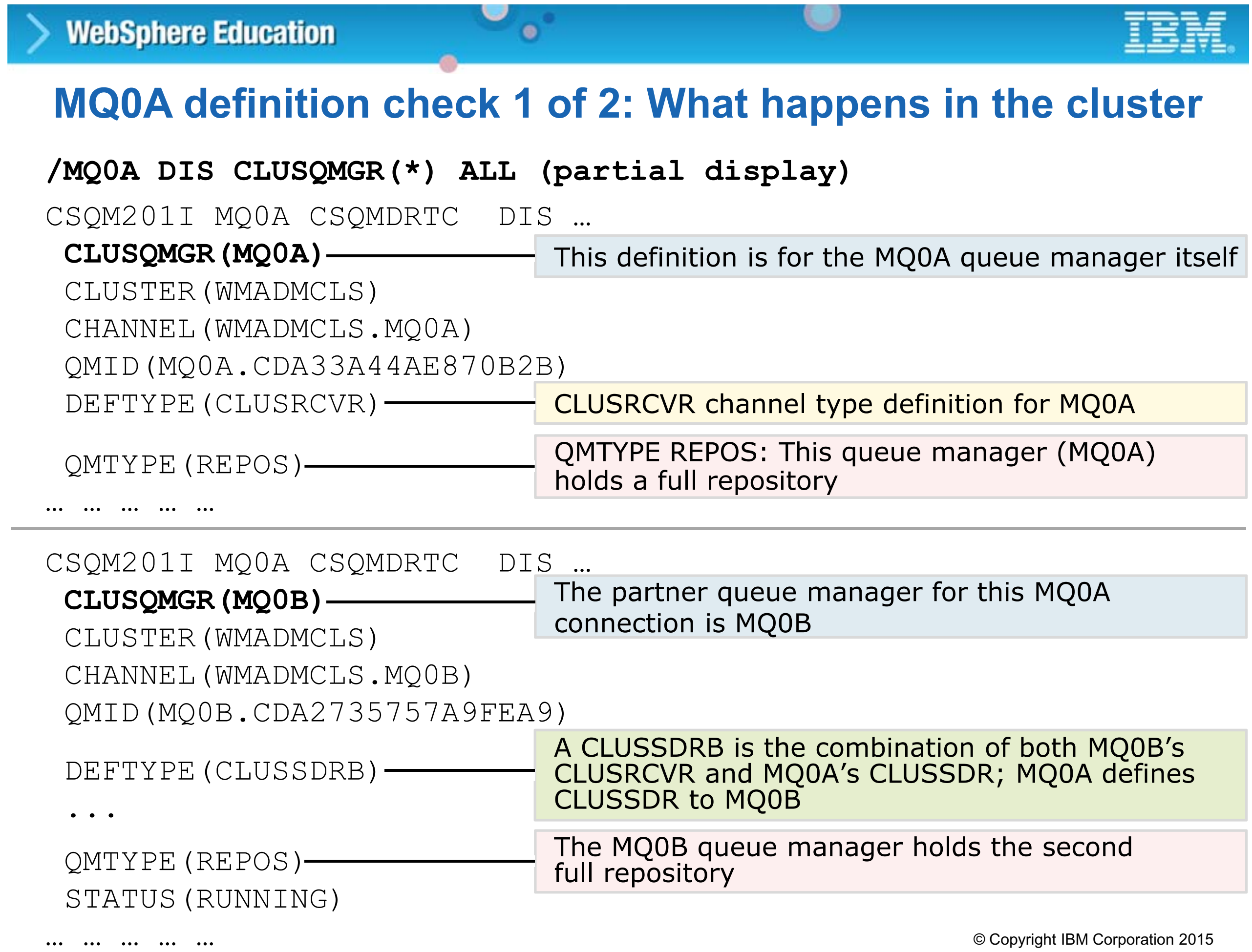


Figure 1-26. MQ0A definition check 1 of 2: What happens in the cluster WM3121.1

Notes:

From one of the full cluster repository queue managers, you enter the **`DIS CLUSQMGR(*) ALL`** command. You use MQ0A. The output from the command spans two slides. What do you check for?

- Immediately after the cluster is created, you expect to see the cluster channels with a `RUNNING` status. For the MQ0A display, it looks good.
- If errors have a “SYSTEM.TEMP.xxx” name as the object, there is a problem. These error entries are usually:
  - `CLUSQMGR(SYSTEM.TEMPQMGR.xxxx.xx.xxx)`
  - `QMID(SYSTEM.TEMPUUID.xxxx.xx.xxx)`
  - No CLUSSDR type channels should be displayed. All should be CLUSSDRA or CLUSSDRB. A CLUSSDR in the DIS CLUSQMGR indicates a problem.

In this case, all looks well.

Reminder

When an error is made in the connectivity information of the cluster channel, the correction is similar to distributed channels. If you see a SYSTEM.TEMP.xxxx.xx.xxx message, the first place to check is the CONNAME attribute for the CLUSRCVR and the CLUSSDR.

Review the notes that are embedded in the slide for the selected `DIS CLUSQMGR` attributes. By the end of the display in this slide, you know that the cluster connection from MQ0A to MQ0B is good, MQ0A and MQ0B are both full repositories (QMTYPE - REPOS), and MQ0B created a dynamic CLUSSDRB channel to MQ0A. The dynamic channel to MQ0A is in `RUNNING` status. Why is this CLUSSDR channel type from MQ0A to MQ0B a CLUSSDRB?

WebSphere Education IBM

MQ0A definition check 2 of 2: What happens in the cluster

```
CSQM201I MQ0A CSQMDRTC  DIS …
  CLUSQMGR(MQ00)
  CLUSTER(WMADMCLS)
  CHANNEL(WMADMCLS.MQ00)
  QMID(MQ00.CDA33FC1E2935036)
  DEFTYPE(CLUSSDRA)
  ...
  QMTYPE(NORMAL)
  STATUS(RUNNING)
  VERSION(08000000)
  XMITQ(SYSTEM.CLUSTER.TRANSMIT.QUEUE)
  TRPTYPE(TCP)
  CONNAME(mvsmc11.ilsvpn.ibm.com(1600))
… … … … …
```

The partner queue manager for this MQ0A connection is MQ00

CLUSSDRAs show a dynamic connection from MQ0A to MQ00’s CLUSRCVR; MQ0A has no defined CLUSSDR to MQ0A

The MQ00 queue manager holds a partial repository

/MQ0B DIS CLUSQMGR(*) ALL output is similar to MQ0A and bypassed for brevity

Figure 1-27. MQ0A definition check 2 of 2: What happens in the cluster WM3121.1

Notes:

You now look at the rest of the DIS CLUQMGR display. In this part of the output you see that:

- The information is what cluster queue manager MQ0A learned about the MQ00 cluster queue manager.
- Channel WMADMCLS.MQ00 is a dynamically defined CLUSSDRA type channel from MQ0A to MQ00. This means that queue manager MQ0A learned how to find MQ00 from the information in the repository because the CLUSSDR for MQ0A points to MQ0B.
- Since MQ0A has no CLUSSDR defined to MQ00, then the DEFTYPE for the dynamic channel that MQ0A creates to MQ00 is a CLUSSDRA type channel.
- The information that MQ0A has about MQ00 also indicates that MQ00 does not hold a full repository, and the QMTYPE shown for MQ00 is NORMAL, not REPOS.

You looked at the output of **DIS CLUSQMGR** entered from MQ0A. The display from MQ0B is skipped for brevity, as it is similar to MQ0A. It is the view from a full repository queue manager pointing to another full repository.

But where is the information for the MQ03 queue manager?

WebSphere Education

MQ03 definition check: What happens in the cluster

```
/MQ00 DIS CLUSQMGR(*) ALL (partial display)
CSQM201I MQ03 CSQMDRTC  DIS CLUSQMGR DETAILS 636
 CLUSQMGR(MQ03)
 CLUSTER(WMBADMCLS)
 CHANNEL(WMADMCLS.MQ03)
 QMID(MQ03.CDB68FBADA5634A2)
 DEFTYPE(CLUSRCVR)
 QMTYPE(NORMAL)
 … … …
 CONNAME(mvsmc11.ilsvpn.ibm.com(1603))
 … … …
CSQM201I MQ03 CSQMDRTC  DIS CLUSQMGR DETAILS 637
 CLUSQMGR(SYSTEM.TEMPQMGR.mvsmc11.ilsvpn.ibm.com(1622))
 CLUSTER(WMADMCLS)
 CHANNEL(WMADMCLS.MQ0B)
 QMID(SYSTEM.TEMPUUID.mvsmc11.ilsvpn.ibm.com(1622))
 DEFTYPE(CLUSSDR)
 QMTYPE(REPOS)
 … … …
 STATUS(INACTIVE)
 XMITQ(SYSTEM.CLUSTER.TRANSMIT.QUEUE)
 TRPTYPE(TCP)
 CONNAME(mvsmc11.ilsvpn.ibm.com(1622))
 … … …
```

This definition is for the MQ03 queue manager itself

CLUSRCVR channel type definition for MQ03

Can you spot the problem?

Channel DEFTYPE should be a CLUSSDRB to MQ0B

Always check new cluster queue managers before proceeding

Figure 1-28. MQ03 definition check: What happens in the cluster WM3121.1

Notes:

You enter **DIS CLUSQMGR** for MQ03. Note what happens:

- MQ03 is directing its CLUSSDR channel to the MQ0B full repository, but the display for the CLUSSDR* QMTYPE has three obvious problems:
  - CLUSQMGR indicates a SYSTEM.TEMPQMGR.xxx name, a red flag.
  - QMID also indicates SYSTEM.TEMOUUID.xxx name, another red flag.
  - The channel DEFTYPE shows CLUSSDR, another red flag. This should be a CLUSSDRB because there is an explicitly defined CLUSSDR to the full repository, MQ0B.

You check all definitions, but the connection information of the CLUSSDR to MQ0B looks correct, and so does the connection information in the CLUSRCVR channel.

Important

Don't confuse `DIS CLUSQMGR(*) DEFTYPE` with `DIS CHS(*) CHLTYPE`. If `DIS CHS` is used, it is OK to see a `CHLTYPE` equal `CLUSSDR`, however a `DEFTYPE` of `CLUSSDR` indicates a problem in the cluster.

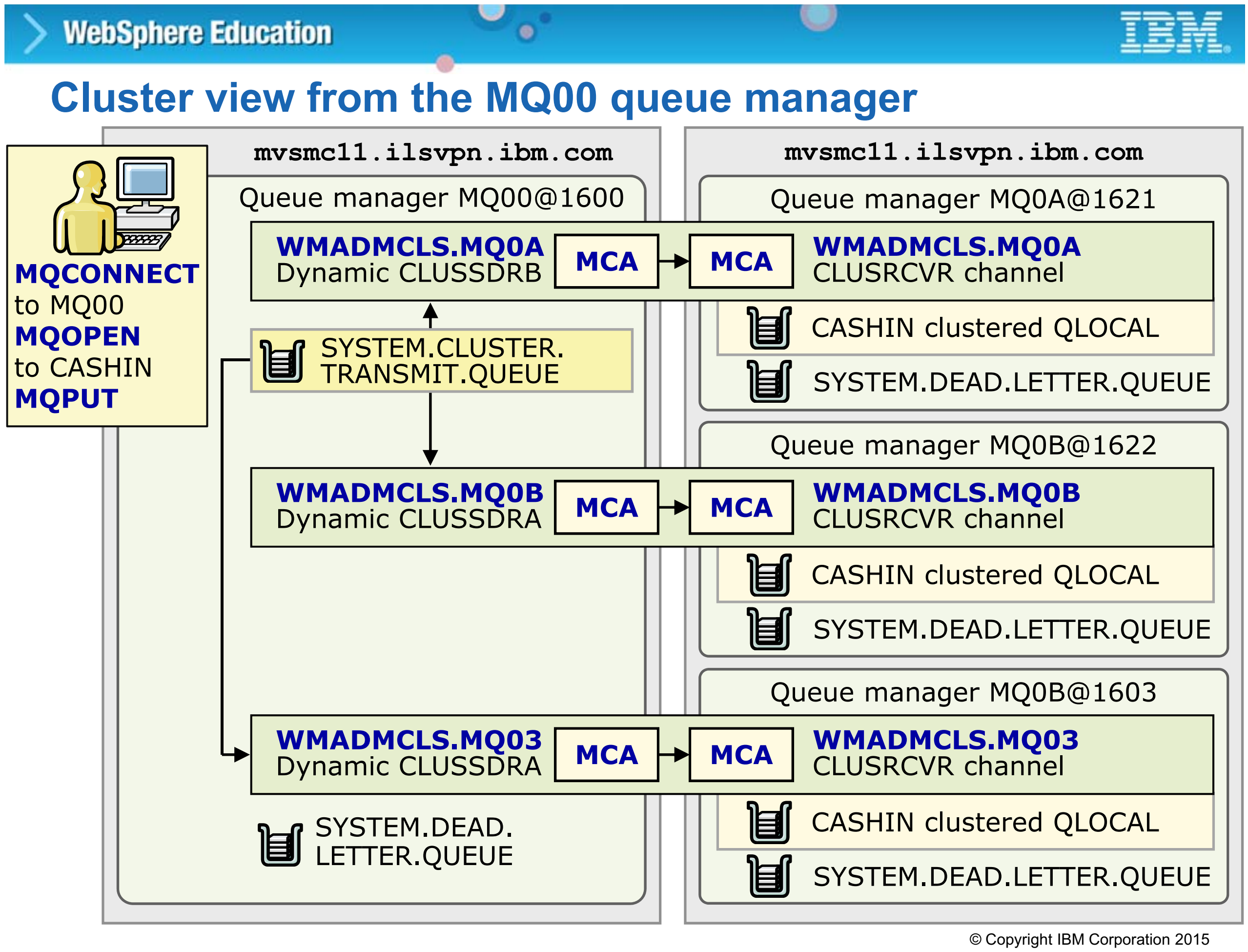


Figure 1-29. Cluster view from the MQ00 queue manager WM3121.1

Notes:

How does clustering compare to the distributed channels? What path do cluster messages take?

Other than the workload balancing, following the path of a message in a cluster queue is similar to working with distributed queuing except that there are some additional cluster queues where the message might go. Assumptions that are taken in this walk through are (1) that the queue manager is not set to allow the messages to be placed outside the local version of the CASHIN queue if one is available. This setting is explained in a later slide. (2) CASHIN is not hosted by the MQ00 queue manager. (3) the queue managers all use the default transmit queue.

- An application enters MQOPEN of clustered queue CASHIN for MQ00. The CASHIN queue is not hosted by the MQ00 queue manager. Cluster workload balancing determines the best candidate queue manager hosting the CASHIN cluster queue to be the target for the message.

Assume that the message goes to CASHIN at MQ0A. The message is first placed on SYSTEM.CLUSTER.TRANSMIT.QUEUE at MQ00. MQ00 has its CLUSSDR channel pointed to MQ0A, so a dynamic channel of type CLUSSDRB called WMADMCLS.MQ0A picks up the message and if no problems are encountered, places this message in the CASHIN queue at MQ0A.

- Should there be a problem with the channel, the message would wait in the SYSTEM.CLUSTER.TRANSMIT.QUEUE at MQ00.
- Should there be a different problem, the message might be forwarded to the dead letter queue identified for either MQ00 or MQ0A depending on the problem.

This trajectory is similar to tracing the route of a message in a distributed channel. If the workload algorithm selected MQ0B or MQ03 as the target for the message, then a CLUSSDRA instead of a CLUSSDRB dynamic channel would carry the message to its target.

If you are using a dedicated transmit queue other than SYSTEM.CLUSTER.TRANSMIT.QUEUE, then instead of looking in SYSTEM.CLUSTER.TRANSMIT.QUEUE you would check in the dedicated named transmit queue. Using different transmit queues is discussed in a later slide.

If it is critical to find which of the clustered queues the message was sent to, the status for the potential candidate cluster queues can be displayed, for instance. In the display, it is assumed that queue monitoring is active for the queue manager and the queue.

```
MQ0A DIS QSTATUS(CASHIN) ALL
QSTATUS(CASHIN)
TYPE(QUEUE)
OPPROCS(0)
IPPROCS(0)
CURDEPTH(17)
UNCOM(NO)
MONQ(LOW)<=== queue monitoring enabled, but need to check queue manager
QTIME(0,0)
MSGAGE(3138292)
LPUTDATE(2014-10-15) <=== date message was last put
LPUTTIME(19.52.35)  <=== date message was last put
LGETDATE()
```

Since you did get the last put date and time populated in the DIS QSTATUS display it is obvious that the queue manager was enabled for monitoring information, however, the command to check whether the queue manager is enabled is shown here in case it is required.

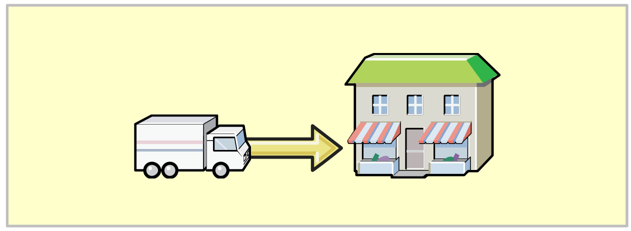
```
MQ0A DIS QMGR MONQ

QMNAME(MQ0A)
MONQ(LOW)   <=== queue manager is enabled
```

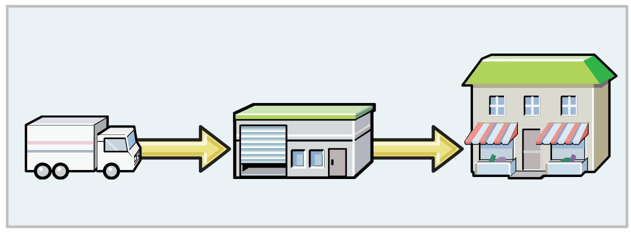
WebSphere Education

Messaging styles

- Point-to-point application
  - Sends messages to a pre-defined destination
  - Application does not need to know where the destination is because IBM MQ locates the target by using object definitions



- Publish/subscribe application
  - Publishes messages to an interim destination according to a topic
  - Interested recipients subscribe to the topic
  - No explicit connection between publishing and subscribing applications



Unless publish/subscribe is mentioned, subsequent topics apply to the point-to-point messaging style.

Figure 1-30. Messaging styles WM3121.1

Notes:

There are two messaging styles, point-to-point and publish/subscribe.

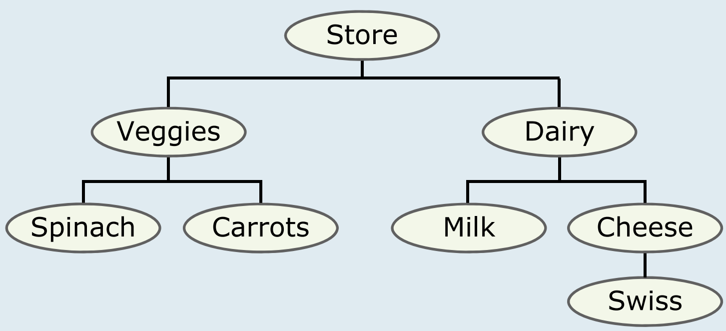
With point-to-point messaging, you know where the message is going. In publish/subscribe, the producers of messages send or publish messages to an interim place where the consumers or subscribers obtain, or subscribe to a topic or topic string that is linked to the messages.

The sending, or publishing of messages and the subscription or getting of messages is done by using the topic tree as an intermediary. It is not apparent where the message is going, or how it shows up, the publish/subscribe process handles those details. Publish/subscribe uses distributed and cluster channels, but the source and destination of the messages is determined by the intermediary publish/subscribe process. Publish/subscribe adds an extra level of decoupling to the application.

WebSphere Education

Publish/subscribe basic components

Topic tree



Topic strings

Store/Dairy/Cheese/Swiss
Store/Veggies/Spinach
Store/Veggies

Topic objects

(Optional)

Publishing application

Use:
MQOPEN
topic Store/Veggies/Spinach
MQPUT

- Point-to-point opens a queue
- Publish/subscribe opens a topic

Subscribing application

Use:
MQSUB verb to Store/Veggies

Figure 1-31. Publish/subscribe basic components WM3121.1

Notes:

At the center of publish/subscribe is the topic tree. A topic tree is a hierarchical structure that contains and organizes topics. A topic tree's organization is influenced by how the topics are created. The topic tree might be compared to file system with directories and subdirectories.

Topics can be created by using the create topic commands or by specifying the topic for the first time in a publication or subscription. Topics are structured into topic strings. Topic strings have separate categories that are separated by a slash ('/') character.

When messages are sent in a publish/subscribe application, they are published, or put to a topic instead of a queue. A topic can be an explicitly defined object, or a topic that is dynamically created by a publish or a subscribe to the topic string. Published messages are called publications.

To consume these publications, applications subscribe to the topics or topic strings. The subscriptions can be an explicitly defined object, or a dynamic subscription that is created by an application.

So you have the topic tree, topics, publications and publishers, and subscribers and subscriptions.

To better understand today's publish/subscribe implementation, you look at publish/subscribe over the years.

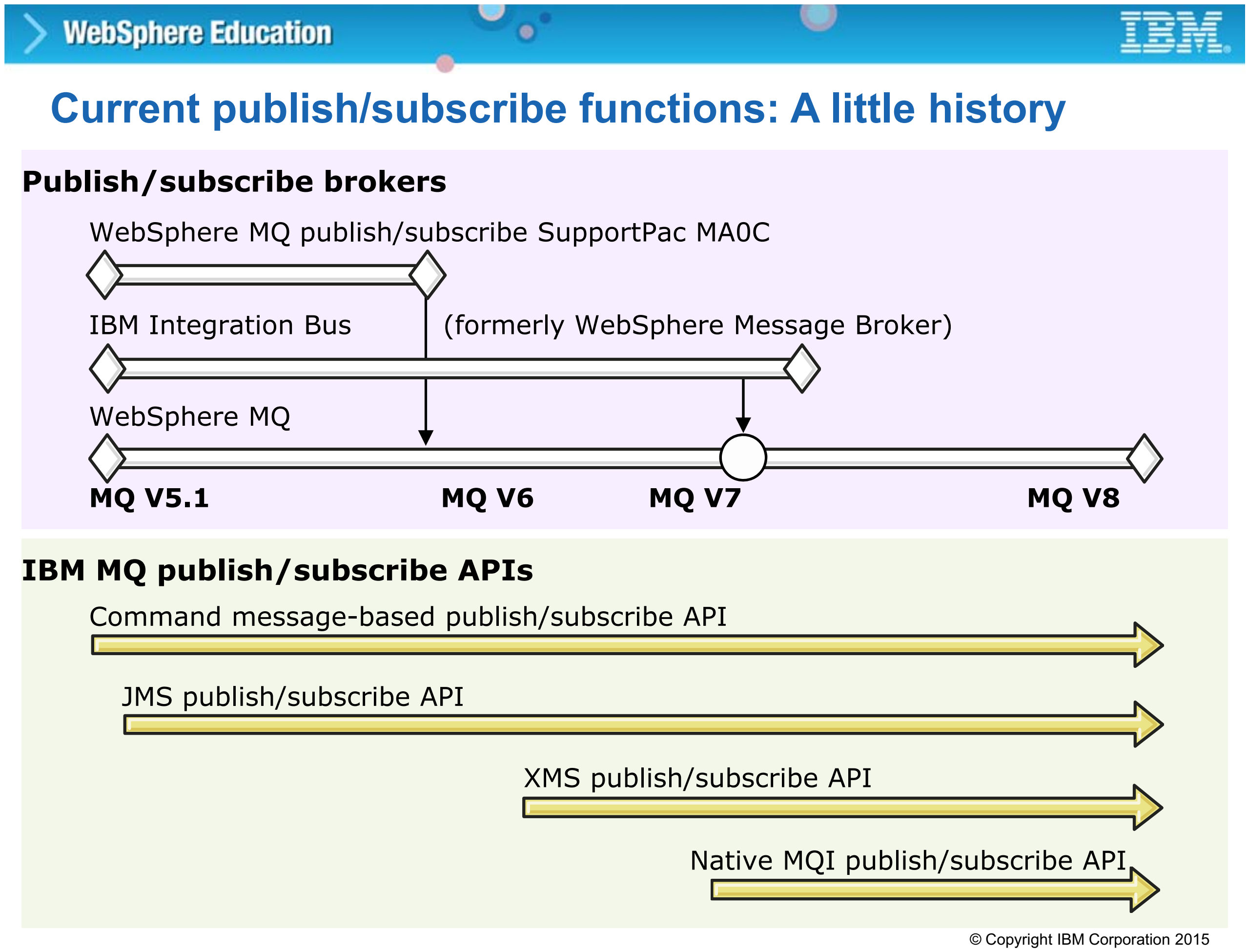


Figure 1-32. Current publish/subscribe functions: A little history WM3121.1

Notes:

The implementation and terminology of publish/subscribe can be confusing. Queue managers have a "mode" or PSMODE parameter. There is mention of "managed queues" and "queued publish/subscribe". Why such terminology?

Publish/subscribe started out as IBM MQ SupportPac MA0C. Publishing and subscribing with the original SupportPac involved by using the RFH command message-based API to send subscription and publication requests. WebSphere Message Broker (WMB) also had a separate publish/subscribe mechanism and used WMB capabilities to do the publish/subscribe actions. Both the SupportPac and WMB referred to the publish/subscribe engine as a "broker".

As publish/subscribe evolved, it was included in the IBM MQ product. The publish/subscribe API was also changed and augmented. While this course does not address migration of the "old" publish/subscribe applications, it is good to keep in mind that the need to address or handle a migration might surface if you work with an existing application that uses the "old" API and broker function.

WebSphere Education IBM

Publish/subscribe functionality

- PSMODE queue manager attribute controls publish/subscribe versions
  - ENABLED: The default, which indicates that both the publish/subscribe engine and the queued publish/subscribe are running
  - COMPAT: Publish/subscribe engine active, and queued publish/subscribe stopped
  - DISABLED: Both publish/subscribe engine and queued publish/subscribe are stopped

```
CSQX000I MQ11 CSQXJST IBM WebSphere MQ for z/OS V8.0.0
CSQX001I MQ11 CSQXJST Channel initiator starting
 … … … … …
 +CSQT975I MQ11 CSQXDPSC Distributed Pub/Sub Controller has started
 … … … … …
 +CSQT975I MQ11 CSQXDPSC Distributed Pub/Sub Fan Out Task has started
 +CSQT975I MQ11 CSQXDPSC Distributed Pub/Sub Command Task has started
 +CSQT975I MQ11 CSQXDPSC Distributed Pub/Sub Publish Task has started
 … … … … …
 +CSQT806I MQ11 CSQXFCTL Queued Pub/Sub Daemon started
 +CSQX023I MQ11 CSQXLSTT Listener started, 664
```


Figure 1-33. Publish/subscribe functionality WM3121.1

Notes:

PSMODE is a queue manager attribute across the z/OS and distributed platforms that controls the type of publish/subscribe functionality that is enabled in a queue manager.

In the snapshot of the channel initiator start console messages, you can see how the publish/subscribe engine starts, then later the “old” publish/subscribe, that is the “queued” publish/subscribe process also starts.

The predominant modes are:

- ENABLED: Use this mode if you are not sure what the mix of applications is. It supports the old and new publish/subscribe, within any limitations that are documented in IBM MQ Knowledge Center.
- COMPAT: The “old” publish/subscribe is not active. Only the new engine is started. Use this mode if you are sure that there are no old publish/subscribe applications running.
- DISABLED: If publish/subscribe is not being used, startup of the publish/subscribe process can be stopped, *however*, other software components outside the applications, such as WebSphere Message Broker or IBM MQ Managed File Transfer might require queued publish/subscribe to be active. Always confirm what runs in the environment before setting PSMODE to DISABLED.

WebSphere Education

Highly available WebSphere MQ topologies

- WebSphere MQ clusters offer scalability by having multiple queue managers to service a request, however
  - Clustering provides availability for new incoming requests
  - When a queue manager fails, messages in transit for that queue manager are marooned
- If the queue manager is a critical resource such as a gateway, a failover capability compatible with the platform where IBM MQ is running must be implemented.
- Explore use of queue-sharing groups for the z/OS platform.

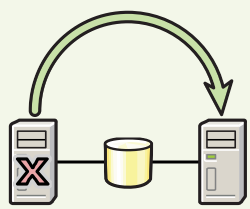


Figure 1-34. Highly available WebSphere MQ topologies WM3121.1

Notes:

WebSphere Education

What are shared queues

- Queues whose messages are kept in a coupling facility
- Queue managers in the same z/OS sysplex access the same shared queues by using connectivity through the coupling facility
- When several z/OS queue managers share the same queues, they are called a queue-sharing group
- Provide high availability in an IBM MQ infrastructure

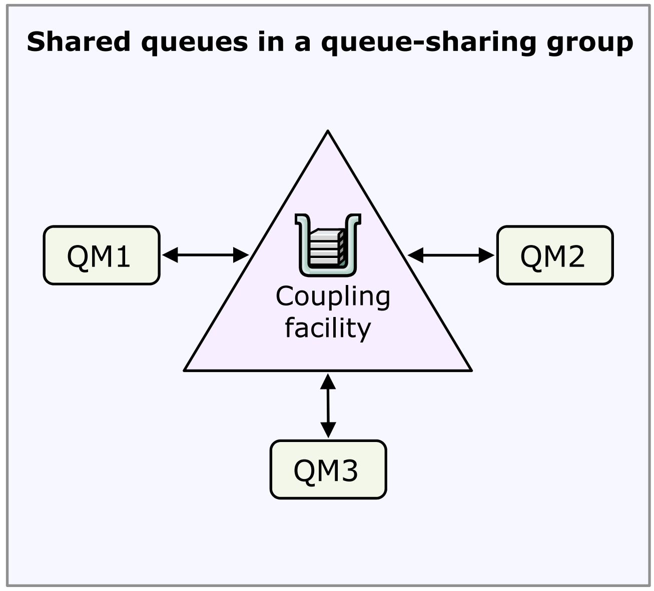


Figure 1-35. What are shared queues WM3121.1

Notes:

A shared queue is a special type of a local queue. The messages on a shared queue are stored in a list structure of the coupling facility (CF), and thus might be accessed from any queue manager in the sysplex.

The definition objects for the shared queues, together with other shareable definition objects, are stored in a DB2 shared database. Non-shared, regular local queues are still there and continue to be served by the local page data sets for each queue manager.

Note

Developing applications for shared queues is similar to developing any other IBM MQ application, however, section *Application Programming with shared queues* in the IBM MQ Knowledge Center that must be reviewed as part of a plan to use shared queues for a new application, or migrate an existing application to shared queues. Two key items to review are: special details on use of correlation or message ID, and dynamic queue consideration, where only permanent dynamic queues are allowed, however, the entire section should be reviewed.

WebSphere Education

IBM MQ for z/OS V8.0

IBM MQ z/OS and extensions	IBM MQ	MQMFT	MQAMS
IBM MQ for z/OS V8.0	✓		
IBM MQ for z/OS Value Unit Edition (VUE) V8.0	✓		
MQ Managed File Transfer for z/OS V8.0 (MQMFT)		✓	
MQ Advanced Message Security for z/OS V8.0 (MQAMS)			✓
MQ Advanced for z/OS V8.0 (MQADV)		✓	✓

Figure 1-36. IBM MQ for z/OS V8.0 WM3121.1

Notes:

In this class, you work with IBM MQ for z/OS V8.0 base product.

IBM MQ for z/OS Value Unit Edition V8.0 is a differently priced version of IBM MQ for z/OS V8.0.

IBM MQ Advanced for z/OS is a packaging of IBM MQ Managed File Transfer and IBM MQ Advanced Message Security for z/OS.

Note

WebSphere MQ is being renamed IBM MQ. Hence, throughout this course, you still see mention of WebSphere MQ in some parts of the course material.

WebSphere Education IBM

Queue manager on z/OS

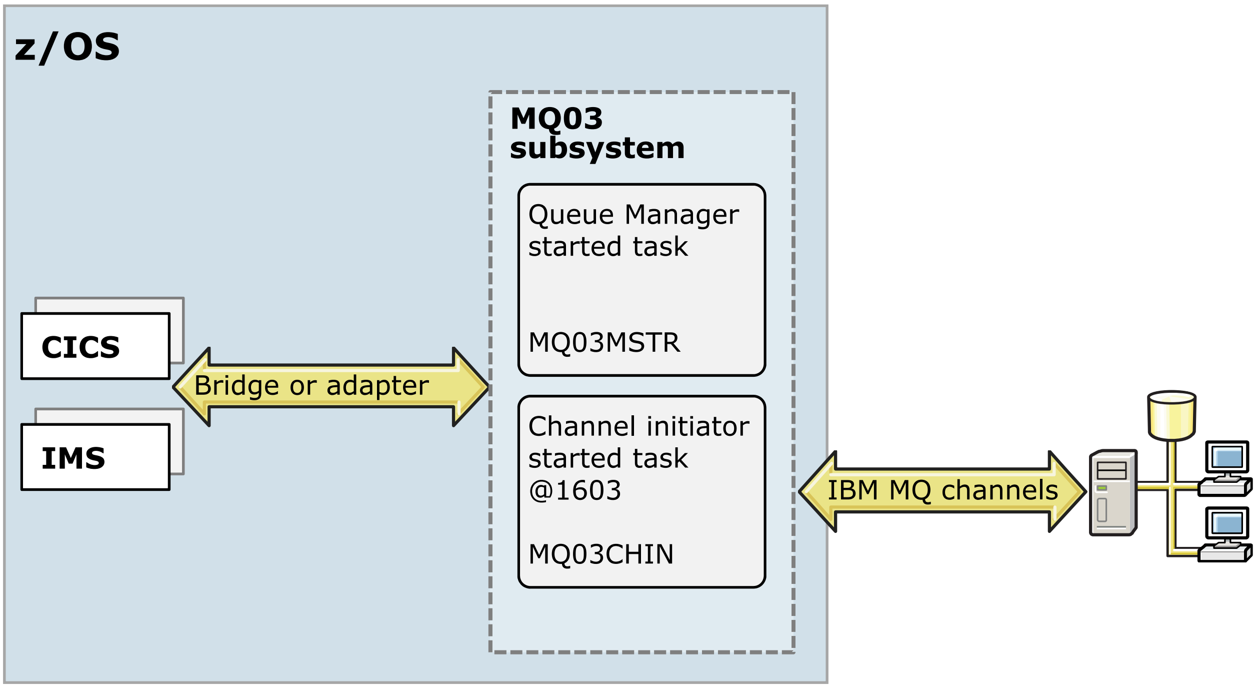


Figure 1-37. Queue manager on z/OS WM3121.1

Notes:

In z/OS, WebSphere MQ runs as a subsystem made up of two started tasks: the queue manager master task, and the channel initiator. The queue manager is given a unique four-character name.

The queue manager provides services to applications and manages its resources. The queue manager is also responsible for managing the data sets it owns, such as the logs and message data sets. It also manages the memory buffers, the channel initiator, and the ability to interface with other subsystems such as CICS or IMS.

WebSphere Education

Installation and configuration

- Collect and review latest documentation for IBM MQ version that is installed to confirm requirements and prepare to install
- Complete SMP/E install
- Perform these actions once for each z/OS system:
  - Identify and update correct z/OS **`SYS1.PARMLIB(IEASYSpp)`** parameters
  - APF authorize the IBM MQ load libraries
  - Update the z/OS link list and LPA
  - Update the z/OS program properties table
- Define the IBM MQ subsystem to z/OS
- Perform queue manager and channel initiator configuration
- Define IBM MQ to a z/OS WLM service class
- If queue-sharing groups are used, perform related tasks
- Implement External Security Manager (ESM) security controls
- Set up the operations and control panels
- Include the IBM MQ dump formatting member for IPCS

Figure 1-38. Installation and configuration WM3121.1

Notes:

Before a queue manager is installed and configured, there are tasks that need to be completed in the z/OS environment.

In this section, you walk through the steps that were completed in advance in the lab environment for this course. You also walk through the section to be completed in the lab, titled *Perform queue manager and channel initiator configuration*.

There are two sources of reference information for the work in this section:

- Program Directory for WebSphere MQ for z/OS V8.0. This document contains key installation information such as SMP/E details, resulting data sets, requirements, and package names.
- Section *Configuring queue managers on z/OS* found on the IBM Knowledge Center.

There can be more than one queue manager in the same z/OS server. When configuring multiple queue managers in the same server, some of the steps that are outlined are done once per server and do not need to be repeated, while other steps are done once per queue manager.

WebSphere Education

Collect and review latest documentation

- Program Directory for WebSphere MQ for z/OS V8
  - Information on components
  - SMP/E instructions
  - Hardware and software requirements
- Check whether there is any IBM MQ for z/OS maintenance to apply
- Configuring queue managers on z/OS documentation from either:
  - WebSphere MQ 8.0.0 IBM Knowledge Center
  - Corresponding pdf download titled Installing IBM WebSphere MQ
- Capacity Planning and Tuning Guide for WebSphere MQ for z/OS V8.0 support pack MP16
- Depending on the functions that are used in your IBM MQ infrastructure, check current and past version z/OS performance support packs

Figure 1-39. Collect and review latest documentation WM3121.1

Notes:

Documentation for IBM MQ is updated frequently. While this course mentions the most recent documentation at the time the course was written, it is important to check that you always have current documentation for the version of IBM MQ you work with.

It might be that sometimes an older document might still apply, as discussed with the information in the IBM MQ performance SupportPacs. Although not listed in this slide, depending on the functions used it might also be helpful to download the IBM MQ performance SupportPacs for the last few IBM MQ versions:

- MP16 Capacity planning and tuning for WebSphere MQ for z/OS
- MP1B WebSphere MQ for z/OS V7.1 Interpreting accounting and statistics data
- MP1H WebSphere MQ for z/OS V7.1 Performance Report
- MP1J WebSphere MQ for z/OS V8.0 Performance Report

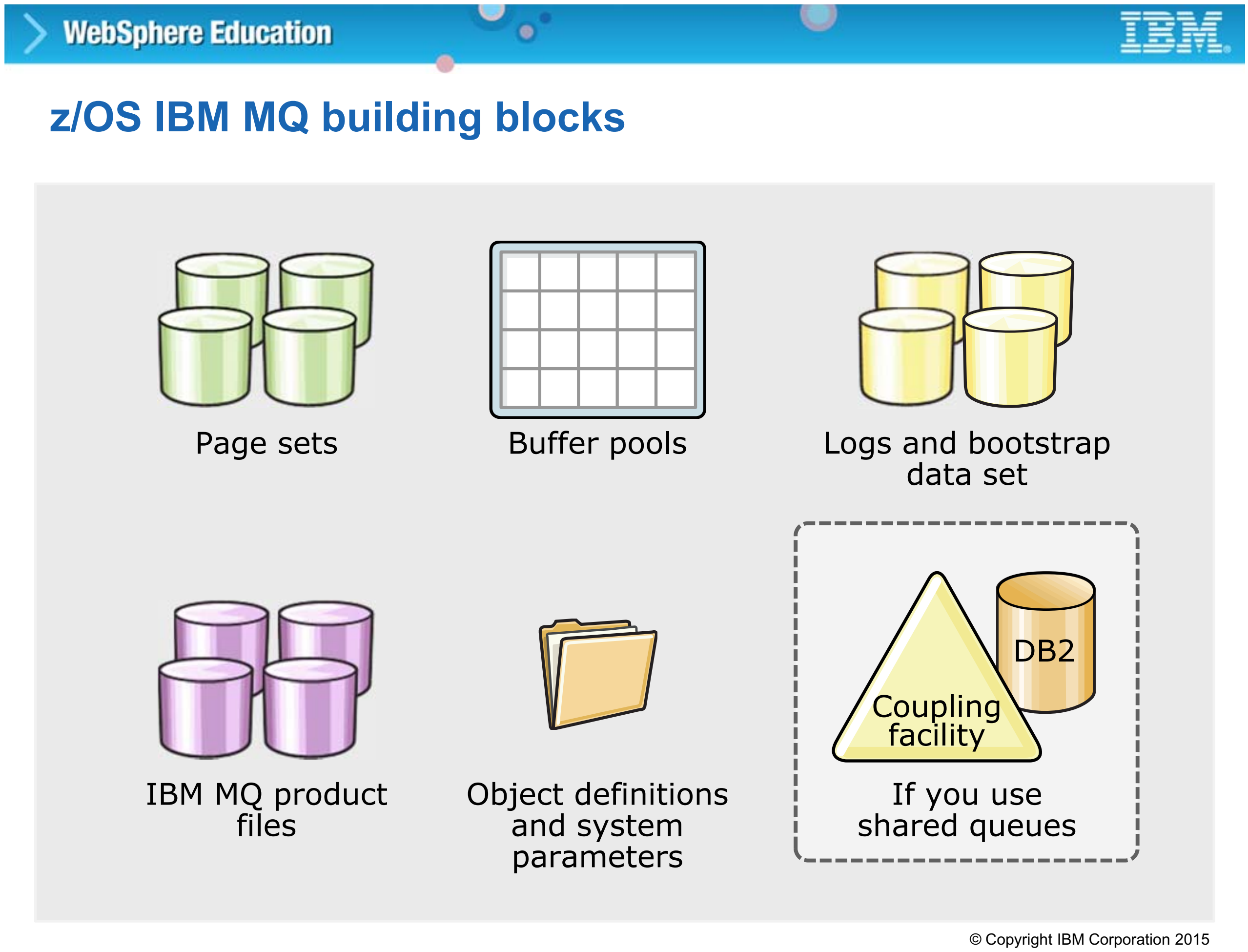


Figure 1-40. z/OS IBM MQ building blocks WM3121.1

Notes:

You start out with the IBM MQ product files, where the IBM MQ software is located after the SMP/E installation. To create a queue manager, you need to define:

- Buffer pools, where messages are held when they arrive.
- Page sets, where messages are written if not retrieved from the queue soon after arriving at the queue.
- Logs, where persistent messages and transactions are recorded, and the resource that is used for restart and recovery. The boot strap data set keeps track of the logs.

The IBM MQ object definitions and properties are an important part of the queue manager definition.

z/OS queue managers are also able to host shared queues and queue manager groups. You discuss shared queues in a later unit.

WebSphere Education

Logging

- IBM MQ logs queue manager information that is needed for restart and recovery in an active log
- The active log is composed of a series of up to 31 data sets
- The bootstrap data set (BSDS) tracks the log data sets
- Logs are archived to maintain available space

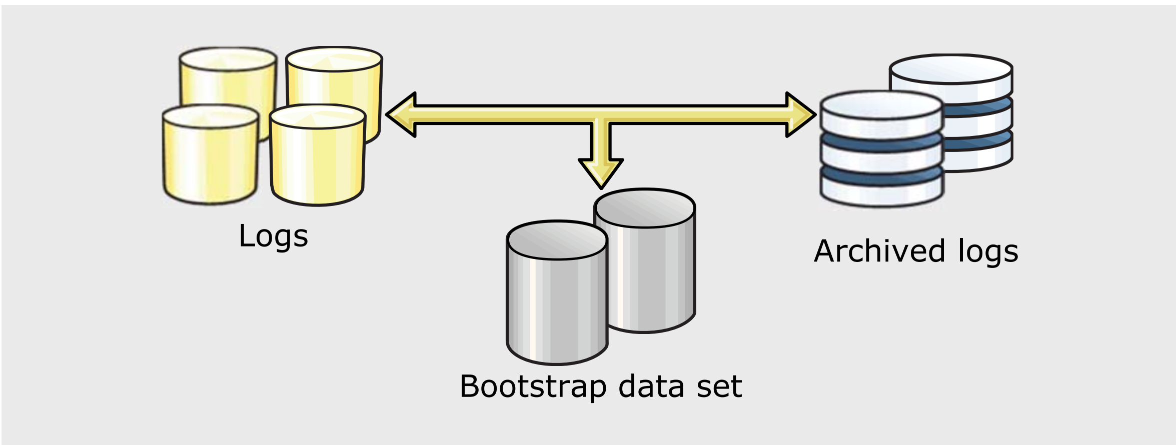


Figure 1-41. Logging WM3121.1

Notes:

Logs are a critical part of IBM MQ. If you look at the queue manager master region started task, you can see how the logs are replayed when a queue manager starts.

On z/OS, the log is a set of up to 31 data sets used in a circular manner that saves persistent messages, IBM MQ objects, and transaction information. Logs do not keep statistics or monitoring information. When it is determined that a log data set is not required for recovery, the log data set is archived to free up space for IBM MQ to continue its work. Without archiving, the logs fill up and IBM MQ stops working due to lack of space. It is critical to ensure that archiving is functional, particularly for production systems. The set of logs available for use is referred to as the active log. Archiving is started in the following situations:

- When an active log data set is full
- Explicitly when the **`ARCHIVE LOG`** command is entered
- If it is detected that a log data set is damaged

The boot strap data set tracks which logs are active and which logs can be archived.

Log data can be written in single mode, or twice in dual mode to mitigate the risk of a data set getting corrupted.

The log is composed of log records. Each byte is accessed by a relative byte address, or RBA. Before IBM MQ V8, some customers exceeded the amount of data that might be logged, that is, their RBA wrapped over the 255 TB, or `x'FFFFFFFFFFFF'` addressing limitation of the 6-byte RBA. When the RBA wrapped, it was necessary to take an extended outage to reset the RBA, sometimes at the risk of losing persistent data.

You look at the IBM MQ V8 8-byte RBA changes in a later unit.

WebSphere Education

Object definitions: CSQ4MSTR in SCSQPROC data set

CSQ4INP2 data set is a concatenation of members such as:

• **CSQ4INSG**	System object definitions: Objects that are prefixed with SYSTEM*.
• **CSQ4INSA**	System object and default rules for channel authentication.
• **CSQ4INSX**	System object definitions.
• **CSQ4INYC**	Clustering definitions: Sample template for objects that are used in clustering.
• **CSQ4INYD**	Distributed queuing definitions: Sample template for distributed objects.
• **CSQ4INYG**	General definitions: Such as dead letter queue, ALTER QMGR, CICS queues.
• **CSQ4INYR**	Storage class definitions, that use multiple page sets for the major classes of message.
• **CSQ4INYS**	Storage class definitions, that use one page set for each class of message.
• **CSQ4INP2** (member)	Include other commands, not necessarily object definitions, to issue when the queue manager is starting. The channel initiator started task can be added here.

Figure 1-42. Object definitions: CSQ4MSTR in SCSQPROC data set WM3121.1

Notes:

Another component of the z/OS queue manager is object definitions. These PDS members are found in the SCSQPROC data set. The members to include are selected depending on what capabilities are included in the queue manager. Some members are used as-is, while other members need to be edited to change variables to the required object names. A subset of these members is used in the first lab exercise.

WebSphere Education

Parameters: CSQ4ZPRM in SCSQPROC data set

- System parameters are included in the CSQ4ZPRM member that includes three macros to be assembled and linked:

  CSQ4ZPRM is an assembly and link job for three macros:

• **CSQ6LOGP**	Logging parameters such as the size of input buffer storage for active and archive data sets, OFFLOAD to turn archiving on or off, and TWOBSDS for using a single or dual BSDS
• **CSQ6ARVP**	Archiving parameters such as the block size of the archive data set, space allocation, retention period, and security settings
• **CSQ6SYSP**	System parameters such as LOGLOAD and tracing options

- A subset of these parameters can be altered when the queue manager is running by using MQSC commands SET SYSTEM, SET LOG, and SET ARCHIVE

Figure 1-43. Parameters: CSQ4ZPRM in SCSQPROC data set WM3121.1

Notes:

A critical component of the queue managers is the parameter module, usually referred to as the "ZPARMS".

It is here that the logging and archiving parameters are set.

The CSQ6SYSP macro has other critical queue manager parameters, including the OPMODE setting to enable the above the bar buffer pools and 8-byte RBA.

WebSphere Education

IBM MQ MSTR region partial restart messages

```
CSQJ127I MQ00 SYSTEM TIME STAMP FOR BSDS=2014-09-18 12:41:05.31
CSQJ001I MQ00 CURRENT COPY 1 ACTIVE LOG DATA SET IS  933
DSNAME=MQ00.LOGCOPY1.DS01, STARTRBA=0000000000000000
ENDRBA=000000000437FFF
CSQJ001I MQ00 CURRENT COPY 2 ACTIVE LOG DATA SET IS  934
DSNAME=MQ00.LOGCOPY2.DS01, STARTRBA=0000000000000000
ENDRBA=000000000437FFF
CSQJ099I MQ00 LOG RECORDING TO COMMENCE WITH  935
STARTRBA=000000000005E000
CSQJ034I MQ00 CSQJW007 END OF LOG RBA RANGE IS FFFFFFFFFFFFFFFF
CSQP007I MQ00 Page set 0 uses buffer pool 0
… … … … …
CSQR001I MQ00 RESTART INITIATED
CSQR003I MQ00 RESTART - PRIOR CHECKPOINT RBA=000000000005C946
CSQR004I MQ00 RESTART - UR COUNTS -  946
IN COMMIT=0, INDOUBT=0, INFLIGHT=0, IN BACKOUT=0
CSQI049I MQ00 Page set 0 has media recovery  947
RBA=0000000000052A51, checkpoint RBA=0000000000052A51
… … … … …
```


Figure 1-44. IBM MQ MSTR region partial restart messages WM3121.1

Notes:

This display shows messages that are produced when the queue manager started task becomes active. A few details can be observed.

One of the first things to notice is the importance of the BSDS and the log data sets, and how the BSDS and logs are used to restart the queue manager.

The queue manager also identifies the active log data set.

Also, of interest is that this queue manager is converted to use the 8-byte RBA as message CSQJ341I shows the end of the log range as x'FFFFFFFFFFFFFFFF'.

WebSphere Education IBM

Channel initiator: CHIN

What takes place in the CHIN started task:

```
CSQX011I MQ03 CSQXGIP Client Attachment available
CSQX151I MQ03 CSQXSSLI 0 SSL server subtasks started, 0 failed
CSQX410I MQ03 CSQXREPO Repository manager started
CSQT975I MQ03 CSQXDPSC Distributed Pub/Sub Controller has start
CSQX022I MQ03 CSQXSUPR Channel initiator initialization complete
CSQT806I MQ03 CSQXFCTL Queued Pub/Sub Daemon started
CSQX004I MQ03 CSQXSPRM Channel initiator is using 30 MB of local
632 storage,1678 MB are free
CSQX251I MQ03 CSQXSTRL Listener started, TRPTYPE=TCP INDISP=QMGR
CSQX023I MQ03 CSQXLSTT Listener started, 634 port 1603 address *,
```


Figure 1-45. Channel initiator: CHIN WM3121.1

Notes:

The display shows a partial list of channel initiator messages.

The first item that you see is the handling of the client attachment, followed by SSL. With IBM MQ V8, there is no more client attach facility. The function is built in the z/OS code. You work with the SSL server subtasks in the security lab.

The channel initiator hosts the IBM MQ Cluster repository, and also publish/subscribe. Notice that two different publish/subscribe processes are started, these processes are discussed in the publish/subscribe unit.

The listener is a key component that is handled by the channel initiator. When the started task came up, the cluster channels that are associated with the queue manager become active.

WebSphere Education

IBM MQ V8 Channel initiator error

```
+CSQX213E MQ00 CSQXSUPR Communications error,  054
  function 'CSFPPRF' RC=12 reason=00000000
```

- Error occurs due to client password obfuscation capability. New in V8.
- Error surfaces even if SSL is not enabled.
- To correct the error enable ICSF.

Figure 1-46. IBM MQ V8 Channel initiator error WM3121.1

Notes:

A new error might surface in the IBM MQ V8 z/OS channel initiator started task.

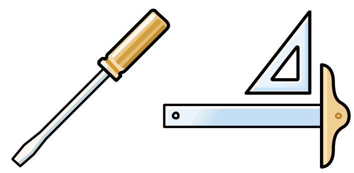
This error occurs because IBM MQ needs the ability to read an incoming password, even if SSL is not enabled. The password is obfuscated at the client side, and needs to be uncovered on the z/OS side. For this purpose z/OS needs to have ICSF started. There was no need to provide access to the cryptographic hardware.

If you see this error in your shop, starting ICSF should correct it.

WebSphere Education

IBM MQ for z/OS administrative capabilities

- Equivalent options to perform common actions on IBM MQ objects such as define, alter, display, delete, and show status
- ISPF operations and control panels
- IBM MQ Scripts (MQSC)
  - CSQUTIL COMMAND option
  - SDSF command prefix string
  - ISPF panels option 8
- IBM MQ Explorer
- Programmable command formats
- CSQUTIL
- IBM MQ for z/OS Utilities



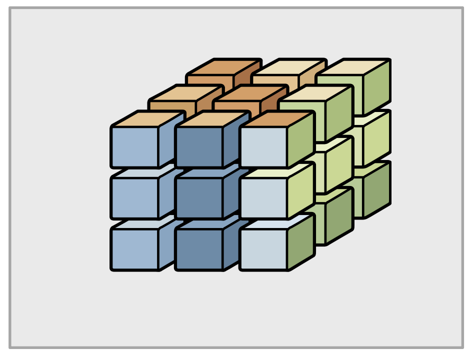


Figure 1-47. IBM MQ for z/OS administrative capabilities WM3121.1

Notes:

IBM MQ for z/OS offers several options for administration. Which option you use depends on what is most appropriate for the task at hand and your personal preference. For instance:

- When creating a large number of objects, maybe associated with a new application, you might want to enter these definitions in a PDS member and define them using the COMMAND option of CSQUTIL.
- If making a single change such as changing MONQ on the queue manager and a single queue, you might find it quicker to use the console with the command prefix string, or CPF.
- If the command string is too long to fit in the log panel, and you use ISPF, you can opt to enter the same command in the ISPF panel command option, which provides ample space.
- If you are checking for the number of messages in a queue or the time a message last arrived in the queue, you might find IBM MQ Explorer quicker to use.

Programmable command formats, or PDF are more suited for automation and their use includes application code. So it is less likely to be used unless a specific application is coded. PCFs were included for completeness, as this is an option available for IBM MQ for z/OS.

WebSphere Education

CSQUTIL command option: Defining IBM MQ objects

```
//MQ00UTIL JOB ,WM30V1,CLASS=A,MSGCLASS=H,NOTIFY=&SYSUID
//* -------------------------------------------------------
//* RUN THE CSQ UTILITY PGM FOR PROCESSING COMMANDS
//* -------------------------------------------------------
//COMMAND EXEC PGM=CSQUTIL,PARM='MQ00'
//STEPLIB    DD DSN=SYS1.WMQ800.SCSQANLE,DISP=SHR
//           DD DSN=SYS1.WMQ800.SCSQAUTH,DISP=SHR
//SYSPRINT   DD SYSOUT=*
//SYSIN      DD *,DCB=BLKSIZE=80
  COMMAND
/*
//CSQUCMD    DD DISP=SHR,DSN=TSM0000.WM30V1.SETUP.CNTL(CSQUTL1)
```

- CSQUCMD DD contains MQSC commands

```
DEFINE QLOCAL(WM302.INPUT)
DEFINE QALIAS(WM302.ALIAS) TARGET(WM302.INPUT)
… … … … …
```


Figure 1-48. CSQUTIL command option: Defining IBM MQ objects WM3121.1

Notes:

This slide is a JCL example showing how to use CSQUTIL to create object definitions. One item to look out for is administrators that use the same JCL and only change the object name, for instance to create a queue. At times, this practice can be error prone if, for instance, you create a queue for an unintended storage class.

Carefully review definitions that are used in a repeated basis to prevent unexpected problems.

WebSphere Education

Security areas

- Identification: Determine the identity of a person that is using a system or resource
- Authentication: Proving the person or resource that is being identified is who they claim to be
- Access control/authorization*: Limiting access to resources to only those people who need it
- Confidentiality: Protection of sensitive information
- Data integrity: Ability to detect tampering with critical data
- Auditing: Ability to determine whether any unauthorized access was done or attempted

Note: *The terms *access control* and *authorization* are used interchangeably.

Figure 1-49. Security areas WM3121.1

Notes:

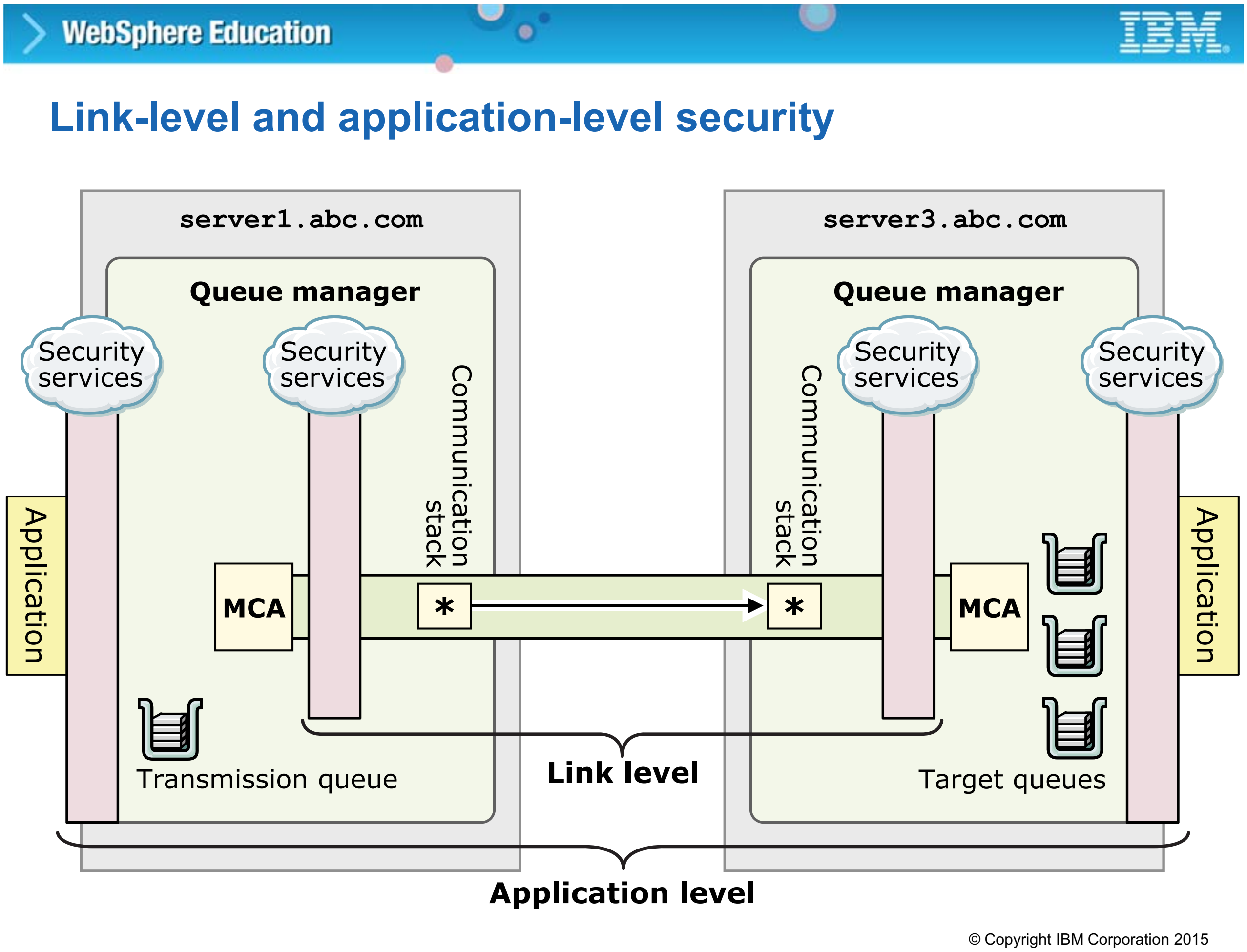


Figure 1-50. Link-level and application-level security WM3121.1

Notes:

Application-level security applies to data at rest, while link-level security applies to data in transit.

When you first set up your z/OS queue manager, you used an external security manager (ESM) to set up switch profiles for your queue manager. You also used the ESM to permit or inhibit different types of access to queues, topics, and channels.

WebSphere Education

Link-level security: Connection authentication

- Queue manager attribute CONNAUTH
- Two new types of authentication objects:
  - OS repository
  - LDAP
- Defaults
  - New queue managers default to using OS system authentication
  - Migrated queue managers have CONNAUTH disabled
- Use of MQSAMP_USER_ID environment variable with distributed sample programs sets ID and allows password entry
- On the application side:
  - Code change is required to add security structure
  - Option to use mqccred exit

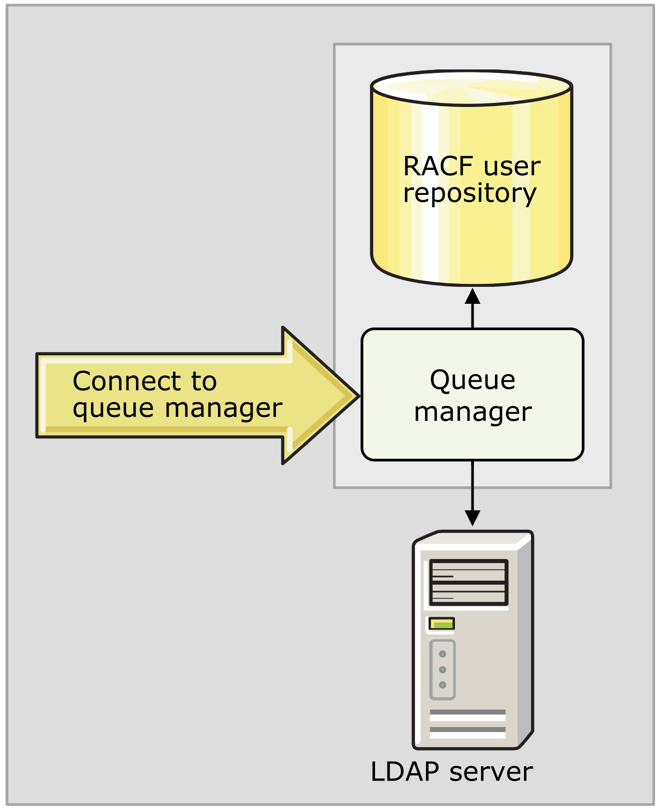


Figure 1-51. Link-level security: Connection authentication WM3121.1

Notes:

IBM MQ V8 introduced the connection authentication feature, which provides the ability to authenticate a user to IBM MQ with an operating system ID or LDAP.

Applications must make code changes or use the `mqccred` exit to use connection authentication. There are ways to test an IBM MQ server's connection authentication settings by using the sample programs with a special environment variable.

WebSphere Education IBM

CONNAUTH settings: Defaults

```
/MQ00 DIS QMGR CONNAUTH
CSQM201I MQ00 CSQMDRTC  DIS QMGR DETAILS
QMNAME(MQ00)
CONNAUTH(SYSTEM.DEFAULT.AUTHINFO.IDPWOS)
 END QMGR DETAILS
```

```
MQ00 DIS AUTHINFO(*) ALL
CSQM293I MQ00 CSQMDRTC 3 AUTHINFO FOUND MATCHING
CSQM201I MQ00 CSQMDRTC  DIS AUTHINFO DETAILS 090
AUTHINFO(SYSTEM.DEFAULT.AUTHINFO.CRLLDAP)
AUTHTYPE(CRLLDAP)
… … … … …
AUTHINFO(SYSTEM.DEFAULT.AUTHINFO.IDPWOS)
AUTHTYPE(IDPWOS)
QSGDISP(QMGR)
ADOPTCTX(NO)
CHCKCLNT(OPTIONAL)
CHCKLOCL(OPTIONAL)
FAILDLAY(1)
```


Figure 1-52. CONNAUTH settings: Defaults WM3121.1

Notes:

When working with IBM MQ applications in server or client bindings on the distributed side, it is possible to test the connection authentication settings in the server with the usual IBM MQ sample programs, amqsput, amqsputc, amqsget, amqsgetc, amqsbcg, or amqsbcgc:

For queue managers in distributed platforms, you can use the `MQSAMP_USER_ID` environment variable to set the user before running:

```
set MQSAMP_USER_ID=youruserID
```

If the `MQSAMP_USER_ID` is blank and the corresponding `AUTHINFO CHCKCLNT` object option is set to OPTIONAL, no checks are done, however, if the user ID is supplied, then a password is required. If the `AUTHINFO CHCKCLNT` option is set to REQUIRED, then both an ID and password are required. If working with the local queue manager, the `AUTHINFO CHCKLOCL` attribute is used instead of the `AUTHINFO CHCKCLNT` attribute.

New connectivity test sample amqscnxc:

Use -u to provide the ID. The program then prompts for a password.

```
amqscnxc -x server1.ibm.com(1625) -c WM302.SVRCONN -u yourUserID MQ00
```


WebSphere Education

IBM MQ security

- Identification
  - OS user IDs
  - Context
  - Channel authentication
  - Channel security exits
- Authentication
  - Original OS logon
  - MQCONNX MQI call
  - Channel security exits
  - Channel authentication records (CHLAUTH)
  - Secure sockets layer (SSL)
  - Connection authentication
- Access control
  - RACF switches
  - Put authority
  - Channel authentication records
  - Firewalls
- Confidentiality
  - Advanced Message Security (AMS)
  - Secure sockets layer (SSL)
  - Channel exits
- Data Integrity
  - Advanced Message Security (AMS)
  - Secure sockets layer (SSL)
- Auditing
  - Events

Figure 1-53. IBM MQ security WM3121.1

Notes:

IBM MQ provides functionality to address different aspects of security. In an earlier slide, you learned how IBM MQ provides connection authentication.

As you see in Unit 3, you can use channel authentication rules. By using connection authentication and channel authentication, you do not need to use a security exit to provide similar functions.

In Unit 2, you learn how to address confidentiality and authorization by using SSL/TLS.

WebSphere Education

Unit summary

- Describe IBM MQ basic components
- Explain how to resolve common channel problems
- Distinguish between IBM MQ servers and IBM MQ clients
- Describe clusters and cluster channels
- Contrast the IBM MQ messaging styles
- Describe IBM MQ architectural options
- Summarize how IBM MQ is implemented in z/OS
- Identify the administrative options for IBM MQ on z/OS
- Describe security options for IBM MQ channels and clients

Figure 1-54. Unit summary WM3121.1

Notes:

WebSphere Education

Checkpoint questions

1. True or False. If a channel is in current state, it is always consuming resources and has a process or thread running.

2. What is the correct method to create a client channel definition table (CCDT) connection from a V8 IBM MQ client to a z/OS queue manager?
   a. Use the **`runmqsc`** tool in a distributed IBM MQ server installation and import it to the client
   b. Run the CSQUTIL with the MAKECLNT option in z/OS and import the CCDT to the client
   c. Use the `mqclient.ini` to hold the CLNTCONN channel definition
   d. Use **`runmqsc -n`** utility from the V8 IBM MQ client to add the CLNTCONN to the CCDT

3. Which of these statements about an IBM MQ cluster is most accurate?
   a. Contribute to scalability and reduce the number of IBM MQ object definitions
   b. Provide high availability and contribute to scalability
   c. Contribute to scalability and high availability and reduce IBM MQ administration requirements
   d. All of the above

4. True or False. Use of the publish/subscribe messaging style further decouples connections between two peer sides.

Figure 1-55. Checkpoint questions WM3121.1

Notes:

Write your answers here:

1.

2.

3.

4.

WebSphere Education

Checkpoint answers (1 of 2)

1. True or False. If a channel is in current state, it is always consuming resources and has a process or thread running.
   Answer: False. Only *active* current channels consume resources. Current channels in STOPPED, STARTED, or RETRYING *inactive* sub states do not consume resources.

2. What is the correct method to create a client channel definition table (CCDT) connection from a V8 IBM MQ client to a z/OS queue manager?
   a. Use the **`runmqsc`** tool in a distributed IBM MQ server installation and import it to the client
   b. Run the CSQUTIL with the MAKECLNT option in z/OS and import the CCDT to the client
   c. Use the `mqclient.ini` to hold the CLNTCONN channel definition
   d. Use **`runmqsc -n`** utility from the V8 IBM MQ client to add the CLNTCONN to the CCDT

   Answer: d. V8 IBM MQ clients must use the client version of the `runmqsc` utility, which is activated by issuing `runmqsc -n` in the client side, to create a CCDT to an IBM MQ V8 queue manager. CSQUTIL MAKECLNT should not be used between IBM MQ V8 clients and queue managers.

Figure 1-56. Checkpoint answers (1 of 2) WM3121.1

Notes:

WebSphere Education

Checkpoint answers (2 of 2)

3. Which of these statements about an IBM MQ cluster is most accurate?
   a. Contribute to scalability and reduce the number of IBM MQ object definitions
   b. Provide high availability and contribute to scalability
   c. Contribute to scalability and high availability and reduce IBM MQ administration requirements
   d. All of the above

   Answer: a. An IBM MQ cluster contributes to scalability but is not considered a high availability solution. Messages can be marooned in a queue manager. A queue manager failover approach or queue-sharing groups in z/OS are high availability solutions.

4. True or False. Use of the publish/subscribe messaging style further decouples connections between two peer sides.
   Answer: True. Publish/subscribe provides additional decoupling between two sides.

Figure 1-57. Checkpoint answers (2 of 2) WM3121.1

Notes:

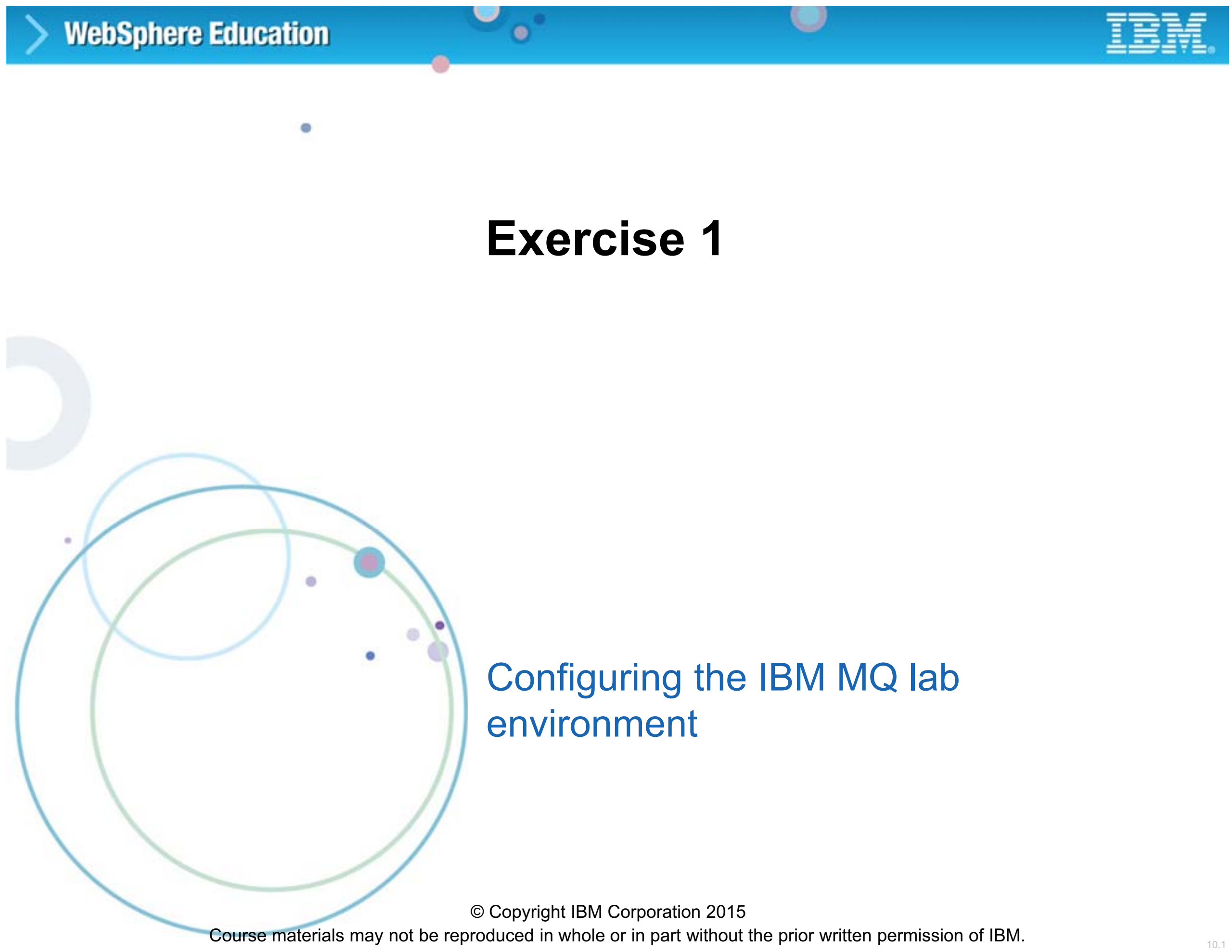


Figure 1-58. Exercise 1 WM3121.1

Notes:

WebSphere Education IBM

Queue manager and TSO ID pairs per student

Queue managers starting on Host 1	Ports	TSO ID	Certificate authority (CA)	Queue managers starting on Host 2	Ports	TSO ID	Certificate authority (CA)
MQ01 MQ21	3101 3101	TSM0001 TSM0021	ST01CA1 ST01CA2	MQ02 MQ22	3102 3102	TSM0002 TSM0022	ST02CA1 ST02CA2
MQ03 MQ23	3103 3103	TSM0003 TSM0023	ST03CA1 ST03CA2	MQ04 MQ24	3104 3104	TSM0004 TSM0024	ST04CA1 ST04CA2
Continue naming pattern for 05 through 08 and 25 through 28 through the omitted TSO IDs							
MQ09 MQ29	3109 3109	TSM0009 TSM0029	ST09CA1 ST09CA2	MQ10 MQ30	3110 3110	TSM0010 TSM0030	ST10CA1 ST10CA2
MQ11 MQ31	3111 3111	TSM0011 TSM0031	ST11CA1 ST11CA2	MQ12 MQ32	3112 3112	TSM0012 TSM0032	ST12CA1 ST12CA2

Figure 1-59. Queue manager and TSO ID pairs per student WM3121.1

Notes:

WebSphere Education

Exercise objectives (1 of 2)

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

Figure 1-60. Exercise objectives (1 of 2) WM3121.1

Notes:

WebSphere Education

Exercise objectives (2 of 2)

- Assemble the parameter module
- Start primary and secondary queue managers and channel initiators
- Create IBM MQ objects that are used in Exercise 2

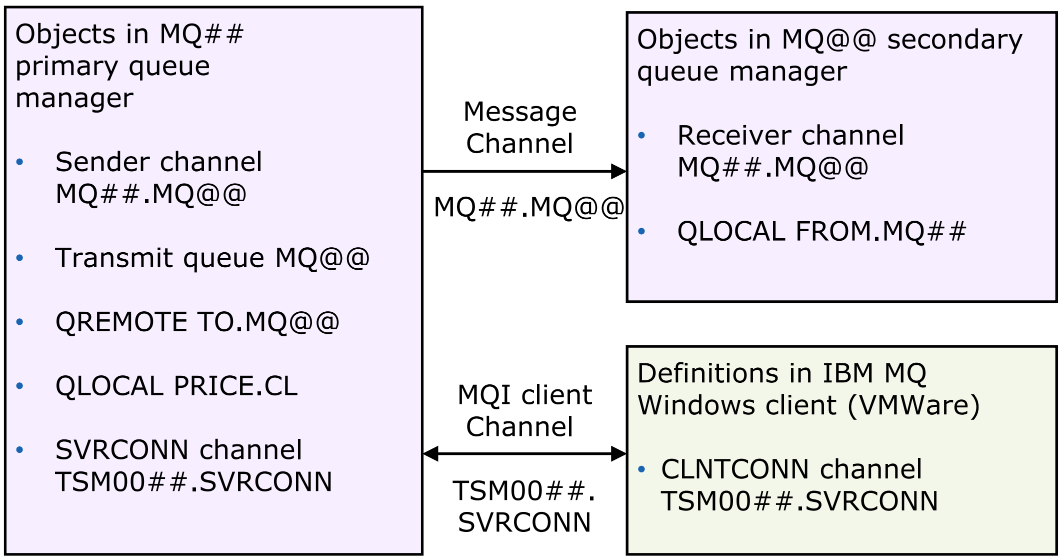


Figure 1-61. Exercise objectives (2 of 2) WM3121.1

Notes:

Unit 2. Channel security with SSL/TLS

What this unit is about

This unit introduces encryption topics and SSL/TLS, details how to implement SSL/TLS in a queue manager, and explains how to configure IBM MQ server and IBM MQ clients to use SSL/TLS.

What you should be able to do

After completing this unit, you should be able to:

- Explain the security considerations for IBM MQ channels
- Describe symmetric and asymmetric cryptography
- Explain the role of a certificate, a digital signature, and a certificate authority
- Distinguish between a self-signed certificate and a certificate-authority signed certificate
- Describe how SSL/TLS works to secure communications
- Distinguish between TLS and SSL cipher suites
- Describe how to implement SSL/TLS in a z/OS queue manager
- Describe how to create a key ring and associate certificates with it
- Explain how to implement SSL/TLS in a channel
- Describe how to request and obtain a certificate from a certificate authority
- Explain how to configure use of multiple certificates for the same queue manager
- Explain how to configure SSL/TLS for an IBM MQ client

How you will check your progress

- Checkpoint questions
- Lab exercises

WebSphere Education

Unit objectives

- Explain the security considerations for IBM MQ channels
- Describe symmetric and asymmetric cryptography
- Explain the role of a certificate, a digital signature, and a certificate authority
- Distinguish between a self-signed certificate and a certificate-authority signed certificate
- Describe how SSL/TLS works to secure communications
- Distinguish between TLS and SSL cipher suites
- Describe how to implement SSL/TLS in a z/OS queue manager
- Describe how to create a key ring and associate certificates with it
- Explain how to implement SSL/TLS in a channel
- Describe how to request and obtain a certificate from a certificate authority
- Explain how to configure use of multiple certificates for the same queue manager
- Explain how to configure SSL/TLS for an IBM MQ client

Figure 2-1. Unit objectives WM3121.1

Notes:

WebSphere Education

IBM MQ security and channels

- Identification
  - OS user IDs
  - Context
  - Channel authentication
  - Channel security exits
- Authentication
  - MQCONNX MQI call
  - Channel security exits
  - Channel authentication (CHLAUTH)
  - Secure Sockets Layer and Transport Layer Security (SSL/TLS)
  - Connection authentication
- Auditing
  - Events
- Access control
  - RACF switches
  - Put authority
  - Channel authentication
  - Firewalls
- Confidentiality
  - Advanced Message Security (AMS)
  - Secure Sockets Layer and Transport Layer Security (SSL/TLS)
  - Channel exits
- Data Integrity
  - Advanced Message Security (AMS)
  - Secure Sockets Layer and Transport Layer Security (SSL/TLS)

Figure 2-2. IBM MQ security and channels WM3121.1

Notes:

The first unit distinguished between application security, which includes “data at rest”, and link level security, as it applies to channels. IBM MQ provides different options for channel security across the different security areas.

In older versions of IBM MQ, it was necessary to use SSL or a channel security exit to provide identification and authentication. Later versions of IBM MQ introduced connection authentication and channel authentication to reduce or remove the need for security exits for identification and authentication.

Channel authentication, which is covered in Unit 3 of this course, is also used to supplement access control in addition to the RACF switches and firewalls. SSL/TLS can be used to ensure the confidentiality of messages without the need to code an encryption exit. Data Integrity can be implemented with a combination of SSL/TLS, and either message exit, or by using the Advanced Message Security offering.

WebSphere Education

Symmetric key encryption

- Establishes confidentiality in data
- Encryption and decryption use the same key
- Significant challenge in securely distributing key
- Common algorithms: DES, Triple DES, AES

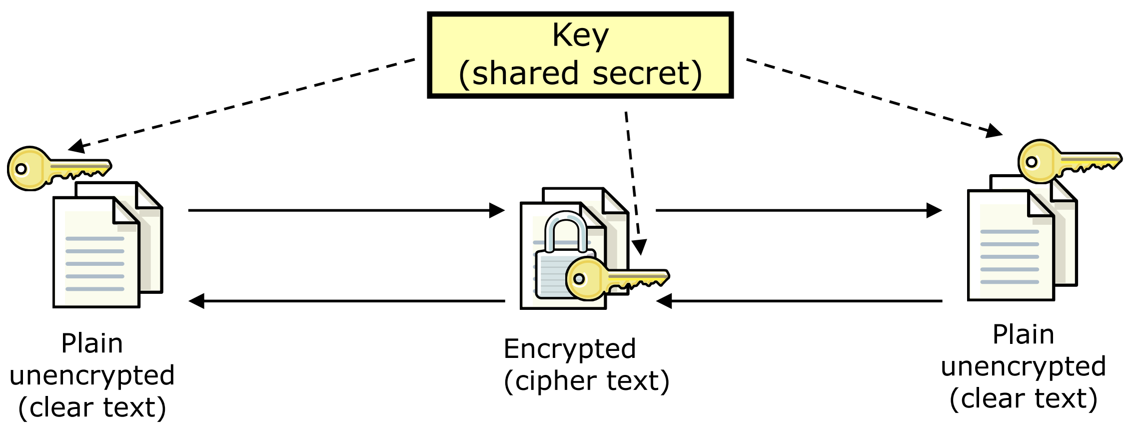


Figure 2-3. Symmetric key encryption WM3121.1

Notes:

Overhead of symmetric key encryption algorithms is relatively low. Key distribution is a significant problem that symmetric encryption faces. It is important that after a key is established between the two parties, it is kept private.

If two parties already exchanged a "shared secret", then use symmetric key encryption to secure communication.

Example: Banks mail passwords or PINs to customers – that works since there is an established relationship.

If two parties who do not have a "shared secret" want to secure their communications, there is a serious challenge.

Symmetric encryption works relatively fast.

WebSphere Education

Asymmetric encryption

- Key pair referred to as public and private key
- Public and related private keys are numerically associated
- Provide for confidentiality, integrity, and non-repudiation
- Data encrypted or signed with one of the keys can only be decrypted or verified with the associated key
- Computationally most expensive and slow
- Distribution
  - Public key is distributed to other peers
  - Private key is securely kept by the owner entity
- Common algorithms are RSA, DSA, and ECC

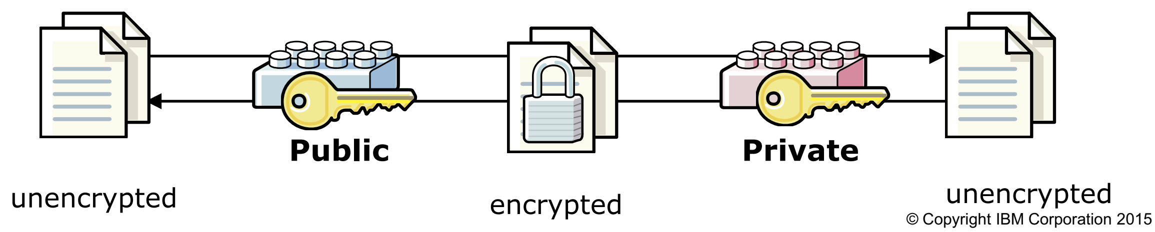


Figure 2-4. Asymmetric encryption WM3121.1

Notes:

Asymmetric algorithms use a pair of keys. One is used for encryption and the other one for decryption. The decryption key is kept private, so it is called a ``private key'' or ``secret key'', while the encryption key is distributed, hence it is called a ``public key''. Anyone who has the public key is able to send encrypted messages to the owner of the secret key. The secret key cannot be reconstructed from the public key.

Asymmetric algorithms seem to be ideally suited for real-world use; the secret key does not have to be shared, so the risk of it being discovered is much smaller. Each user needs to keep one secret key private and maintain a collection of public keys that can be shared as necessary.

However, asymmetric algorithms are much slower than symmetric ones. Therefore, in many applications, a combination of both symmetric and asymmetric encryption is used.

WebSphere Education

Hash or message digest

- A unique, fixed-length value that is generated from variable length data
  - Same input data always generates the same hash or digest value
  - Slightest change in data causes significant variation in digest value
  - Chance to find two different data values that result in same digest value is theoretically impossible
- Digest results are determined by the digest or hash algorithm – no keys are involved
- One-way digest value cannot be reversed into the original data
- Play a role in data integrity and origin authentication
- Some of the common algorithms are SHA1, SHA256, SHA384
- The message digest can be encrypted and becomes a digital signature

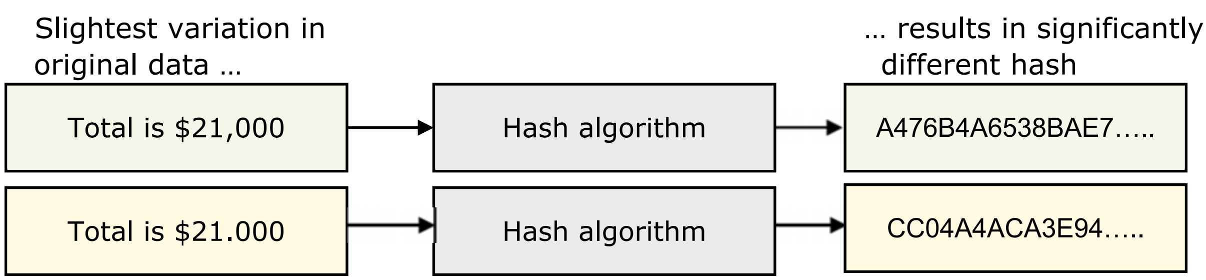


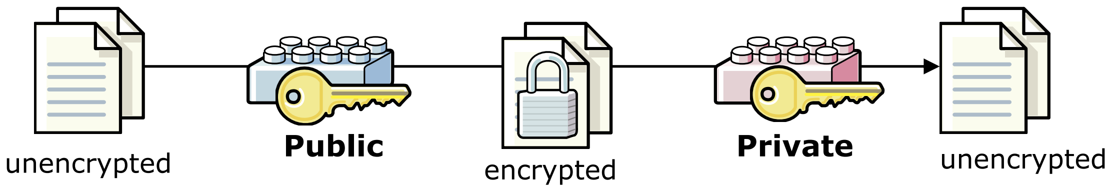
Figure 2-5. Hash or message digest WM3121.1

Notes:

WebSphere Education

Confidentiality: Encryption

- Public key cryptography
  - If you encrypt a message with the public key, you must use the private key to decrypt it.



  - If you encrypt a message with the private key, you must use the public key to decrypt it.

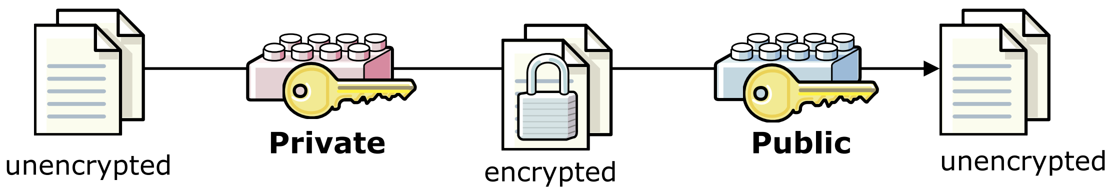


Figure 2-6. Confidentiality: Encryption WM3121.1

Notes:

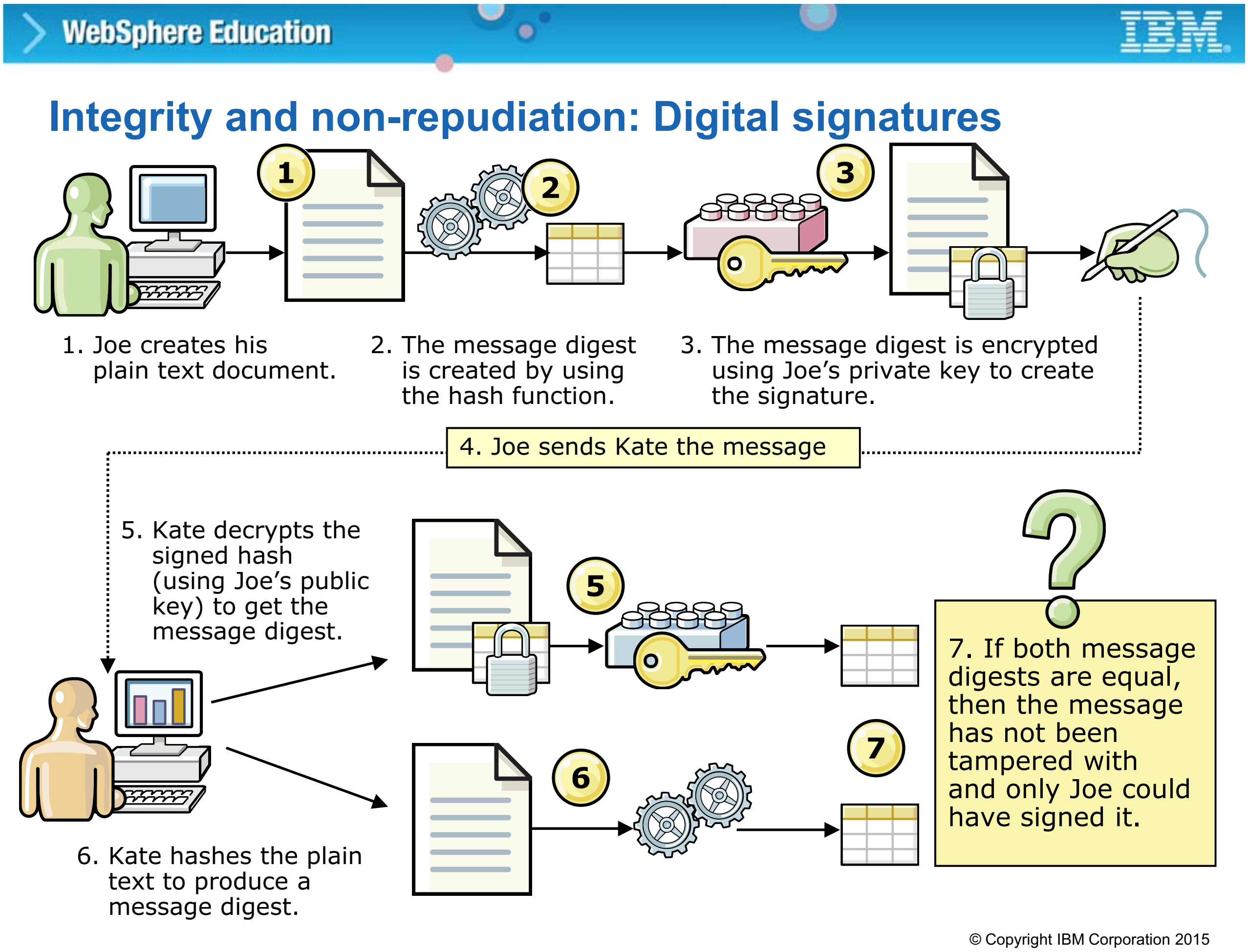


Figure 2-7. Integrity and non-repudiation: Digital signatures WM3121.1

Notes:

A digital signature does not encrypt the whole message; only the message digest is encrypted.

WebSphere Education

How are the public keys distributed?

- When a client needs a server's public key, the server itself supplies it in the form of a certificate.
  - Important question: How can you trust the public key really belongs to Kate, when Kate is the one that provided it?

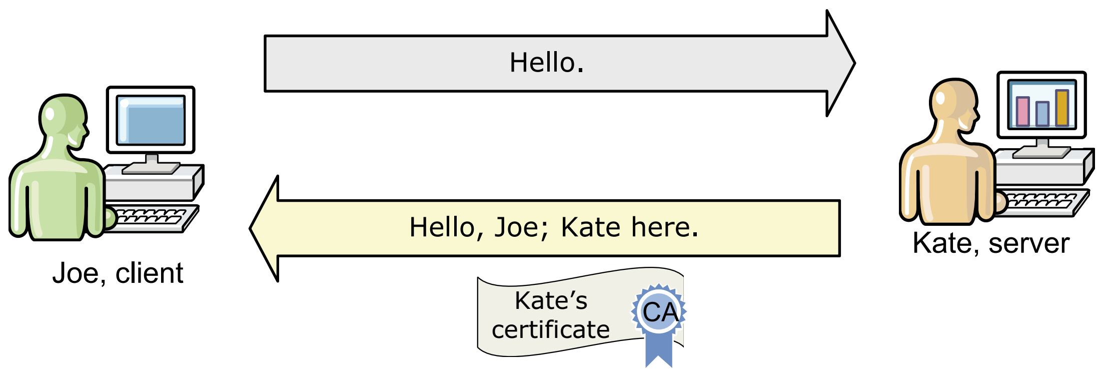


Figure 2-8. How are the public keys distributed? WM3121.1

Notes:

Keys are distributed in the form of certificates. You now look at the need for certificates and certificate authorities.

WebSphere Education

Certificates

- A certificate is a document that identifies you, and that a third party vouched for both you and the document itself, such as:
  - An employee badge (vouched for by your employer)
  - A drivers license (vouched for by your state)
  - A passport (vouched for by your country)
- As applied to cryptography: Digital certificate
  - Includes information about you or your server
  - Includes public key
  - Is signed by a certificate authority

Figure 2-9. Certificates WM3121.1

Notes:

WebSphere Education

What is a certificate authority or CA?

- A trusted third party that signs public keys and creates certificates
  - A CA validates Kate's identity before vouching for her (signing her certificate)
  - A special type of signer (a trusted signer)
- Well-known CAs have their signers built into servers

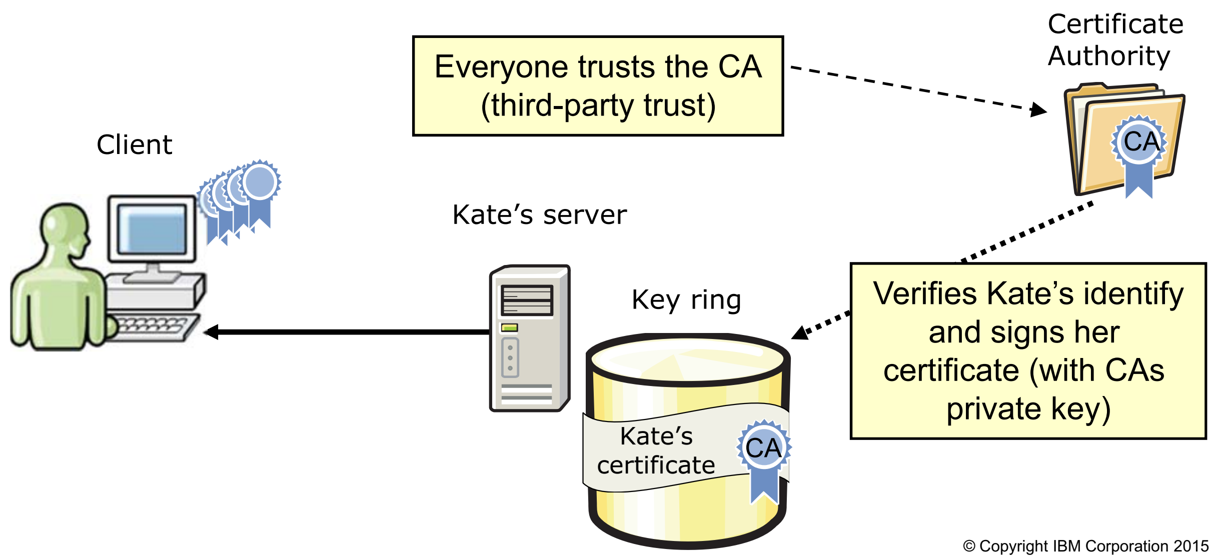


Figure 2-10. What is a certificate authority or CA? WM3121.1

Notes:

You can display some of these "well-known CA certificates" in the lab exercise.

WebSphere Education

Digital certificates

- Digital document that is generated by a trusted third party that binds and end-entity to a public key
- End entity is a person or device that needs an electronic identity
  - Identity that is encoded in certificate as the Subjects Distinguished Name (SDN)
  - Can prove possession of the corresponding private key
- Public key is the shared half of the public/private key pair. Usually digitally signed by the CA
- Established standardized format and contents packaging is known as the X.509 digital certificate
- Standards that are evolved with extensions added to the X.509 format to include other functions such as
  - Bind basic identity information to the public key
  - How public key can be used
  - Revocation
- Kept on platform or application-dependent certificate stores

Figure 2-11. Digital certificates WM3121.1

Notes:

WebSphere Education IBM

Different types of certificates by method created and usage

- Self-signed
  - Self-issued
  - Issuer and subject names are identical
  - Signed by itself using associated private key
  - No trusted party
- CA signed certificate
  - Signed by a trusted CA by using its private key
  - User sends a certificate request to CA
  - Validity of information in the certificate is certified by the CA
  - The signed certificate can be an end-entity certificate or a CA certificate
- Types of certificate by usage can be:
  - Personal certificate
  - Certificate authority, or CA signer certificate
  - Code signing certificate

Figure 2-12. Different types of certificates by method created and usage WM3121.1

Notes:

WebSphere Education IBM

What does the certificate contain? X.509 formats

- Single certificate
- Public-Key Cryptographic Standards, or PKCS packages
  - PKCS #7 package contains 1 or more certificates
  - PKCS #12 package contains one or more certificates plus the private key that is associated with the end-entity
- Encoding
  - DER file types are usually binary-encoded
  - PEM is Base64 format that is used to convert binary data to displayable data to enable cut and paste
  - PEM format contents show the `---- BEGIN CERTIFICATE ----` and `---- END CERTIFICATE ----` text markers

Figure 2-13. What does the certificate contain? X.509 formats WM3121.1

Notes:

WebSphere Education

Certificate chains

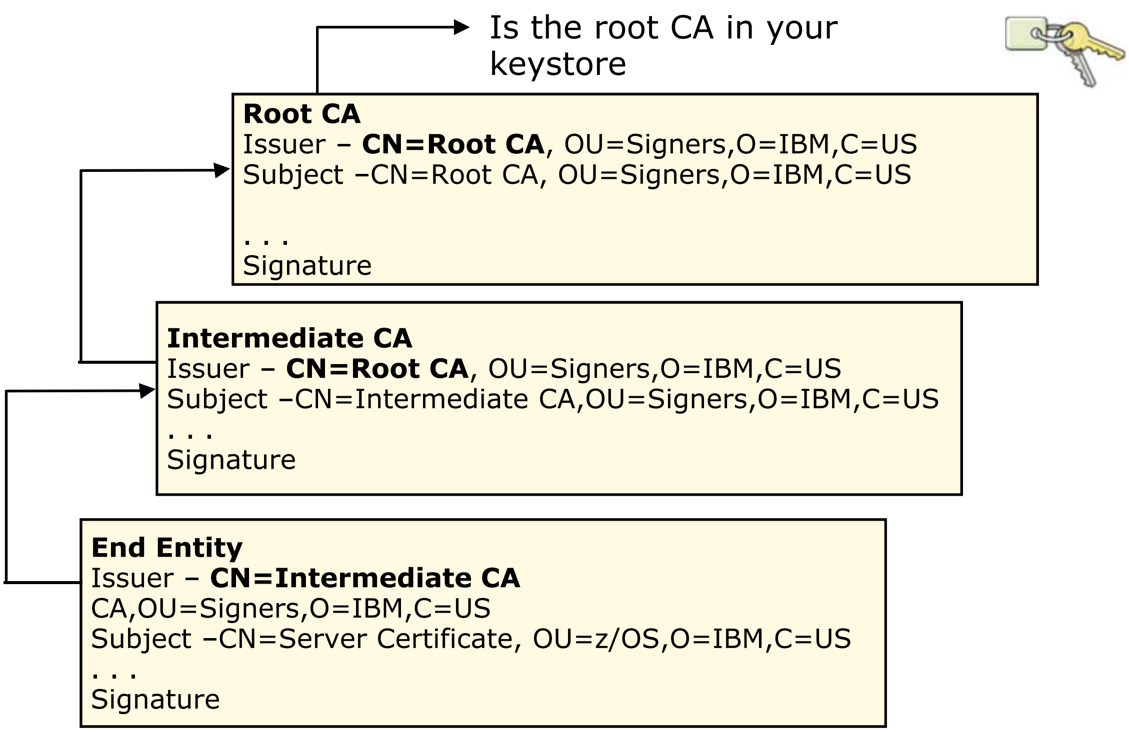


Figure 2-14. Certificate chains WM3121.1

Notes:

You must have all certificates higher than your certificate in your key store. In the case that is shown in this figure, you need the Root CA and Intermediate CA.

WebSphere Education

What information is in a certificate

```
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.SP=HAMPSHIRE.C<

      >=UK<
 Subject's Name:
      >CN=STUDENT 01 CA1.T=IBM
MQ.OU=TRAINING.O=IBM.L=HURSLEY.SP=HAMPSHIRE.C<

      >=UK<
 Signing Algorithm: sha1RSA
 Key Usage: CERTSIGN
 Key Type: RSA
 Key Size: 1024
  Private Key: YES
 Ring Associations:
 *** No rings associated ***
```


Figure 2-15. What information is in a certificate WM3121.1

Notes:

The fields that are shown are part of the X.509 standard and extensions. You see the certificate label and the certificate valid date ranges.

If you compare the issuer name and subject name, you can tell that it is a self-signed certificate as both values are the same.

WebSphere Education

Secure Sockets Layer and Transport Layer Security

- Secure Sockets Layer (SSL) and Transport Layer Security (TLS) are protocols that use digital certificates to provide confidentiality, data integrity, identification, and authentication.
- SSL and TLS are similar protocols, but different enough to prevent interoperation between the two.
- SSL and TLS use both symmetric and asymmetric cryptography.
- Application that initiates connection is considered the client. The responder is the server.
- A connection is initiated with the SSL or TLS handshake.

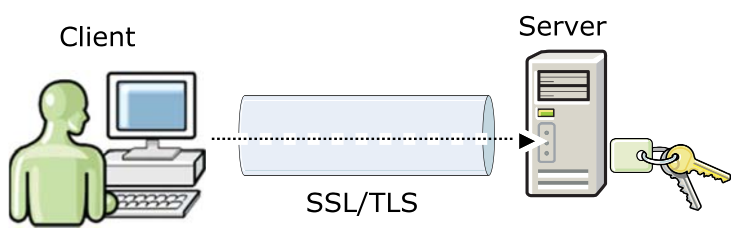


Figure 2-16. Secure Sockets Layer and Transport Layer Security WM3121.1

Notes:

While most instructions on implementing SSL/TLS for IBM MQ channels only mention SSL, the process to implement SSL and TLS in the queue manager and channels is similar.

SSL and TLS are similar protocols, but their differences are significant so that it is not possible to mix the protocols in the same connection. For example, the ability to configure multiple certificates in the same queue manager is enabled by TLS capabilities only.

WebSphere Education IBM

Federal Information Processing Standards and National Security Agency Suite B cryptography

- Federal Information Processing Standards (FIPS) and National Security Agency (NSA) Suite B cryptography are government suggested standards
- Latest standard requirements must be checked regularly
- FIPS requires a combination of algorithms and key sizes
- Suite B requires use of specific encryption algorithm (AES), key exchange algorithm (ECDH), digital signature algorithm (ECDSA), and hashing algorithms (SHA-256 or SHA-384)
- Suite B has a compliant profile for TLS 1.2, and a transitional profile that can use TLS 1.0 or TLS 1.1
- IBM MQ FIPS and Suite B compliance for z/OS and distributed platforms (for most platforms) is documented in the IBM Knowledge Center

Figure 2-17. Federal Information Processing Standards and National Security Agency Suite B cryptography WM3121.1

Notes:

Other terminology that you might hear with TLS is FIPS and NSA Suite B standards. These standards require certain combinations of algorithms to be compliant.

WebSphere Education

Encryption and message digest (hash) recap

- Some common encryption algorithms
  - Examples of symmetric encryption algorithms are DES, Triple DES, and AES
  - Examples of asymmetric encryption algorithms are RSA, DSA, and ECC
- Examples of hash or message digest algorithms are
  - SHA1, SHA256, and SHA384
  - MD5 is also a hash algorithm but due to faults encountered use of SHA algorithms is favored over MD5

Figure 2-18. Encryption and message digest (hash) recap WM3121.1

Notes:

Before you proceed to the CipherSpecs slide, you review the algorithms that were described earlier.

WebSphere Education

CipherSpecs, SSL, and TLS

- A cipher specification, or **`CipherSpec`** is the combination of the encryption algorithm and the message authentication code, or MAC algorithm that is used to generate the message digest
- Use of TLS or SSL is determined by the **`CipherSpec`** selected
- Organizations are moving to TLS due to SSL V3.0 vulnerabilities

Plat-form	CipherSpec name	Protocol	Data Integrity	Encryption algorithm	Encr. bits
All	TRIPLE_DES_SHA_US	SSL 3.0	SHA-1	3DES	168
All	TLS_RSA_WITH_AES_128_CBC_SHA	TLS 1.0	SHA-1	AES	128
All	TLS_RSA_WITH_AES_256_CBC_SHA	TLS 1.0	SHA-1	AES	256
All	TLS_RSA_WITH_DES_CBC_SHA	TLS 1.0	SHA-1	DES	56

Table is a partial example from the IBM MQ V8 IBM Knowledge Center Security > Data integrity of messages > Specifying CipherSpecs

Figure 2-19. CipherSpecs, SSL, and TLS WM3121.1

Notes:

CipherSpec selection should be resolved before starting to implement SSL/TLS.

Check the IBM MQ bulletins for any warnings on flaws that are encountered for a specific CipherSpec.

WebSphere Education

SSL handshake

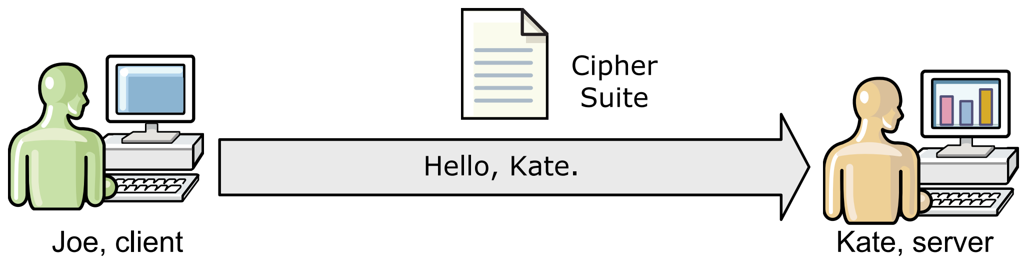
- The handshake is the way SSL establishes a secure communication link between a client and a server.
- During the handshake process, SSL:
  - Shares the server certificate with the client
  - Negotiates the level of SSL to use
  - Decides on a cipher suite that both parties can use
  - Authenticates the server and (optionally) the client
  - Builds a session key that is used for this session only
- The handshake is initiated by the client, sending a “hello” message to a server.

Figure 2-20. SSL handshake WM3121.1

Notes:

WebSphere Education

SSL handshake (1 of 7): Client hello



- Joe sends Kate a hello message.
- Within this initial contact message is a list of cipher suites that Joe (the client) can use.
- Joe is considered the client since he initiated communication.

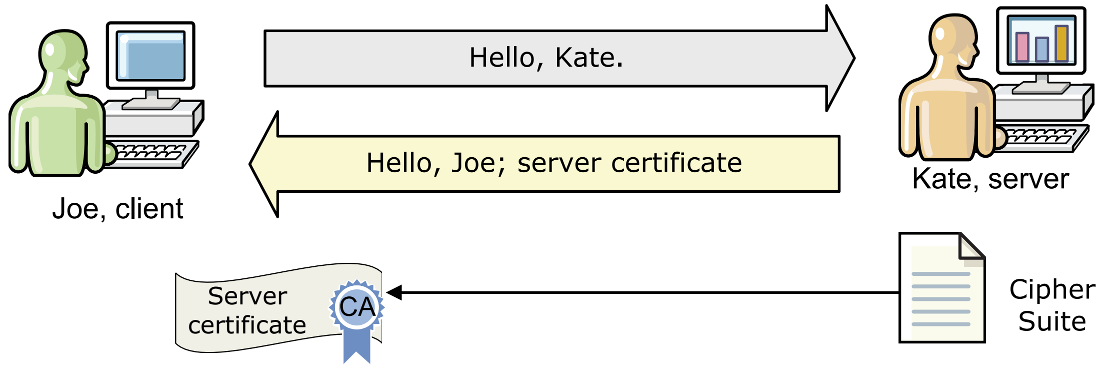
Figure 2-21. SSL handshake (1 of 7): Client hello WM3121.1

Notes:

At the beginning of an SSL session, an SSL handshake takes place. This handshake produces the cryptographic parameters of the session.

WebSphere Education

SSL handshake (2 of 7): Server hello



- Kate (the server) responds with a “Hello”.
- This response contains the cipher suite that Kate selected from the list Joe sent.
- Kate also sends Joe a signed digital certificate that contains her public key.

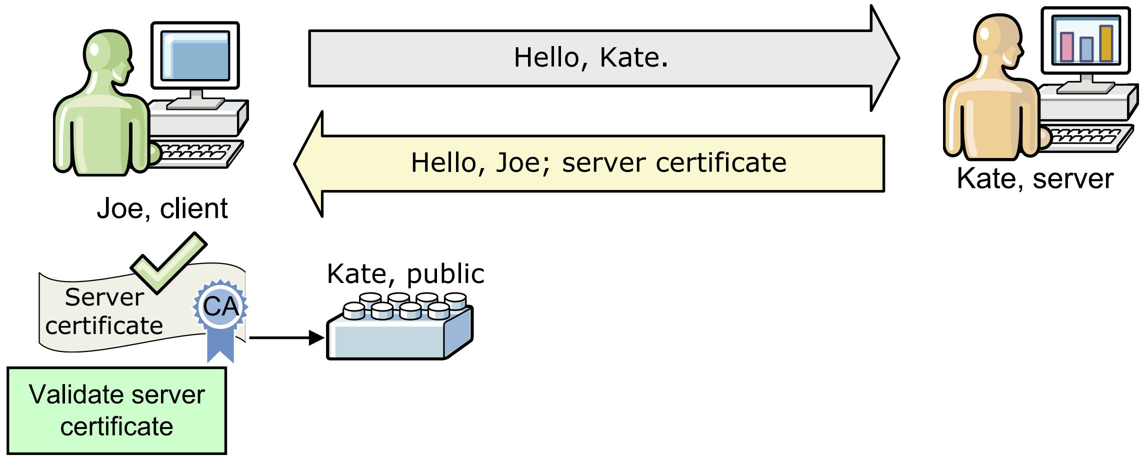
Figure 2-22. SSL handshake (2 of 7): Server hello WM3121.1

Notes:

Following the server hello message, the server sends its certificate. This message contains the server’s digital certificate and all other certificates up to the "root". The client must match the issuers of the certificates all the way up the certificate chain to the root certificate to find a match with an issuer that it trusts.

WebSphere Education

SSL handshake (3 of 7): Verify server certificate



- Check whether certificate expired.
- Validate server's certificate.
- Receive Kate's public key from certificate.
- If client authentication is being used, then Kate would have also requested a digital certificate from the client.

Figure 2-23. SSL handshake (3 of 7): Verify server certificate WM3121.1

Notes:

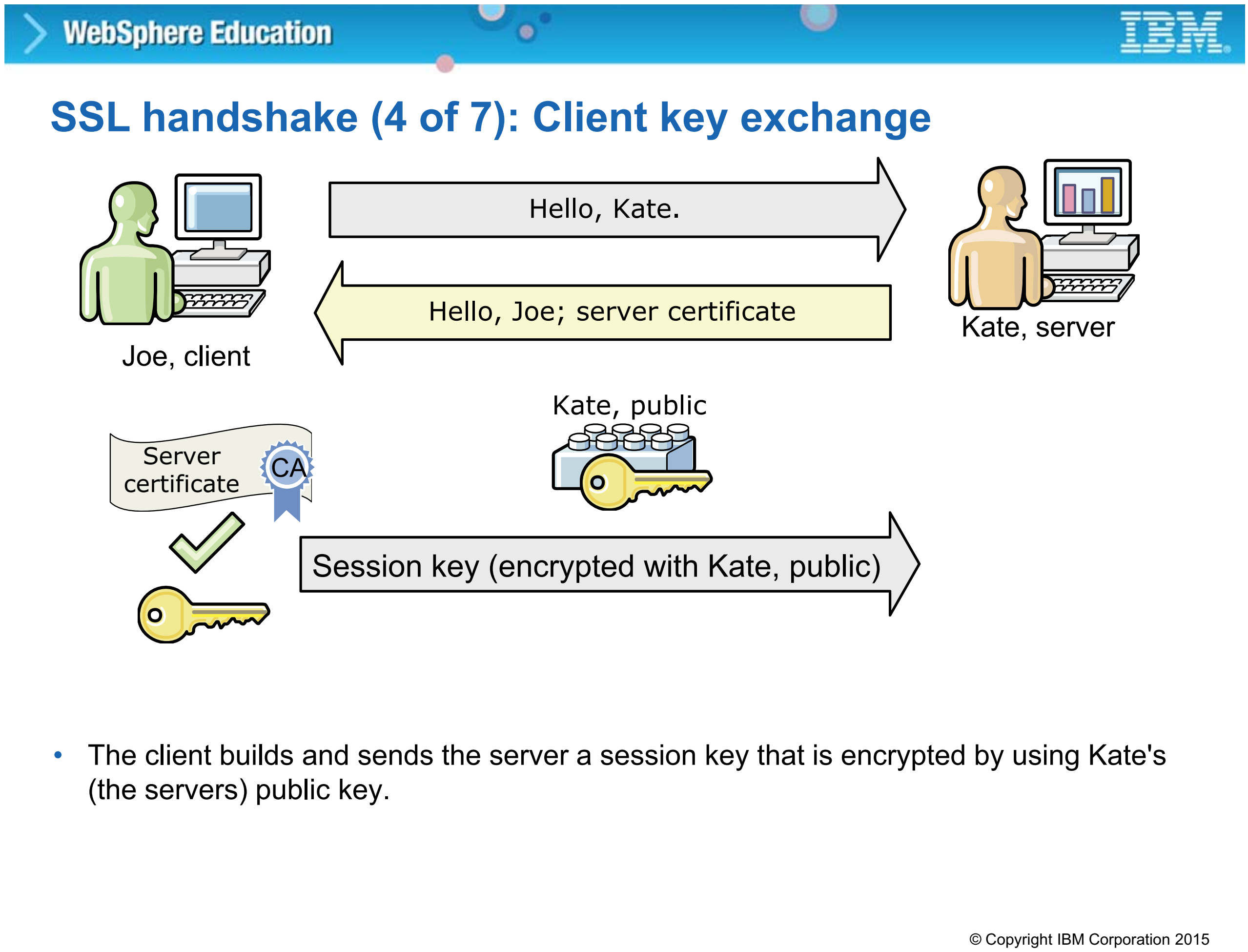


Figure 2-24. SSL handshake (4 of 7): Client key exchange WM3121.1

Notes:

The client sends a random byte string that is used by both client and server to compute a secret key that is used for encrypting message data. This random byte string is encrypted with the server's public key.

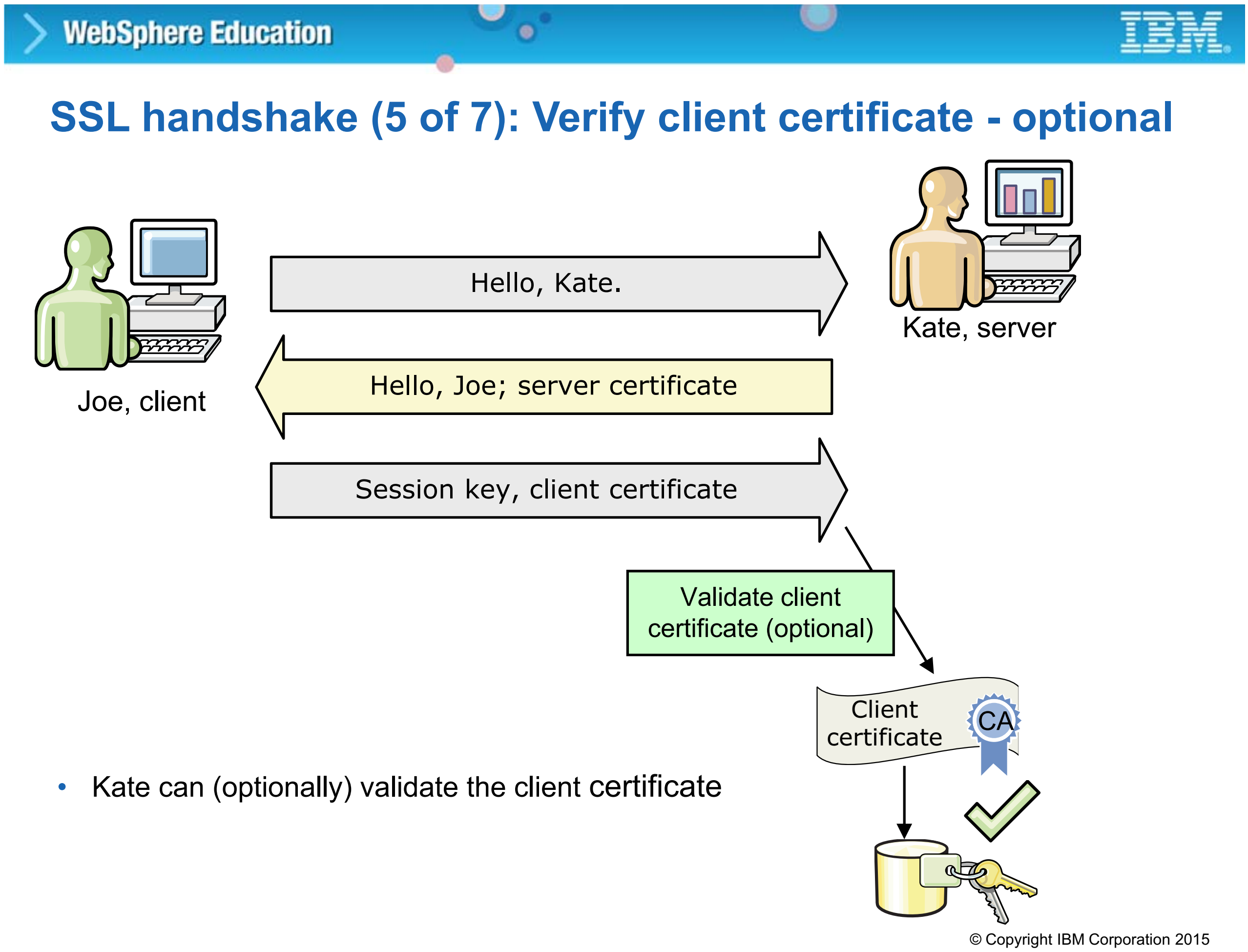


Figure 2-25. SSL handshake (5 of 7): Verify client certificate - optional WM3121.1

Notes:

If the server sent a "client certificate request", the client sends:

- A random byte string encrypted with the client's private key
- The client's digital certificate or a "no digital certificate alert"

Although this alert is only a warning, in some implementations the handshake fails if client authentication is mandatory and the warning is received.

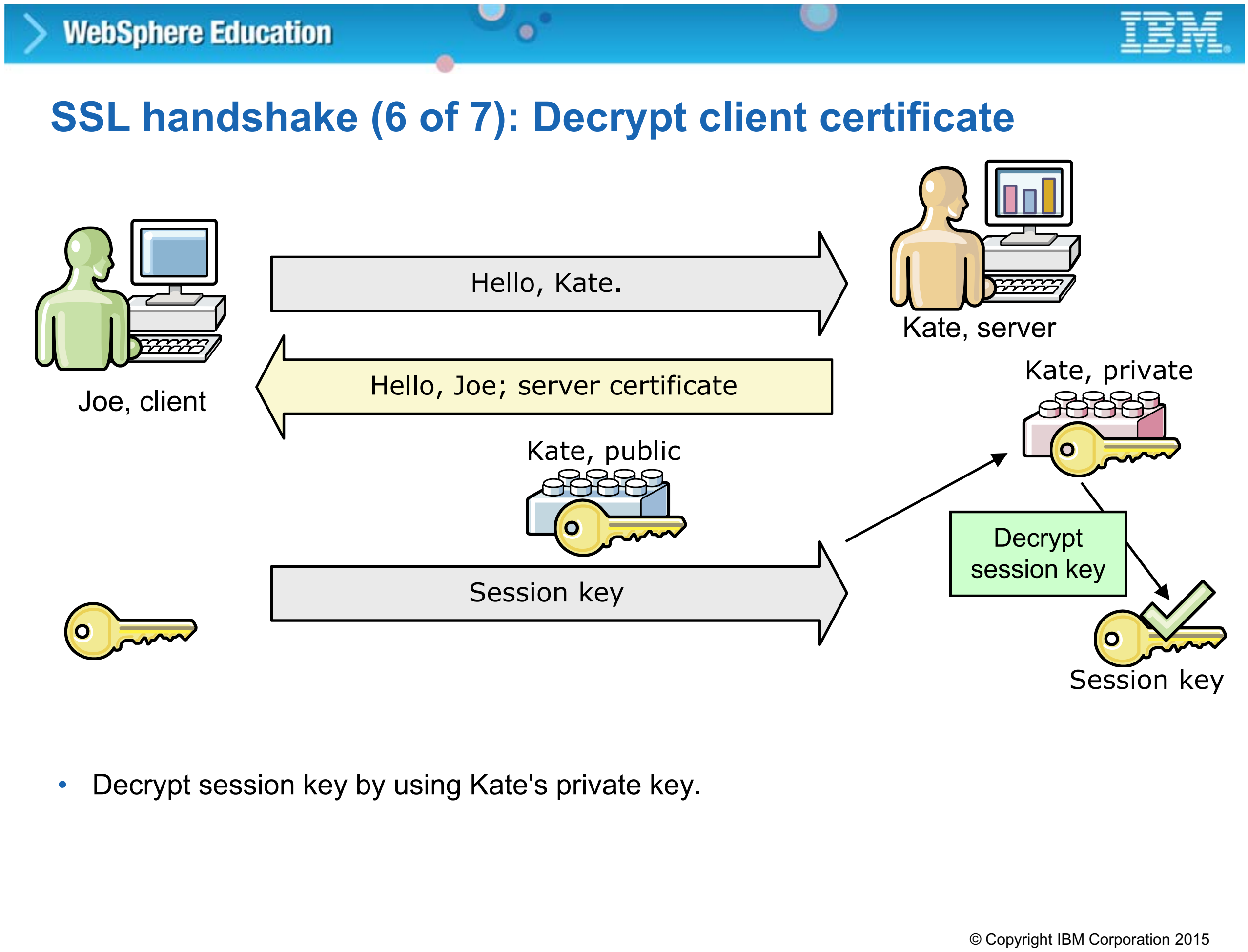


Figure 2-26. SSL handshake (6 of 7): Decrypt client certificate WM3121.1

Notes:

The server verifies the client's certificate.

SSL handshake (7 of 7): Complete

Using the session key (which only the client and server knows – a shared secret), a switch is made to using symmetric key encryption.

Figure 2-27. SSL handshake (7 of 7): Complete WM3121.1

Notes:

The client sends the server a “finished” message and the server sends the client a “finished” message.

By using the shared secret key and symmetric encryption, messages are encrypted during the SSL session.

WebSphere Education

IBM

Before you get started

- Use of SSL/TLS in the infrastructure requires adequate planning and governance
- Expired certificates can lead to a production outage
- Organizations must establish roles
  - Who owns certificates
  - Who monitors certificate expiration
  - Who monitors security updates
- Security standards must be documented and enforced
  - What **`CipherSpecs`** to implement
  - What connection partners need to update **`CipherSpecs`**
  - Who has permission to use RACDCERT
  - What security is implemented to protect key databases in distributed platforms

Figure 2-28. Before you get started WM3121.1

Notes:

Before rushing to implement SSL/TLS, an IBM MQ administrator needs to obtain pertinent information on the security requirements and the organizational roles for SSL/TLS implementation. If these steps are not handled, renewing certificates and reaching contacts who own peer connections becomes a significant problem that might result in an outage.

WebSphere Education

IBM MQ SSL/TLS z/OS checks

- z/OS prerequisites
  - SSL support by using z/OS System SSL with correct FMIDs installed to contain most current **`CipherSpecs`**
  - z/OS System Authorization Facility (SAF)
  - An external security manager (ESM), such as Resource Access Control Facility (RACF), ACF2, or Top Secret
  - *ESM references in this course use RACF*
- Administrator needs appropriate authority to enter ESM commands
- Channel initiator procedure must have access to the SSL runtime library SIEALNKE
- Integrated Cryptographic Server Facility (ICSF) should be available
- Channel initiator user ID must have READ access to
  - IRR.DIGTCERT.LISTRING profile
  - Appropriate CSF* profiles in CFSERV class

Figure 2-29. IBM MQ SSL/TLS z/OS checks WM3121.1

Notes:

This slide contains some of the areas you need to check or confirm before proceeding to implement SSL/TLS. These areas depend on your IBM MQ and z/OS levels.

Some of the IBM MQ CipherSpecs might require support from specific z/OS system SSL FMIDs. These FMIDs can be checked in the z/OS IBM Knowledge Center corresponding to your system. For example, for z/OS 1.13, see: ***z/OS 1.13.0 > z/OS Cryptographic Services > z/OS Cryptographic Services System SSL Programming > Cipher Suite Definitions***.

WebSphere Education

SSL/TLS configuration

- RACDCERT utility
  - Request CA or create self-signed certificates with RACDCERT utility
  - Add or import certificates to the database with RACDCERT utility
  - Create key ring with RACDCERT utility
  - Associate certificates with the key ring
- IBM MQ configuration
  - Configure queue manager to use SSL/TLS
  - Confirm that new or existing channels to be configured with SSL/TLS starts successfully before applying SSL/TLS attribute changes
  - Alter the channel to apply SSL/TLS required changes

Figure 2-30. SSL/TLS configuration WM3121.1

Notes:

Implementing SSL/TLS in IBM MQ requires separate steps. Depending on your organization, these steps might be conducted by the same or by different resources. Such possible role divisions might be:

- Application coordinator: A large application, might have a role that interfaces with peer organizations to agree on CipherSpecs and other details such as certificate expiration procedures.
- Owner of certificate role: Request and receive certificates from certificate authority.
- z/OS or IBM MQ administrator: Use RACDCERT to create the SSL/TLS resources for the queue manager.
- IBM MQ administrator: Configure the queue manager to use SS/TLS. After confirming that the channels to use SSL/TLS run before implementing SSL/TLS, configure SSL/TLS in each designated channel. This work might need to be coordinated with the owner of the peer channel.

WebSphere Education IBM

Certificate administration: RACDCERT (1 of 2)

- Certificate request and generation
  - RACDCERT GENREQ generates certificate request to be sent to CA
  - RACDCERT GENCERT generates certificate and key pair
- RACDCERT ADD is used to install a certificate and private/public key pair
- Administration
  - RACDCERT LIST: Display certificate information
  - RACDCERT DELETE: Delete certificate and key pair from database
  - RACDCERT ALTER: Change certificate information

Figure 2-31. Certificate administration: RACDCERT (1 of 2) WM3121.1

Notes:

If RACF is your ESM, RACDCERT is the command that is used to manage your certificates and key rings in the z/OS environment.

There are many uses of this command. For example, RACDCERT GENCERT generates a certificate and certificate key pair. You use this command to generate a certificate to use in RACDCERT GENCERT when you need to request a certificate from a CA. When you receive the certificate or need to add root or intermediate certificates from a CA, you use the RACDCERT ADD.

A valuable resource for this work is the *z/OS Security Server RACF Command Language Reference* for the z/OS version that your queue manager uses. This documentation includes:

- Syntax help
- Valid ways to run the command, such as in a TSO command.
- RACF authorizations that your TSO ID requires to run each command
- Detail of the processing that is done by the command
- If applicable, how to activate the command that was run

Certificate administration: RACDCERT (2 of 2)

- Ring administration
  - RACDCERT ADDRING creates a key ring
  - RACDCERT CONNECT associates a certificate with a key ring
  - RACDCERT REMOVE removes association of a certificate from a key ring
  - RACDCERT LISTRING displays information from a key ring
  - RACDCERT DELRING deletes a key ring

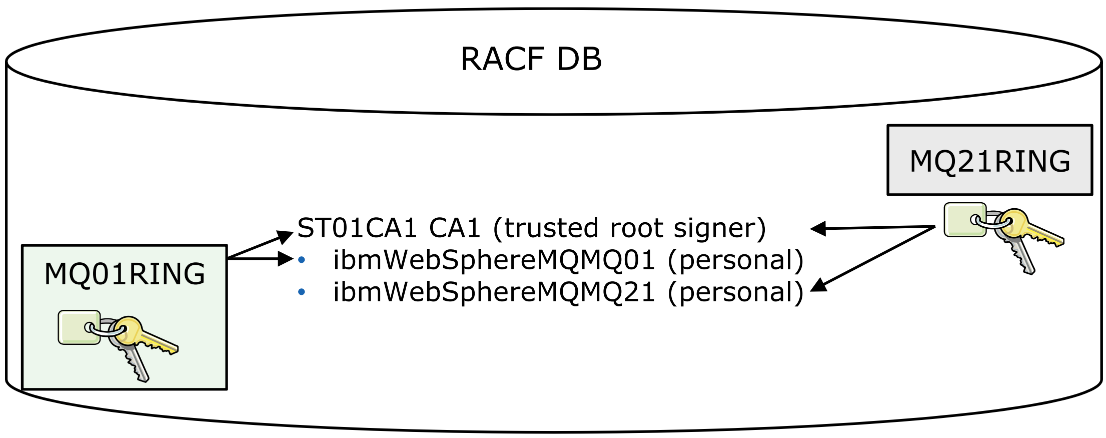


Figure 2-32. Certificate administration: RACDCERT (2 of 2) WM3121.1

Notes:

There are RACDCERT commands to create and administer the key ring for your z/OS queue manager. The certificates can be added independently from the key rings. You “connect”, or associate the certificates that the queue manager requires to each key ring.

In the slide, it is assumed the RACF database for both queue managers is in the same z/OS image. It is likely that when you work with peer connections the RACF database for each queue manager is separate, not as depicted in the slide.

WebSphere Education

SSL/TLS Implementation: Queue manager

- Create a key ring by using the RACDCERT administrative tool
- Configure the queue manager to use SSL/TLS – ALTER QMGR
  - CERTLABL: ibmWebSphereMQxxxx or blank.
  - CERTQSGL: Same as CERTLABL but takes precedence over CERTLABL if queue manager is part of a queue-sharing group.
  - SSLCRYP: Used if hardware encryption needs to be configured.
  - SSLEV: If SSL queue manager events need to be enabled. `ENABLED` or `DISABLED`. `DISABLED` is initial value.
  - SSLFIPS: YES or NO. If yes, forces use of FIPS certified **`CipherSpecs`** for any SSL channels.
  - SSSLKEYC: Number of bytes to send and receive for secret key renegotiation. Optional setting that is best left at initial value.
  - SSLKEYR: Name of SSL repository; in z/OS, key ring.
  - SSLTASKS: Number of server subtasks available for SSL. Must be at least 2.
  - SUITEB: NONE. Used only if Suite B compliant cryptography is used. NONE.
- Optional: Configure certificate revocation lists

Figure 2-33. SSL/TLS Implementation: Queue manager WM3121.1

Notes:

If this is the first time SSL/TLS is used by the queue manager, you must configure the queue manager first. Create a key ring by using the RACDCERT command, then change the required queue manager parameters.

The two required changes are increasing the number of SSL tasks, and identifying the key ring to your queue manager. Updating the SSLTASKS attribute to increase the number of SSL tasks requires a restart of the channel initiator to take effect. Check IBM Knowledge Center for the refresh or restart requirements for the other attributes. You look at the use of CERTLABL later in this unit.

If your security requirements include checking of certificate revocation lists (CRLs), you must configure an authorization object, a name list object, specify the name list to the queue manager, and configure LDAP. For example:

```
DEFINE AUTHINFO(LDAP1) AUTHTYPE(CRLLDAP)
DEFINE NAMELIST(LDAPNL) NAMES(LDAP1)
ALTER QMGR SSLCRLNL(LDAPNL)
```

The IBM Knowledge Center provides details on the objects and LDAP configuration that is required for CRLs.

WebSphere Education

SSL/TLS implementation: Channels

- CERTLABL
  - What certificate should be used for this channel
  - V8 and later queue managers
  - *TLS **`CipherSpecs`** only
- SSLCIPH
  - **`CipherSpec`** to use
  - Must match on corresponding connecting channel
- SSLCAUTH
  - Specifies whether the channel needs to authenticate the clients certificate
- SSLPEER
  - Optional
  - Used to compare the distinguished name (DN) of the certificate
  - If the DN of the incoming client connection does not match SSLPEER, connection fails

Figure 2-34. SSL/TLS implementation: Channels WM3121.1

Notes:

After the queue manager is configured to use SSL/TLS, the channels that require SSL can be created or changed to use SSL/TLS. Unless extra options are required, SSLCIPH is the only attribute that requires configuration and it contains the selected CipherSpec. The values correspond to the entries in the CipherSpec table in the IBM Knowledge Center.

- CERTLABL is new with IBM MQ V8. You learn more about this attribute in a later slide.
- In the SSL handshake, there was an optional step for the server to request the client certificate. When SSLCAUTH is set to REQUIRED, the server requests the client certificate during the SSL handshake.
- SSLPEER is used to check the distinguished name in the certificate of the peer queue manager or client. An alternative to the use of SSLPEER in the channel is to use a channel authentication rule.

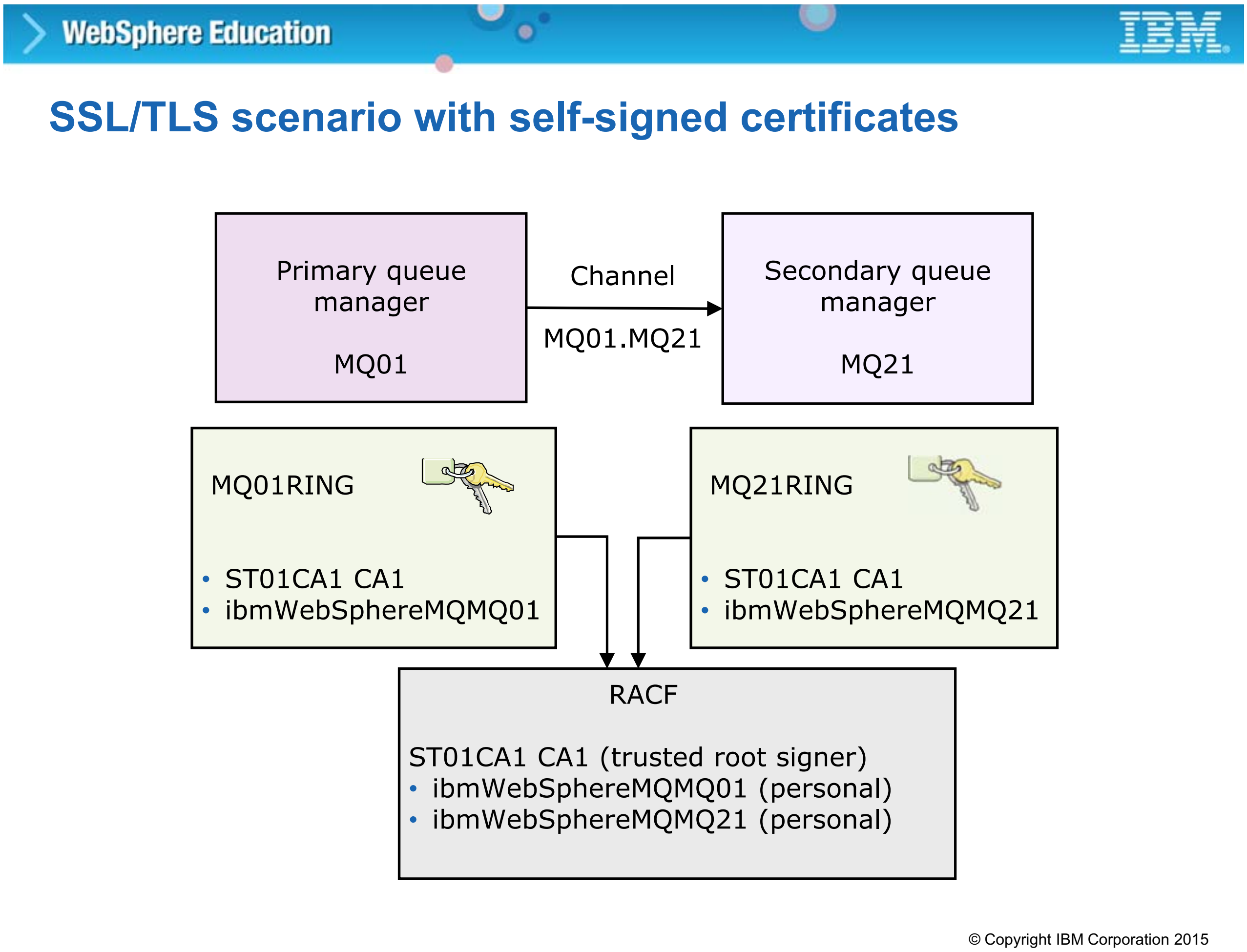


Figure 2-35. SSL/TLS scenario with self-signed certificates WM3121.1

Notes:

A basic SSL z/OS configuration is shown. In this configuration, both queue managers are in the same z/OS server and the same RACF database is used. In a real world environment, it is more likely that the certificates are in different RACF databases, one for each z/OS server.

If the RACF database was separate, the MQ01 server RACF would have the ST01CA1 root signer and the ibmWebSphereMQMQ01 certificate. The MQ21 server RACF would also have the ST01CA1 root signer then the ibmWebSphereMQMQ21 certificate.

The environment that is shown resembles the lab environment for this course, where all certificates are in the same RACF database. You now step through the work that is done to complete this configuration.

WebSphere Education IBM

Generate CA certificate

```
RACDCERT CERTAUTH GENCERT -
SUBJECTSDN(CN('STUDENT 01 CA1') -
T('IBM MQ') -
OU('TRAINING') -
O('IBM') -
L('HURSLEY') -
SP('HAMPSHIRE') -
C('UK')) -
NOTBEFORE(DATE(2015/05/01) TIME(00:00:00)) -
NOTAFTER(DATE(2016/05/01) TIME(23:59:59)) -
WITHLABEL('ST01CA1') -
KEYUSAGE(CERTSIGN)
RACDCERT CERTAUTH LIST
```


Figure 2-36. Generate CA certificate WM3121.1

Notes:

You use JCL to run the RACF commands. The first task is to create the self-issued CA signer, or root certificate for a self-signed “Certificate authority 1” or CA1.

Note the certificate’s label is ST01CA1 and its usage is CERTSIGN.

If you do not specify dates, the certificate is created to be valid starting today through midnight a year from today.

A second RACDCERT command is run to list the certificates of type CERTAUTH. The syntax that is used lists all certificates of type CERTAUTH in the RACF database.

WebSphere Education

Generate personal or server certificate

```
RACDCERT ID(TSM0001) GENCERT -
SUBJECTSDN(CN('MQ01 PERSONAL') -
T('QMGR') -
OU('TRAINING') -
O('IBM') -
L('HURSLEY') -
SP('HAMPSHIRE') -
C('UK')) -
NOTBEFORE(DATE(2015/05/03) TIME(00:00:00)) -
NOTAFTER(DATE(2016/05/01) TIME(23:59:59)) -
WITHLABEL('ibmWebSphereMQMQ01') -
SIGNWITH(CERTAUTH LABEL('ST01CA1'))
RACDCERT LIST(LABEL('ibmWebSphereMQMQ01'))
ID(TSM0001)
```


Figure 2-37. Generate personal or server certificate WM3121.1

Notes:

You now use RACDCERT again to generate your self-signed personal certificate, signed with the “CA root” created earlier. You now have a “CA root” signer certificate. Carefully check the date range and SIGNWITH value. This certificate is called ibmWebSpereMQMQ01.

The list operation displays the specific certificate by name.

WebSphere Education IBM

Caveats when working with RACDCERT

- Always look at the messages around the READY prompt – "COND CODE = 0" is misleading
- Always specify date ranges within signer certificate date range

```
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
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
  Serial Number:
       >01<
  … … … … …
```


Figure 2-38. Caveats when working with RACDCERT WM3121.1

Notes:

It is imperative to double check the actual messages in your job output. This particular job ended with COND CODE 0, although it resulted in an error warning.

Always check the valid dates for all certificates higher in the certificate chain.

WebSphere Education IBM

Work with key ring

- Create the key ring, connect certificates, and list contents of the ring

```
RACDCERT ID(TSM0001) ADDRING(MQ01RING)


RACDCERT ID(TSM0001)
CONNECT(CERTAUTH(LABEL('ST01CA1')-
RING(MQ01RING))
RACDCERT ID(TSM0001) CONNECT(ID(TSM0001)
LABEL('ibmWebSphereMQMQ01') -
RING(MQ01RING))


RACDCERT LISTRING(*) ID(TSM0001)
```


Figure 2-39. Work with key ring WM3121.1

Notes:

You now create the RACF key ring and associate both the “CA root” and the queue manager personal certificate with this key ring.

The list command as shown displays all key rings that are associated with TSO ID TSM0001.

WebSphere Education

Configure the queue manager to use SSL/TLS

- Increase the number of SSL tasks and identify the key ring to the queue manager

```
ALTER QMGR SSLKEYR(MQ01RING) SSLTASKS(5)
```

- Restart the channel initiator

Figure 2-40. Configure the queue manager to use SSL/TLS WM3121.1

Notes:

You alter the SSLKEYR and SSLTASKS attributes of your queue manager.

The channel initiator must be restarted to activate the SSLTASKS change. The restart also activates the SSLKEYR change.

WebSphere Education

Configure the channel to use SSL/TLS

- Confirm that the corresponding channel is ready to use SSL/TLS
- Stop channel if needed
- Before updating the channel, make sure that it starts successfully *without* SSL/TLS
- Change the required attributes

```
ALTER CHL(MQ01.MQ21) CHLTYPE(RCVR) +
   SSLCIPH(TLS_RSA_WITH_AES_128_CBC_SHA256)
```

- Start the channel
- Confirm that the channel started successfully without any error messages

Figure 2-41. Configure the channel to use SSL/TLS WM3121.1

Notes:

The first statement “Confirm that the corresponding channel is ready to use SSL/TLS” might sound obvious if you are working with a business peer, however, there can be several questions, such as.

- Is the corresponding IBM MQ administrator ready to move forward with adding SSL/TLS to the channel?
- Is the CipherSpec agreed upon?
- Is the connecting queue manager at the correct IBM MQ level to use the agreed upon CipherSpec?

Ensuring that the channel starts successfully before applying the SSL/TLS change can save significant time, as you want to isolate any pre-existing issues before attempting to activate SSL/TLS.

WebSphere Education IBM

Single queue manager certificate

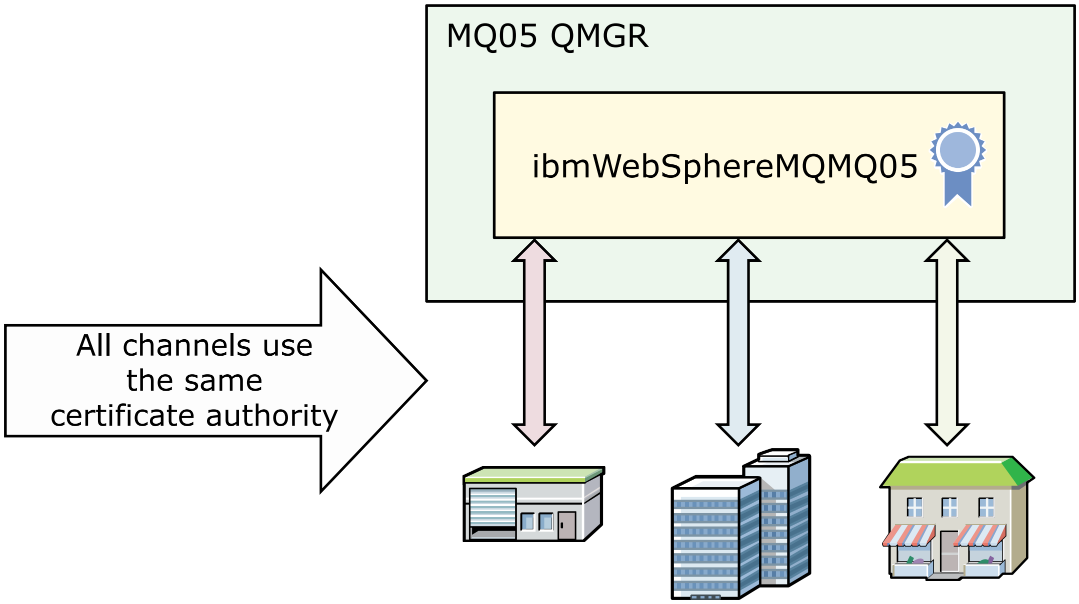


Figure 2-42. Single queue manager certificate WM3121.1

Notes:

In this figure, you see three possible connection partners for queue manager MQ05. As shown, all these partners use the same CA, so all queue managers have the same root and if necessary intermediary CA certificates.

The queue manager is configured with the standard IBM MQ certificate and label, ibmWebSphereMQMQ05. The CipherSpecs are agreed upon, and the channels are configured with SSL/TLS that uses the same CA.

WebSphere Education

Request a certificate from a certificate authority (1 of 2)

1. Create a self-signed personal certificate – RACDCERT GENCERT

```
RACDCERT ID(TSM0001) GENCERT -
SUBJECTSDN(CN('MQ05 QM') -
T('QMGR') -
OU('TRAINING') -
O('IBM') -
L('HURSLEY') -
SP('HAMPSHIRE') -
C('UK')) -
WITHLABEL('ibmWebSphereMQMQ05')
```

2. Use the self-signed certificate to create a certificate request and write this request to a file – RACDCERT GENREQ

```
RACDCERT ID('TSM0005') GENREQ(LABEL'ibmWebSphereMQMQ05')) -
  DSN('TSM0005.MQ05.CERT.REQ')
```


Figure 2-43. Request a certificate from a certificate authority (1 of 2) WM3121.1

Notes:

In the earlier slides, you saw how a self-signed certificate and root was created by using RACDCERT.

In a more realistic approach, a certificate request is sent to the certificate authority to obtain a certificate that the CA can vouch for. The certificate authority then returns the signed certificate to the requester, who in turns adds this certificate to the RACF database and eventually connects it to a queue manager key ring.

This slide shows the first two steps:

- Issue of a self-signed certificate
- Use of this certificate to generate the CA certificate request

WebSphere Education IBM

Request a certificate from a certificate authority (2 of 2)

3. Send the file to a certificate authority
4. Use the file that is returned by the CA to add the certificate back into the RACF database by using the same label name as in the GENCERT – RACDCERT ADD

```
RACDCERT ID('TSM0005') ADD('FROMCA.MQ05.CERTIFICATE.???') -
   WITHLABEL('ibmWebSphereMQMQ05')
```


Figure 2-44. Request a certificate from a certificate authority (2 of 2) WM3121.1

Notes:

After the CA receives and process the request, it returns the certificate to the requester, who then needs to add the CA certificate to the RACF database.

The certificate can now be associated with one or more key rings.

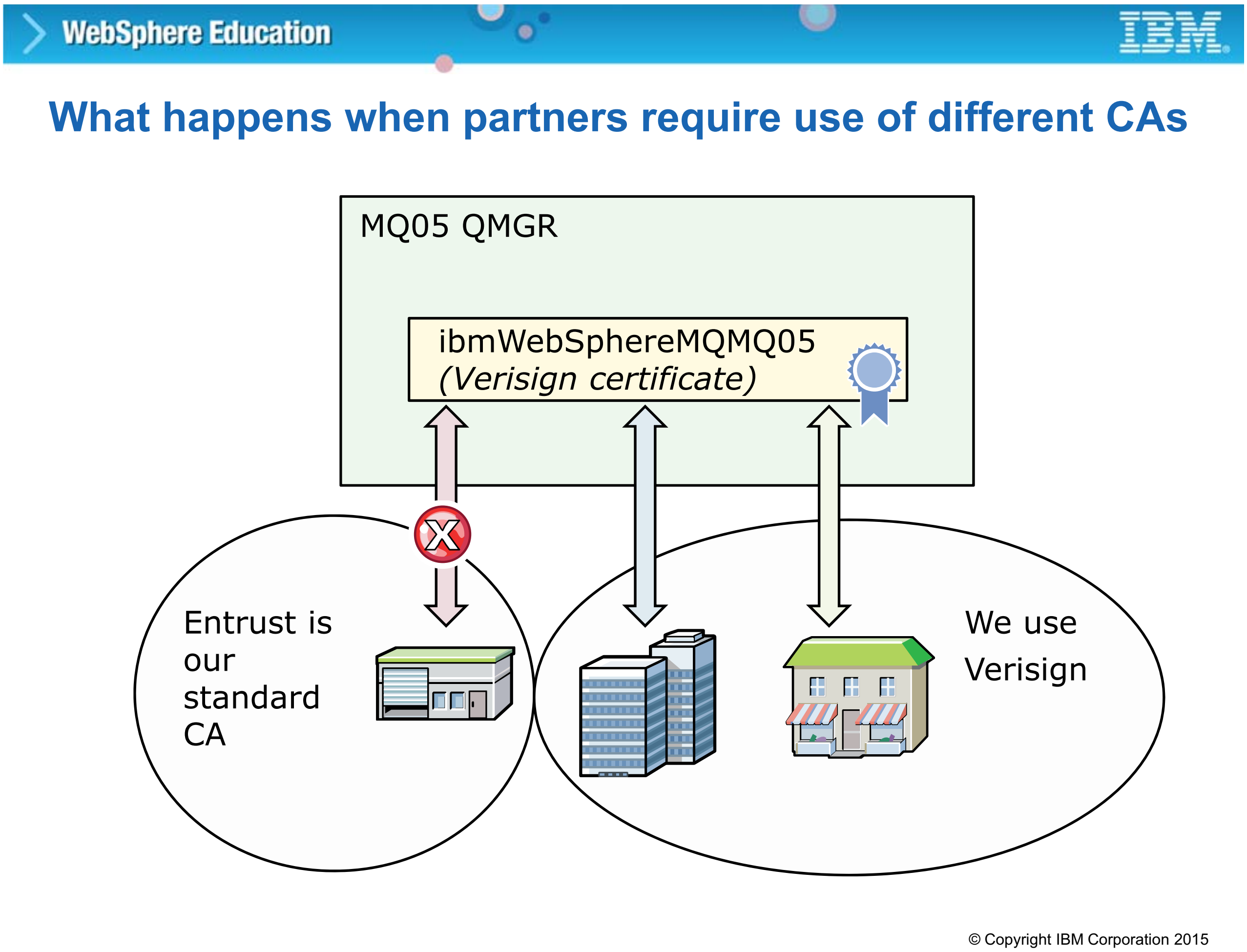


Figure 2-45. What happens when partners require use of different CAs WM3121.1

Notes:

When an organization had connection partners that required use of different CAs, the solution was to create one or more separate queue managers and configure extra queue managers with any additional CA. Before TLS, the authentication of the certificate took place well before the name of which receiver channel to use was known.

With added requirements from large server farms that needed to handle multiple servers, TLS needed to change to pass additional information as part of its handshake. Enhancements to the TLS protocol allow the extra information that is passed during the handshake. This change is called Server Name Indication, or SNI. After SNI was implemented, WebSphere MQ V8 uses SNI to provide a channel name instead of a host name.

- The sender (or client) end of the channel is enhanced to put the channel name into the Server Name Indication (SNI) hint for the TLS Handshake.
- The receiver (or server-conn) end of the channel is enhanced to retrieve the channel name from the SNI hint, and select the appropriate certificate based on the CERTLABEL attribute.

WebSphere Education

TLS and Server Name Indication (SNI)

- What is SNI?
  - A TLS protocol extension
  - Provides the name that a client is attempting to connect to at the start of the handshake process
  - Allows a server to present different certificates at the same IP address and port location
- Before use of SNI technology, channel name was not yet known at the start of the SSL handshake
- IBM MQ V8 uses SNI technology to pass the channel name before the TLS/SSL handshake

Figure 2-46. TLS and Server Name Indication (SNI) WM3121.1

Notes:

The TLS server name indication, or SNI extension to TLS, provides space in the TLS handshake for a server name or other data that is used to correlate information about the connection.

IBM MQ V8 now uses SNI technology to correlate the channel name connection and is able to select a per-channel certificate.

Since SNI is a TLS enhancement that IBM MQ started to use with V8, it is easy to remember that to use multiple CA certificates in the same queue manager:

- Both sides of the connection must be at IBM MQ V8
- Both sides of the connection must use an SNI enabled TLS CipherSpec

IBM MQ V8 introduced the CERTLABL parameter for the queue manager and the channel object definitions to use the capability that is introduced by SNI.

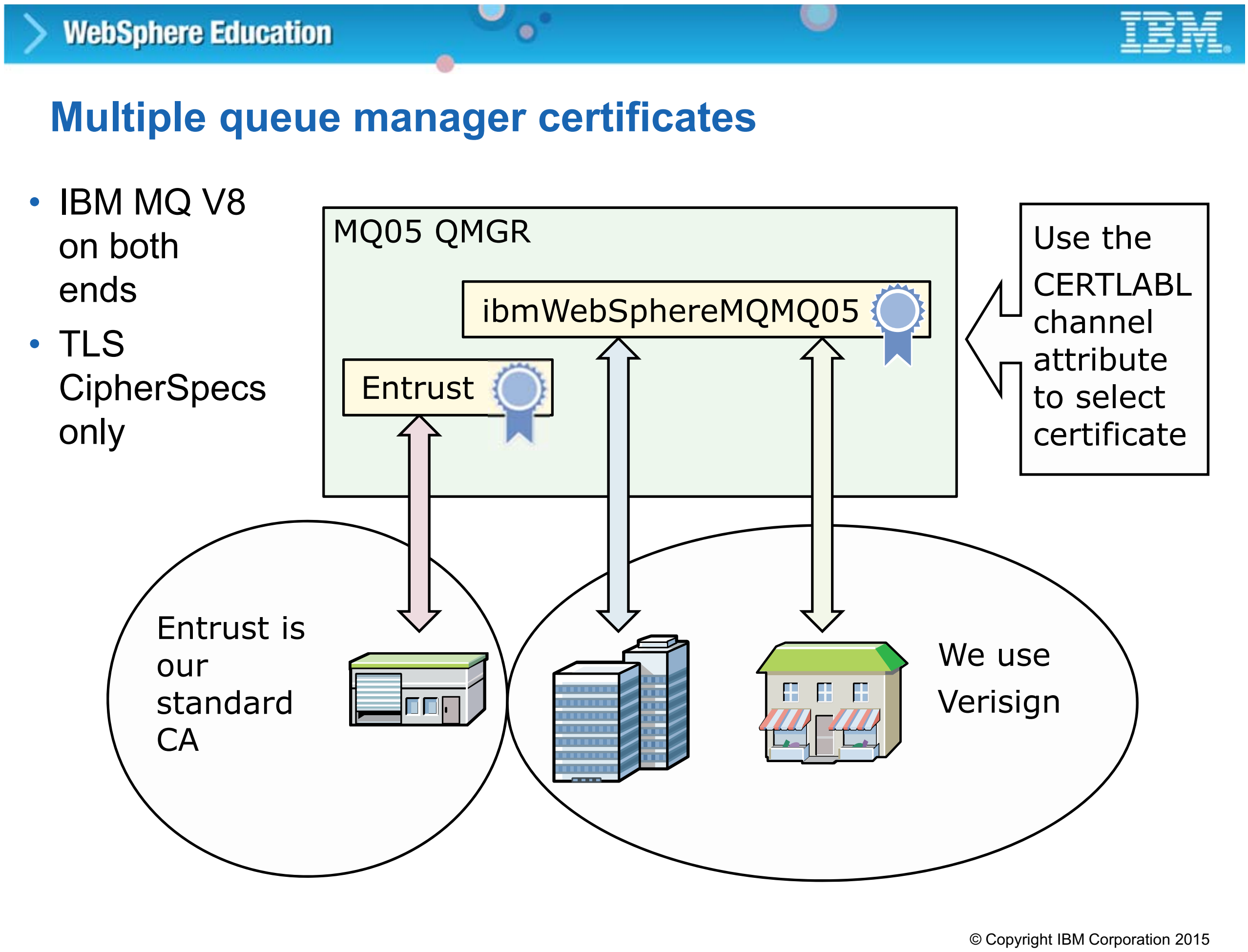


Figure 2-47. Multiple queue manager certificates WM3121.1

Notes:

You now use of the CERTLABL attribute to use more than one CA in the same queue manager.

The CERTLABL attribute in a channel is used by IBM MQ when both sides of the connection are at IBM MQ V8 and use specific TLS CipherSpecs.

If a channel encounters a queue manager at a lower version, it ignores the CERTLABL parameter and functions like a channel in a lower IBM MQ version.

WebSphere Education IBM

Ensuring use of the correct certificate

- Single queue manager certificate
  – ALTER QMGR CERTLABL('CertName')
  – No CERTLABL value in channels
- Multiple certificate requirements
  – Both sides of the connection must be at IBM MQ V8
  – Must use selected TLS Cipher Specs
  – Channel CERTLABL attribute contains certificate name
- Certificate matching
  – What happens if a rogue CA creates a certificate with the same name as a partner certificate?
  – Certificate channel authentication matching rules

Figure 2-48. Ensuring use of the correct certificate WM3121.1

Notes:

A recap of the possible certificate options in queue managers:

- Queue managers before IBM MQ V8 have a single certificate.
- Channels in IBM MQ V8 queue managers that connect to lower-level IBM MQ channels cannot use the CERTLABL attribute. CERTLABL is ignored if a lower version IBM MQ is detected.
- Channels with IBM MQ V8 on both sides of the connection can use multiple certificates and have channels that use different certificates as identified by the channel CERTLABL attribute.
- Administrators with IBM MQ V8 on both sides of the connection have the option to use the CERTLABL attribute as a tool to renew certificates by bringing in the renewed certificate under a different label name.

The question on the rogue CA is answered in the channel authentication unit.

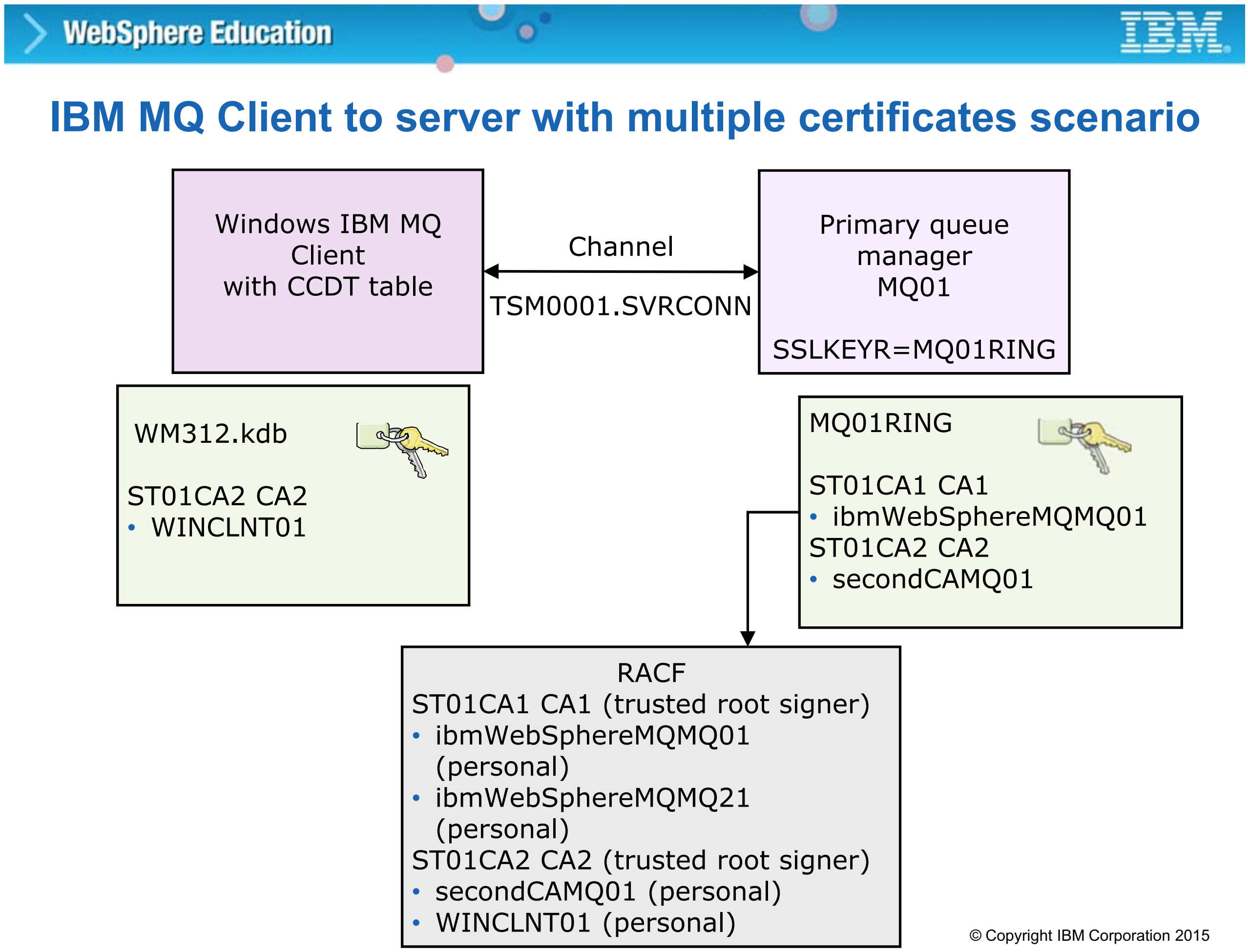


Figure 2-49. IBM MQ Client to server with multiple certificates scenario WM3121.1

Notes:

You now look at two new concepts:

- How to implement multiple certificates in the same queue manager
- How to connect an IBM MQ client to the z/OS queue manager

The view of the RACF and MQ01RING figure is cumulative and shows all certificates. MQ21 can be ignored.

In this scenario, RACF acts like the CA and generates root and personal certificates for the MQ01 queue manager and the IBM MQ client. Since the display involves having more than one CA in the same queue manager, an extra CA, ST01CA2 is created to sign the queue manager and client certificates. The steps that are required *on the z/OS side* for this scenario are:

- Create ST01CA2 root signer
- Create second MQ01 certificate, secondCAMQ01, signed by ST01CA2
- Associate both new certificates to the MQ01 ring
- Create the IBM MQ client personal certificate that is signed by ST01CA2
- Export both the second root and client certificates to a file for transfer to the client
- When the client is ready, alter the server connection channel SSLCIPH with the CipherSpec

WebSphere Education IBM

Create second CA

```
RACDCERT CERTAUTH GENCERT -
SUBJECTSDN(CN('STUDENT 01 CA2') -
T('IBM MQ') -
OU('TRAINING') -
O('IBM') -
L('HURSLEY') -
SP('HAMPSHIRE') -
C('UK')) -
WITHLABEL('ST01CA2') -
KEYUSAGE(CERTSIGN)
```


Figure 2-50. Create second CA WM3121.1

Notes:

Creating CA root in RACF is business as usual. Date ranges are allowed to use initial or default values. Be careful when creating subsequent personal certificates and their date ranges.

WebSphere Education IBM

Create second CA queue manager certificate

```
RACDCERT ID(TSM0001) GENCERT -
SUBJECTSDN(CN('MQ01 PERSONAL CA2') -
T('QMGR') -
OU('TRAINING') -
O('IBM') -
L('HURSLEY') -
SP('HAMPSHIRE') -
C('UK')) -
WITHLABEL('secondCAMQ01') -
SIGNWITH(CERTAUTH LABEL('ST01CA2'))
RACDCERT LIST(LABEL('secondCAMQ01')) ID(TSM0001)
```


Figure 2-51. Create second CA queue manager certificate WM3121.1

Notes:

The certificate in this slide is the second certificate for queue manager MQ01 signed by the second CA root.

WebSphere Education

Connect the second CA and queue manager certificates

```
RACDCERT ID(TSM0001) CONNECT(CERTAUTH LABEL('ST01CA2') -
RING(MQ01RING))
RACDCERT ID(TSM0001) CONNECT(ID(TSM0001)
LABEL('secondCAMQ01') -
RING(MQ01RING))
RACDCERT LISTRING(*) ID(TSM0001)
```


Figure 2-52. Connect the second CA and queue manager certificates WM3121.1

Notes:

Associate the new certificates to the MQ01 ring.

WebSphere Education

Create client certificate that is signed by second CA

```
RACDCERT ID(TSM0001) GENCERT -
SUBJECTSDN(CN('MQ01 WIN CA2 PERS') -
T('CLIENT') -
OU('TRAINING') -
O('IBM') -
L('ORLANDO') -
SP('FLORIDA') -
C('US')) -
WITHLABEL('WINCLNT01') -
SIGNWITH(CERTAUTH LABEL('ST01CA2'))
RACDCERT LIST(LABEL('WINCLNT01')) ID(TSM0001)
```


Figure 2-53. Create client certificate that is signed by second CA WM3121.1

Notes:

This slide shows the certificate for use by the IBM MQ client in a distributed IBM MQ platform, signed by the second CA, ST01CA2.

Export the second CA certificate and the client certificate

- Download data sets to client in binary format

```
RACDCERT CERTAUTH EXPORT ( -
LABEL('ST01CA2')) -
DSN('TSM0001.DIGCERT.SIGNER.DER') -
FORMAT(CERTDER)
```

```
RACDCERT ID(TSM0001) -
EXPORT (LABEL('WINCLNT01')) -
DSN('TSM0001.DIGCERT.WIN.PKCS12') -
FORMAT(PKCS12DER) -
PASSWORD('ZOSPWD')
```


Figure 2-54. Export the second CA certificate and the client certificate WM3121.1

Notes:

Export the two certificates to the IBM MQ client. You use the RACDCERT EXPORT, but notice the differences:

- The root CA uses the `RACDCERT CERTAUT EXPORT` command.
- The format is `.der` and the file contains one certificate.
- The windows personal certificate is created by using the pkcs12 format.
- The windows export contains both the private and the public keys, so a password is used to protect mostly the private key. The password needs to be provided before this file can be used.

WebSphere Education

Make required changes to server connection channel

- Check that the client channel starts before changing the channel attributes to use SSL/TLS

```
ALTER CHL(TSM0001.SVRCONN) CHLTYPE(SVRCONN) +
   SSLCIPH(TLS_RSA_WITH_AES_128_CBC_SHA256) +
   CERTLABL('secondCAMQ01')
```


Figure 2-55. Make required changes to server connection channel WM3121.1

Notes:

Before proceeding with this step, check that the client channel starts successfully without SSL/TLS. If any problems are found, make any necessary corrections before applying SSL/TLS to the channel.

WebSphere Education

Tasks on the IBM MQ V8 Client

- Create a key database by using the key management utility that is provided in the distributed platform
- Add the signer and personal certificates that are downloaded from the RACF CA to the key database
- Ensure that the IBM MQ Client can locate the key database
  - `mqclient.ini` SSL stanza SSLKeyRepository
  - MQSSLKEYR environment variable
- Alter the client channel in the client channel connection table (CCDT) by using the client version of the `runmqsc` utility (`runmqsc -n`)

```
ALTER CHL(TSM0001.SVRCONN) CHLTYPE(CLNTCONN) +
   SSLCIPH(TLS_RSA_WITH_AES_128_CBC_SHA256) +
   CERTLABL('WINCLNT01')
```


Figure 2-56. Tasks on the IBM MQ V8 Client WM3121.1

Notes:

Before IBM MQ V8, a client channel definition table, or CCDT, had to be created in a queue manager and downloaded to the client.

When IBM MQ V8 queue managers are used with IBM MQ clients, you must have a V8 client and create the CCDT in the client side by using `runmqsc -n` to use IBM MQ V8 functions.

Note how both the CiperSpec and the certificate name attributes are updated. IBM MQ clients default to a label composed of the string `ibmwebspheremqxxxx` all in lowercase, where `xxxx` is the client user logon ID. Since the label of the certificate that is created for the client is `WINCLNT01`, the `CERTLABL` attribute must be used.

WebSphere Education

Overview of SSL/TLS tools on UNIX, Linux, and Windows

- Different types of certificate or key databases possible, such as
  - Certificate Management System (CMS) database **.kdb**
  - Java keystore database that is used by Java applications such as **IBM MQ Explorer.jks**
- The Global Security Kit (GSKit) provides tools to manage certificates
  - **runmqakm** (GSKCapiCmd): Command-line interface
  - **runmqckm** (iKeycmd): Command-line interface
  - **strmqikm** (iKeyman): For iKeyman GUI
- **runmqakm**
  - FIPS 140-2 compliant
  - Can be configured to operate in a FIPS-compliant manner by using **-fips** option
  - Supports stronger encryption of the key repository by using **-strong** option
  - Supports Elliptic curve certificates and requests
- **runmqckm** supports JKS and Java Cryptography Extension Key Store (JCEKS) key repository file formats

Figure 2-57. Overview of SSL/TLS tools on UNIX, Linux, and Windows WM3121.1

Notes:

This slide presents an overview of the different options to consider when working with SSL/TLS in distributed IBM MQ platforms.

For purposes of the lab, you use the `runmqakm` command.

WebSphere Education

Typical Windows CMS certificate database setup (1 of 2)

- Create the CMS key database

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

- Add the CA certificate from the `.der` file

```
runmqakm -cert -add -db C:\certs\wm312.kdb -pw passw0rd -label
  ST01CA2 -format binary -trust enable -file
  C:\certs\SIGNER.der
```

- Import the personal certificate from the `.pkcs12` file

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


Figure 2-58. Typical Windows CMS certificate database setup (1 of 2) WM3121.1

Notes:

The slide presents a *partial* summary of work that is required in a distributed IBM MQ platform, focused on the `runmqakm` tool commands that are used in a Windows IBM MQ client.

You must ensure to indicate to the client where to find the key database, and make the required changes to the `CLNTCONN` object in the client side `CCDT`.

WebSphere Education

Typical Windows CMS certificate database setup (2 of 2)

- List contents of the key database

```
runmqakm -cert -list all -db wm312.kdb -pw passw0rd
  -type cms
5724-H72 (C) Copyright IBM Corp. 1994, 2014.
Certificates found
* default, - personal, ! trusted, # secret key
!       ST01CA2
-       WINCLNT01
```


Figure 2-59. Typical Windows CMS certificate database setup (2 of 2) WM3121.1

Notes:

This display lists the contents of the key database. Use the command to confirm successful completion of the SSL/TLS work in the client.

WebSphere Education

IBM

Unit summary

- Explain the security considerations for IBM MQ channels
- Describe symmetric and asymmetric cryptography
- Explain the role of a certificate, a digital signature, and a certificate authority
- Distinguish between a self-signed certificate and a certificate-authority signed certificate
- Describe how SSL/TLS works to secure communications
- Distinguish between TLS and SSL cipher suites
- Describe how to implement SSL/TLS in a z/OS queue manager
- Describe how to create a key ring and associate certificates with it
- Explain how to implement SSL/TLS in a channel
- Describe how to request and obtain a certificate from a certificate authority
- Explain how to configure use of multiple certificates for the same queue manager
- Explain how to configure SSL/TLS for an IBM MQ client

Figure 2-60. Unit summary WM3121.1

Notes:

WebSphere Education

Checkpoint questions (1 of 2)

1. A **CipherSpec** is the combination of
   a. The hash or message digest and the number of encryption bits
   b. The encryption algorithm and the message authentication code algorithm
   c. The standards that are mandated by the National Security Agency (NSA) Suite B spec
   d. The X.509 components of the certificate request sent to a certificate authority

2. True or false. Use of RACDCERT GENCERT to create a personal certificate without adequate signer certificate valid date checks can lead to a certificate created with NOTRUST status.

Figure 2-61. Checkpoint questions (1 of 2) WM3121.1

Notes:

Write your answers here:

1.

2.

WebSphere Education IBM

Checkpoint questions (2 of 2)

3. Channel attribute used to require validation of the client certificate
   a. SSLCAUTH
   b. CERTLABL
   c. SSLKEYP
   d. SSLPEER

4. Choose the best answer. Only IBM MQ V8 queue managers and clients can use multiple queue manager certificates because:
   a. The later TLS `CipherSpecs` use server name indication (SNI) technology
   b. The CERTLABL parameter is not available in earlier IBM MQ versions
   c. IBM V8 queue managers are not restricted to use the standard label names
   d. Only IBM MQ V8 uses the TLS SNI technology to provide the channel name in the TLS handshake

Figure 2-62. Checkpoint questions (2 of 2) WM3121.1

Notes:

Write your answers here:

3.

4.

WebSphere Education

Checkpoint answers (1 of 3)

1. A `CipherSpec` is the combination of
   a. The hash or message digest and the number of encryption bits
   b. The encryption algorithm and the message authentication code algorithm
   c. Standards mandated by the National Security Agency (NSA) Suite B specification
   d. The X.509 components of the certificate request file sent to a certificate authority
   Answer: b. The encryption algorithm and MAC algorithm.

2. True or False. Use of RACDCERT GENCERT to create a personal certificate without adequate signer certificate valid date checks can lead to a certificate created with NOTRUST status.
   Answer: True. The valid date range for a personal certificate must be equal to or within the valid date ranges of the signer certificate.

Figure 2-63. Checkpoint answers (1 of 3) WM3121.1

Notes:

WebSphere Education IBM

Checkpoint answers (2 of 3)

3. Channel attribute used to require validation of the client certificate
   a. SSLCAUTH
   b. CERTLABL
   c. SSLKEYP
   d. SSLPEER
   Answer: a. A client certificate is validated by setting SSLCAUTH to `REQUIRED`

Figure 2-64. Checkpoint answers (2 of 3) WM3121.1

Notes:

WebSphere Education

Checkpoint answers (3 of 3)

4. Choose the best answer. Only IBM MQ V8 queue managers and clients can use multiple queue manager certificates because:
   a. The later TLS **`CipherSpecs`** use server name indication (SNI) technology
   b. The CERTLABL parameter is not available in earlier IBM MQ versions
   c. IBM V8 queue managers are not restricted to use the standard label names
   d. Only IBM MQ V8 uses the TLS SNI technology to provide the channel name in the SNI hint of the TLS handshake
   Answer: d. SNI is an enhancement to TLS used by IBM MQ to pass the channel name in the SNI hint TLS extension.

Figure 2-65. Checkpoint answers (3 of 3) WM3121.1

Notes:

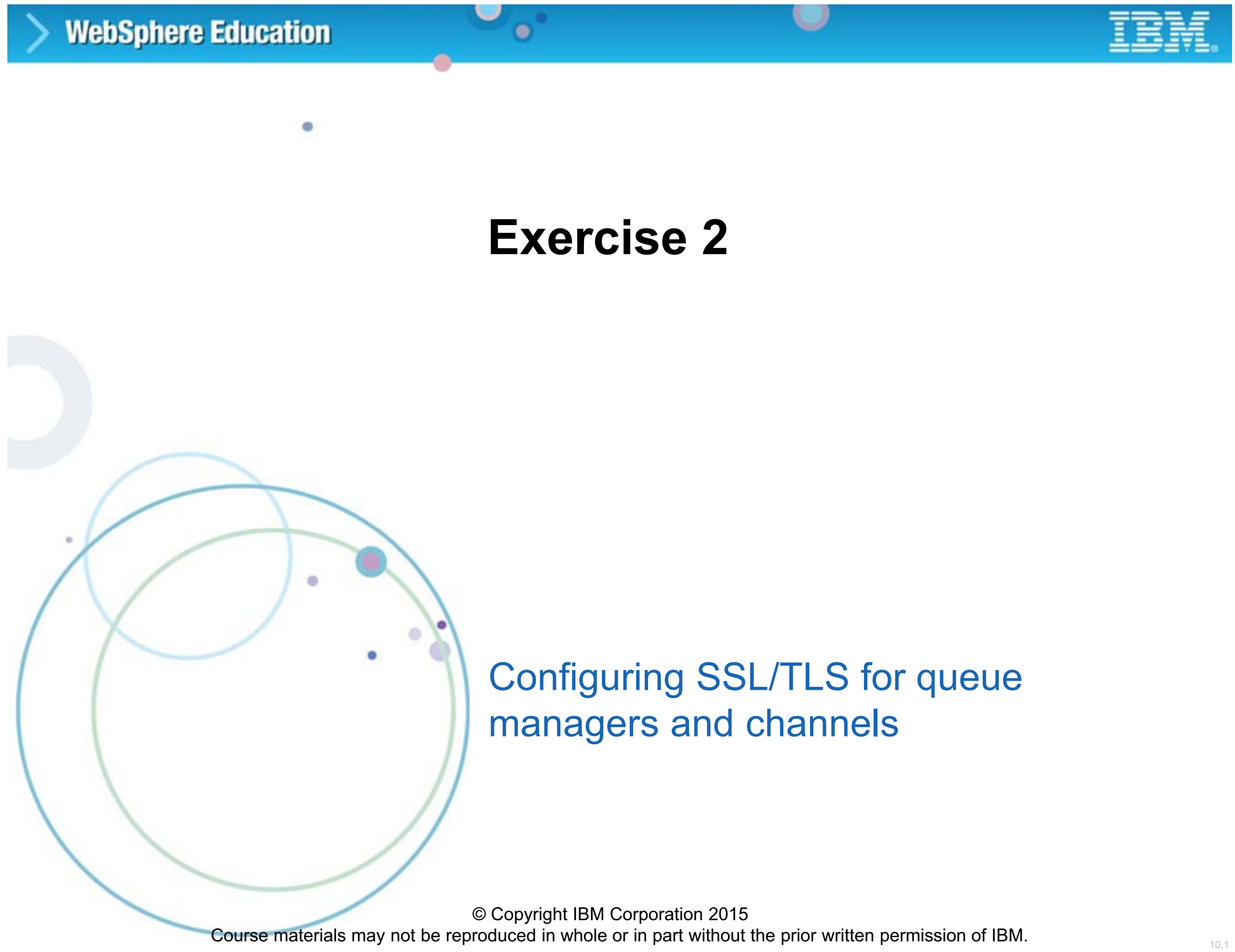


Figure 2-66. Exercise 2 WM3121.1

Notes:

WebSphere Education

Exercise objectives (1 of 2)

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

Figure 2-67. Exercise objectives (1 of 2) WM3121.1

Notes:

WebSphere Education

Exercise objectives (2 of 2)

- For the first part of the exercise, scenario 1, you
  - Work with the RACF RACDCERT utility to generate CA signers and queue manager certificates
  - Learn how to check the certificate valid date ranges
  - Associate the certificates to specific queue managers by using a certificate ring
  - Enable queue managers to use SSL/TLS
  - Configure selected channels to use SSL/TLS
- In the second part of the exercise, scenario 2, you
  - Extend the RACF RACDCERT work that is completed in part 1 to add a second CA signer and a second queue manager certificate by using a non-standard named queue manager certificate label
  - Learn how to configure channels in the situation where a queue manager can have certificates that are signed by more than one CA
  - Learn how to configure SSL/TLS in a Windows IBM MQ V8 client and use the CERTLABL

Figure 2-68. Exercise objectives (2 of 2) WM3121.1

Notes:

Unit 3. Channel security with channel authentication rules

What this unit is about

This unit reviews the channel authentication feature of IBM MQ.

What you should be able to do

After completing this unit, you should be able to:

- Review the use of channel authentication (CHLAUTH) in IBM MQ
- Distinguish between the roles of a firewall, SSL/TLS, and channel authentication
- Describe the various types of channel authentication rules
- Describe channel authentication precedence behavior
- Describe how to set channel authentication rules
- Identify channel authentication best practices
- Review IBM MQ Explorer capabilities to administer channel authentication

How you will check your progress

- Checkpoint questions
- Lab exercises

WebSphere Education IBM

Unit objectives

- Review the use of channel authentication (CHLAUTH) in IBM MQ
- Distinguish between the roles of a firewall, SSL/TLS, and channel authentication
- Describe the various types of channel authentication rules
- Describe channel authentication precedence behavior
- Describe how to set channel authentication rules
- Identify channel authentication best practices
- Review IBM MQ Explorer capabilities to administer channel authentication

Figure 3-1. Unit objectives WM3121.1

Notes:

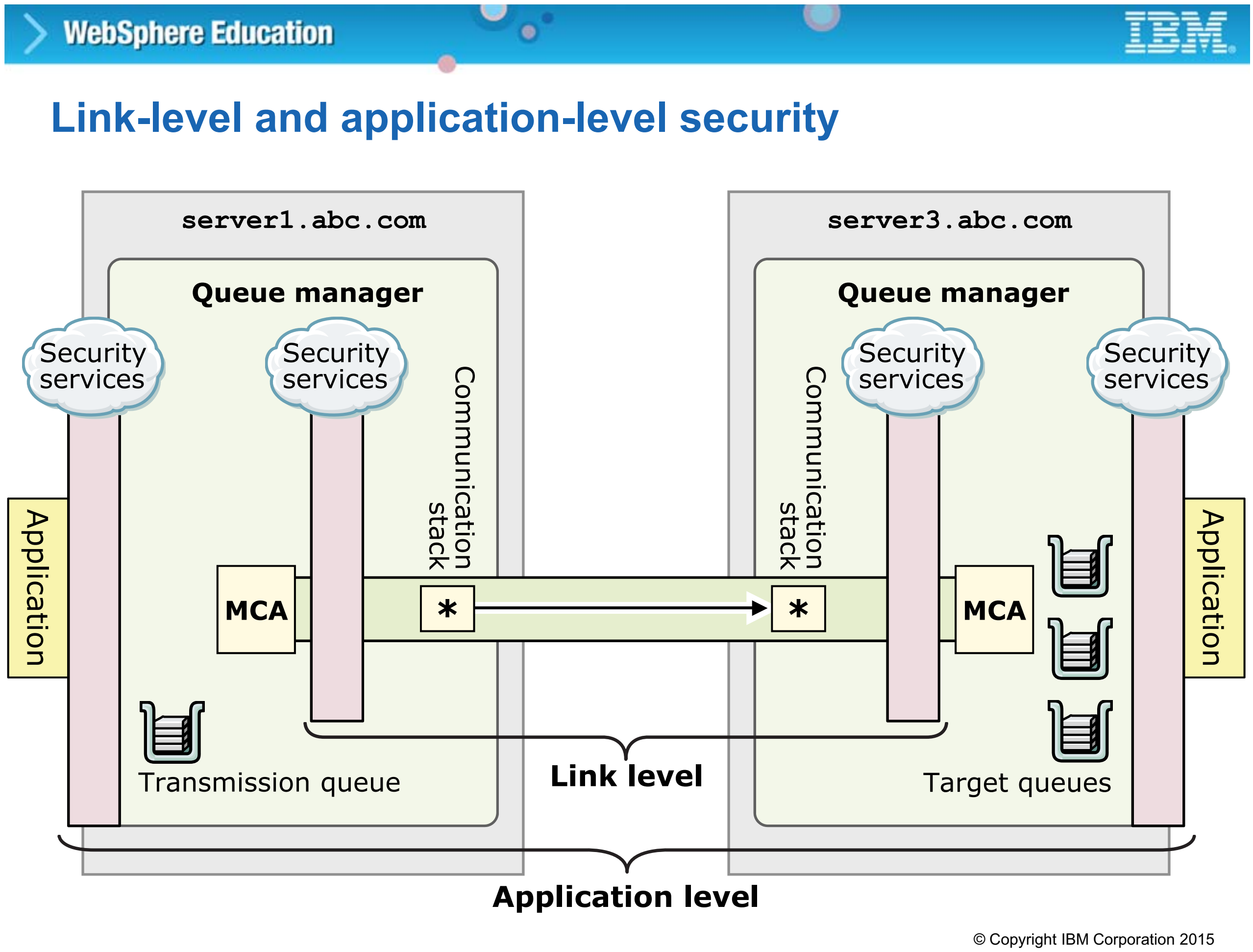


Figure 3-2. Link-level and application-level security WM3121.1

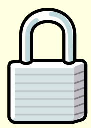
Notes:

Channel authentication falls in the link level security area.

WebSphere Education

Link-level security: Identification and authorization

Channel authentication records:



A security mechanism that enables setting rules to determine which connections are allowed and which are excluded from accessing the queue manager.

- Rules can be set to allow or inhibit connections, privileged users, or a combination of attributes
- Rules can be set to use different details of the incoming connection, such as an IP address or a remote queue manager name
- Mitigates the need to use security exits

Figure 3-3. Link-level security: Identification and authorization WM3121.1

Notes:

It is no longer necessary to code and maintain security exits. Channel authentication provides ways to block out connections based on different criteria. If used with channel authentication and SSL/TLS, it provides security for your data in transit and access to your IBM MQ resources.

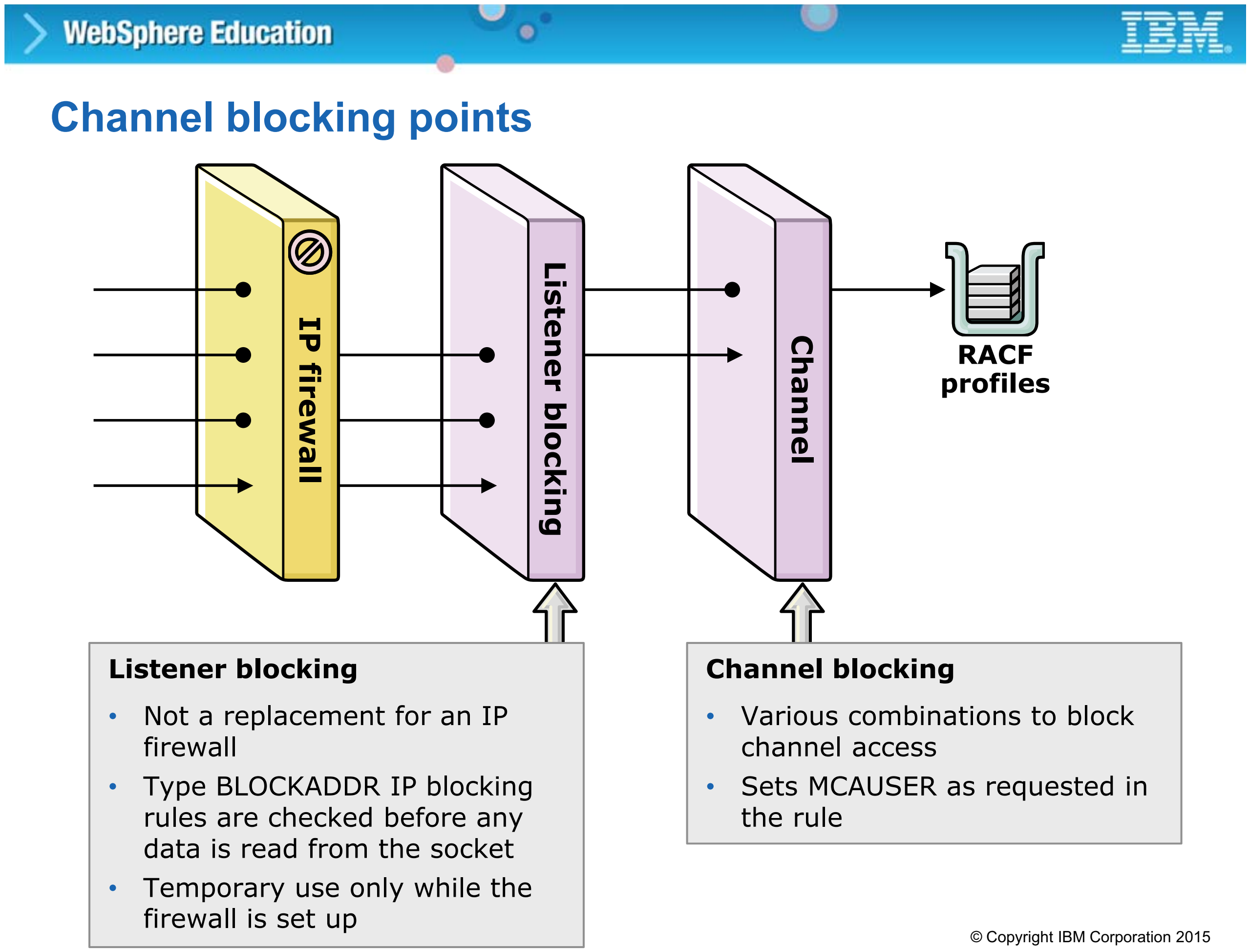


Figure 3-4. Channel blocking points WM3121.1

Notes:

An inbound connection must clear several stages before getting to a queue.

- The IP firewall should be at the forefront of these connections.
- Next point that can be blocked is the listener. Listener blocking should not be a replacement for a firewall, but it is possible to use CHLAUTH to block at the listener as a temporary measure while the firewall rules are in place.
- Listener blocking is accomplished with a BLOCKADDR type of CHLAUTH rule. The different rule types are presented later in this unit.
- After clearing the listener, CHLAUTH might be used at the channel level. The outcome of the CHLAUTH:
  - Can block a connection
  - Behave in a positive manner such as when the rule is used to map a user ID and allow the connection

WebSphere Education IBM

CHLAUTH rule types

CHLAUTH type	Description	Required attribute
BLOCKUSER	Prevents specified users from connecting	USERLIST
BLOCKADDR	• Prevents connections from specified IPs • Works at the listener level	ADDRLIST
SSLPEERMAP	Maps SSL/TLS distinguished names (DNs) to MCAUSER values	SSLPEER
ADDRESSMAP	Maps IP addresses to MCAUSER values	ADDRESS
USERMAP	Maps user IDs to MCAUSER values	CLNTUSER
QMGRMAP	Maps remote queue manager names to MCAUSER values	QMNAME

Figure 3-5. CHLAUTH rule types WM3121.1

Notes:

This slide shows the different types of CHLAUTH rules that might be created. The records take attributes other than the required attributes. Due to the different functions performed, some of these records also require other parameters. There are different interactions among some of the CHLAUTH records and its attributes. For that reason, it is good to consult the IBM Knowledge Center when creating CHLAUTH rules, or after discovering a rule resulted in an error. CHLAUTH does provide good feedback for most errors that are committed when creating rules.

Some rules also work different in distributed platforms than on z/OS. For instance on distributed platforms, it is possible to override a CONNAUTH setting by using a CHLAUTH type ADDRESSMAP with CHCKCLNT, such as:

```
SET CHLAUTH('CONNECT') TYPE(ADDRESSMAP) ADDRESS('192.167.3.4')
USERSRC(CHANNEL) CHCKCLNT(ASQMGR) ACTION(ADD)
```

In z/OS, use of CHCKCLNT and CHCKLOCL with CHLAUTH is not valid.

WebSphere Education

Which rule to use: ADDRESSMAP or BLOCKADDR

- When you use TYPE(ADDRESSMAP), the data already flowed.
  - You know the name of the channel.
  - Full details of the reason the connection was blocked are available in the log, and in the system events if configured.
  - ADDRESSMAP is the type of rule that should be used in most cases.
- TYPE(BLOCKADDR) applies much earlier in the process before any data flows.
  - The channel name is not yet known.
  - Use of BLOCKADDR is to block problem addresses that later get handled with a firewall.
  - BLOCKADDR rules should be used on a temporary basis, and should not permanently substitute a firewall.
  - Channel name attribute must be the generic name, a single asterisk ‘*’.

Figure 3-6. Which rule to use: ADDRESSMAP or BLOCKADDR WM3121.1

Notes:

One frequent question is what type of CHLAUTH rule to use. What is the difference between a BLOCKADDR and an ADDRESSMAP with USERSRC(NOACCESS)? The answer has two parts:

- BLOCKADDR works at the listener level. This type of CHLAUTH rule is applied before the connection sends any data and IBM MQ does not have the incoming channel name. However, the appropriate place for this rule is the firewall. The BLOCKADDR CHLAUTH rule can be used to mitigate the risk of the connection while the firewall rule is in place. Care should be taken to follow up and remove any such rules after the firewall rule is in place.
- ADDRESSMAP with USERSRC(NOACCESS) is the correct method to set IP address rules. When an inbound connection is blocked as a result of one of these rules, the error message that is written to your error log. If events are enabled, the event message is written to the SYSTEM.ADMIN.CHANNEL.EVENT queue that contains details about the blocked connection.
- For BLOCKADDR type rules, this information might not be available.

WebSphere Education

CHLAUTH rules using IP addresses and host names

Rules	Using IP address	Using host names

Listener blocking list	`SET CHLAUTH('*') +`
	`TYPE(BLOCKADDR) +`
	`ADDRLIST('10.5.*','192.168.2.3')+`
	`USERSRC(NOACCESS)`
		Not allowed

Channel-based blocking	`SET CHLAUTH('MQ00.CL.*') +`
	`TYPE(ADDRESSMAP) +`
	`ADDRESS('10.168.*.3')+`
	`USERSRC(NOACCESS)`
		`SET`
		`CHLAUTH(‘MQ00.CL.*')+`
		`TYPE(ADDRESSMAP) +`
		`ADDRESS('*.ibm.com') +`
		`USERSRC(NOACCESS)`

Channel allowed	`SET CHLAUTH('*.SVRCONN') +`
	`TYPE(ADDRESSMAP) +`
	`ADDRESS('10.5.6.*') +`
	`MCAUSER(INGMUSR)`
		`SET`
		`CHLAUTH('*.SVRCONN')+`
		`TYPE(ADDRESSMAP) +`
		`ADDRESS('ac.ibm.com')+`
		`MCAUSER(INGMUSR)`

Qualification of another rule type with an IP address or a host name	`SET CHLAUTH('*') +`
	`TYPE(USERMAP) +`
	`CLNTUSER('NOMAD123') +`
	`ADDRESS('10.4.*')`
	`MCAUSER('INGMUSR')`
		`SET CHLAUTH('*') +`
		`TYPE(USERMAP) +`
		`CLNTUSER('NOMAD123') +`
		`ADDRESS('a*.ibm.*')+`
		`MCAUSER('INGMUSR')`

Figure 3-7. CHLAUTH rules using IP addresses and host names WM3121.1

Notes:

You now look at some examples of CHLAUTH rules that use IP addresses and host names. Host names can be used in most places, except with the BLOCKADDR type rule.

When starting to use CHLAUTH, as a general rule, it is better to start with a rule to block everything, then add rules to allow valid connections. You discuss this approach later in this unit.

WebSphere Education IBM

CHLAUTH actions

- ADD
  - For BLOCKUSER and BLOCKADDR, configuration added to the list
  - If a rule exists for other rule types, command fails
  - ADD is default
- REPLACE
  - Might result in an ADD for SSLPEERMAP, ADDRESSMAP, USERMAP, and QMGRMAP rule types
  - For BLOCKUSER and BLOCKADDR, REPLACE with an empty list works like a REMOVEALL
- REMOVE
  - Deletes a single specified record
  - If the last entry from a list is removed, it works like a REMOVEALL
- REMOVEALL
  - For BLOCKUSER and BLOCKADDR, removes all list members
  - For SSLPEERMAP, ADDRESSMAP, USERMAP, and QMGRMAP it removes all mappings from the channel authentication records

Figure 3-8. CHLAUTH actions WM3121.1

Notes:

The actions that can be taken with channel authentication rules are different than other IBM MQ objects that use `DEFINE`, `ALTER`, and `DELETE`.

Warning

Sometimes a `REPLACE` becomes an `ADD` if the change is not properly qualified with the required fields.

WebSphere Education

Replacing and removing CHLAUTH rules

Key fields

Channel name	Type	Values for type	Address (restrictor)	Source of ID	Mapped ID	User ID and password
SET CHLAUTH(AP1.*) TYPE(SSLPEERMAP) SSLPEER('CN=OrdApp') ADDRESS('1.2.3.4')				USERSRC(MAP) MCAUSER('Ordmgr')		
SET CHLAUTH(ADMIN.SVRCONN) TYPE(ADDRESSMAP) ADDRESS('*.ibm.com')				USERSRC(CHANNEL) CHCKCLNT(REQUIRED)		

```
SET   CHLAUTH('AP1.*') TYPE(SSLPEERMAP) SSLPEER('CN=OrdApp') ADDRESS('1.2.3.4')
      ACTION(REMOVE)

SET   CHLAUTH(ADMIN.SVRCONN) TYPE(ADDRESSMAP) ADDRESS('*.ibm.com')
      USERSRC(MAP) MCAUSER('IBMUSER') ACTION(REPLACE)

SET   CHLAUTH('AP1.*') TYPE(SSLPEERMAP)
      ACTION(REMOVEALL)
```

All the key fields

Figure 3-9. Replacing and removing CHLAUTH rules WM3121.1

Notes:

CHLAUTH records are “SET” with an ACTION attribute to either ADD, REPLACE, REMOVE, or REMOVEALL.

When you start working with CHLAUTH rules, there are subtleties to be aware of:

- ACTION(ADD) is the default.
- When using ACTION(REPLACE) and ACTION(REMOVE), it is critical to use all required key fields to qualify the change or removal. If all the required fields are not used, the expected behavior is that the rule is added instead, since a rule that matched the attributes was not found.
- When a rule gets inadvertently added with an ACTION(REMOVE), and now you have two rules with a similar name, sometimes the best route is to enter an ACTION(REMOVEALL) by using a partial wildcard name for the rules. Then, add the rule as intended.
- *Be careful to use quotation marks, particularly for IDs, wildcards, and host names.*

WebSphere Education IBM

Generic IP addresses

- Single IPv4 address such as 10.15.131.7
- IPv4 address with wildcard, such as:
  - 10.15.131.*
  - 10.15.*
  - 10.15.*.7
  - 10.*.131
- Single or wildcard IPv6 addresses
- Use of a hyphen to indicate a range in IPv4 or IPv6, such as 10.15.131.1-7
- Combination of hyphen and an asterisk in IPv4 or IPv6 addresses, such as 10.15.1.*.1-7
- Must be within dot separators
- No asterisk must precede a trailing asterisk
  - 10.*.131 is valid
  - 10.5.*.* is invalid

Figure 3-10. Generic IP addresses WM3121.1

Notes:

IBM Knowledge Center has a section with the same title as this slide, *Generic IP addresses,* which provides additional information about the use of generic IP addresses.

WebSphere Education

CHLAUTH rule precedence

Higher precedence	Channel name
↑	SSL distinguished name
	Client user ID
	Queue manager name
	IP address
	Host name

```
CHLAUTH('MQ00.*') TYPE(SSLPEERMAP)
SSLPEER('O=”IBM US”')
MCAUSER(INGMUSR)

CHLAUTH('MQ00.*') TYPE(USERMAP)
CLNTUSER('NOMAD123')
MCAUSER(TSM0000)

CHLAUTH('MQ00.*') TYPE(ADDRESSMAP)
ADDRESS('10.6.187.184')
MCAUSER(TSM0000)

CHLAUTH('MQ00.*') TYPE(ADDRESSMAP)
ADDRESS('*.ibm.com')
MCAUSER(IBMUSER)
```

- Channel authentication record selection criteria
  - Channel authentication with a full channel name takes precedence over a name that uses a wildcard
  - SSL DN or TLS DN takes precedence over a user ID, a queue manager name, or an IP address
  - Client ID or queue manager name takes precedence over an IP address
- Within each category, the most specific entry takes precedence
- Precedence also within SSL rules

Figure 3-11. CHLAUTH rule precedence WM3121.1

Notes:

CHLAUTH rules follow precedence rules based on two factors:

- The type of rule, where SSL type rules have the highest weight and host names have the least.
- Specificity rules, that is, the more specific a name, the higher in the precedence rank. For example, if you had an IP address of ‘10.6.187.184’ and another address by using a wildcard such as ‘10.6.*’, the full IP address would precede the partial wildcard address.

Precedence also exits within SSL types. The SSL precedence list can be found under the *Interaction between channel authentication records* topic in the IBM Knowledge Center.

Information

For TYPE=SSLPEERMAP rules, IBM MQ V8 and later also allows to check the issuer certificate by using the CHLAUTH SSLCERTI attribute.

WebSphere Education

SSL peer and issuer DN matching

- Check that the Subject DN of the certificate that is presented matches the Subject DN set in the CHLAUTH rule

```
SET CHLAUTH('BILLING.SVRCONN') +
  TYPE(SSLPEERMAP) +
  SSLPEER('CN="MQ01 WIN CA2 PER",O= "IBM"') +
  MCAUSER('Billing1')
```

- Check that the Issuer DN of the certificate that is presented matches the Issuer DN set in the CHLAUTH rule (IBM MQ V8 and later)

```
SET CHLAUTH('BILLING.SVRCONN') +
  TYPE(SSLPEERMAP) +
  SSLPEER('CN="MQ01 WIN CA2 PER",O="IBM"') +
  SSLCERTI('CN="STUDENT 01 CA2",O="IBM"')
  MCAUSER('Billing1')
```

When the SSLPEER attribute is populated for both the channel and the channel authentication rule, the inbound certificate must match both patterns

Figure 3-12. SSL peer and issuer DN matching WM3121.1

Notes:

SSL matching is at the highest precedence level.

With IBM MQ V8 and later, you can also validate the certificate issuer's distinguished name by using the SSLCERTI attribute.

WebSphere Education IBM

CHLAUTH initial settings

```
MQ00 DIS QMGR CHLAUTH
CSQM201I MQ00 CSQMDRTC  DIS QMGR DETAILS 163
QMNAME(MQ00)
CHLAUTH(ENABLED)
END QMGR DETAILS
```

```
MQ00 DIS CHLAUTH('*')
CSQM293I MQ00 CSQMDRTC 3 CHLAUTH FOUND MATCHING REQUEST CRITERIA
CSQM201I MQ00 CSQMDRTC  DIS CHLAUTH DETAILS 168
CHLAUTH(*)  DESCR(Default rule to disallow privileged users)
TYPE(BLOCKUSER) USERLIST(*MQADMIN)
CSQM201I MQ00 CSQMDRTC  DIS CHLAUTH DETAILS 169
CHLAUTH(SYSTEM.*) DESCR(Default rule to disable all SYSTEM channels)
TYPE(ADDRESSMAP) ADDRESS(*)
USERSRC(NOACCESS)
CSQM201I MQ00 CSQMDRTC  DIS CHLAUTH DETAILS 170
CHLAUTH(SYSTEM.ADMIN.SVRCONN) DESCR(Def rule to allow MQ Explorer access)
TYPE(ADDRESSMAP) ADDRESS(*)
USERSRC(CHANNEL)
 END CHLAUTH DETAILS
```


Figure 3-13. CHLAUTH initial settings WM3121.1

Notes:

The initial settings for CHLAUTH in a queue manager are found in two places:

- First, in the queue manager object CHLAUTH attribute. Default is ENABLED.
- Second, the three default rules. Each rule is explained in the DESCR attribute of the display.

WebSphere Education

Allow access to administrative users

- Problem: Channel TSM0021.XPL is defined with MCAUSER('TSM0021')
  - The channel initiator for the queue manager is running under ID TSM0021
  - After CHLAUTH is enabled, channel TSM0021.XPL is blocked with error message:
    ```
    CSQX776E MQ21 CSQXRESP Channel TSM0021.XPL from
    10.61.23.250 has been  blocked due to userid, Detail:
    MCAUSER(TSM0021)CLNTUSER(Administrator)
    ```
- Solution:
  - Do not remove the initial CHLAUTH rule to disallow privileged users
  - Create a BLOCKUSER rule for the channel
  - The name of the channel should specific
  - Use a bogus user name in the USERLIST parameter
    ```
    SET CHLAUTH(TSM0021.XPL) TYPE(BLOCKUSER) +
      USERLIST('GTMETHRU')
    ```

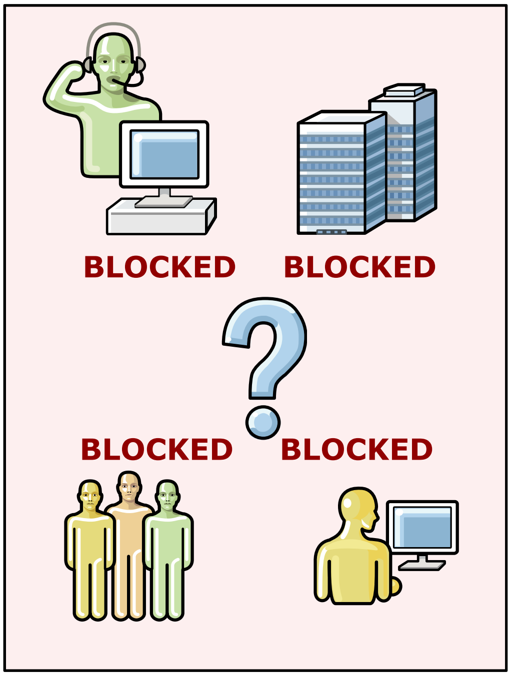

Figure 3-14. Allow access to administrative users WM3121.1

Notes:

The initial CHLAUTH rules should be left in place. To resolve the problem of a connection blocked due to the MCAUSER value set to an administrative user, create a type BLOCKUSER CHLAUTH rule.

WebSphere Education

The back-stop rule and warning mode



- Preferred way to implement CHLAUTH is to inhibit access to all users; then, configure rules to allow known users
- Leave the initial CHLAUTH rules in place
- Called the “back-stop” rule:

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

- WARN attribute
  – Allows connections to continue
  – Captures would-be blocked connection information
  – Default behavior is NO warning (block)

Figure 3-15. The back-stop rule and warning mode WM3121.1

Notes:

This figure depicts potential rejected users when the back-stop rule is implemented. The inbox or cell phone for the IBM MQ administrator is busy. But how else are you going to determine who is connecting to the queue manager to implement the correct rules?

CHLAUTH can be set in warning mode by using attribute WARN(YES). With WARN(YES), CHLAUTH blocked messages do not display in the logs. You must have events that are enabled and check the channel events to determine what channels are blocked.

When you use WARN(YES), CHLAUTH does allow the connections to proceed. In this manner, you can implement the rules, starting with the back-stop rule, and better organize the rules that need to be set for the users. You now identify the known users by review of the event messages, and plan the rules without disrupting access to users.

After the rules are in place and tested and no longer in warning mode, the number of rejected connections should disappear or diminish to a manageable volume.

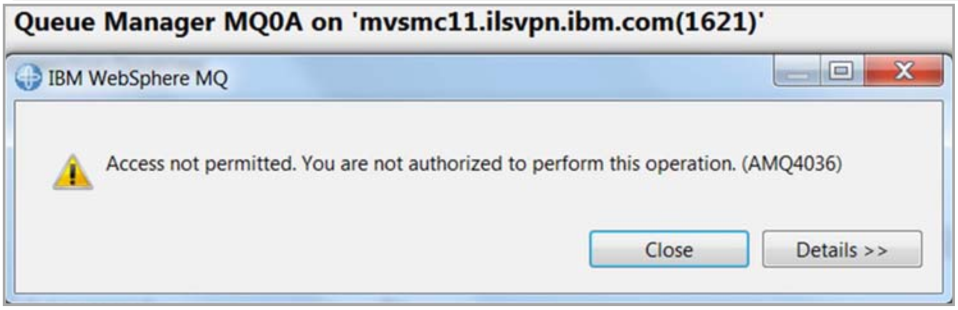
WebSphere Education

Getting started with CHLAUTH (1 of 3)

Start with the “back-stop” rule to keep out everyone.

Back-stop rule can be initially implemented in warning mode.

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



If there are active users, soon there are numerous connection failures after the back-stop rule is set.

```
+CSQX511I MQ0A CSQXRESP Channel WM302.SVRCONN started 181
 connection 10.4.127.184
+CSQX777E MQ0A CSQXRESP Channel WM302.SVRCONN from 10.4.127.184 182
 (10.4.127.184) has been blocked due to USERSRC(NOACCESS), Detail:
 CLNTUSER(TSMUSER)
+CSQX512I MQ0A CSQXRESP Channel WM302.SVRCONN no longer active
```


Figure 3-16. Getting started with CHLAUTH (1 of 3) WM3121.1

Notes:

The recommendation is to start with what is referred to as the “back-stop rule”. This rule stops all access. Soon after you set this rule you see numerous blocking messages in the system log. If you have channel events that are configured, along with the messages in the log there are also event messages that are placed in the `SYSTEM.ADMIN.CHANNEL.EVENT` queue.

If you use warning mode, there are no log messages; you must use events.

WebSphere Education

Getting started with CHLAUTH (2 of 3)

```
The billing client cannot connect …
 +CSQX511I MQ0A CSQXRESP Channel BILLING.MASTER started 192
  connection 10.4.127.184
 +CSQX777E MQ0A CSQXRESP Channel BILLING.MASTER from 10.4.127.184
  (10.4.127.184) has been blocked due to USERSRC(NOACCESS), Detail:
  CLNTUSER(nomad789)
 +CSQX512I MQ0A CSQXRESP Channel BILLING.MASTER no longer active
the orders application cannot connect …
+CSQX777E MQ0A CSQXRESP Channel QMGR1.MQ0A from 10.4.127.184 203
 (10.4.127.184) has been blocked due to USERSRC(NOACCESS), Detail:
 QMNAME(QMGR1)
+CSQX599E MQ0A CSQXRESP Channel QMGR1.MQ0A ended abnormally 204
 connection 10.4.127.184
… and the cluster channels are also challenged
 +CSQX191I MQ0B CSQXRCTL Channel WMADMCLS.MQ0A beginning message 2
  reallocation
 +CSQX777E MQ0A CSQXRESP Channel WMADMCLS.MQ0A from mvsmc11 225
  (10.31.187.59) has been blocked due to USERSRC(NOACCESS), Detail:
  QMNAME(MQ0B)
 +CSQX599E MQ0A CSQXRESP Channel WMADMCLS.MQ0A ended abnormally 226
  connection 10.31.187.59
```


Figure 3-17. Getting started with CHLAUTH (2 of 3) WM3121.1

Notes:

Using IBM MQ events to check CHLAUTH rejections.

In this sample, you use the message handler sample program, option H in the lab environment for this course, to look at the event record.

The return code is at offset X'1C' into the message data: **X'0A11'** or 2577, Channel Blocked.

```
Queue Manager     : MQ00
Queue             : SYSTEM.ADMIN.CHANNEL.EVENT
Message Content :
   PutApplName       : 'MQ00CHIN <=the CHIN region put the event message'
   PutDate           : '20141013'
   PutTime           : '13111657'
   Message Buffer : 200 byte(s)

   00000000 :  0000 0007 0000 0024 0000 0001 0000 002E
   00000010 :  0000 0001 0000 0001 0000 0001 0000 0A11
```

You can scroll down the rest of the message data for more details.

WebSphere Education

Getting started with CHLAUTH (3 of 3)

- Add rules to allow known valid connections:

```
SET CHLAUTH(QMGR1.MQ0A) TYPE(QMGRMAP) ADDRESS('10.4.127.184') +
  DESCR('Access for ORDERS application')  +
  MCAUSER(TSMUSER) QMNAME(QMGR1)

SET CHLAUTH('WMADMCLS.*') TYPE(QMGRMAP) ADDRESS('10.31.*') +
  DESCR('Access for WMADMCLS cluster channels') +
  MCAUSER(TSMUSER) QMNAME(MQ*)

SET CHLAUTH(BILLING.MASTER) TYPE(USERMAP) +
 ADDRESS(10.4.127.184)  +
 CLNTUSER('nomad789') MCAUSER(TSMUSER)   +
 DESCR('Access for BILLING client')

SET CHLAUTH(WM302.SVRCONN)  +
 TYPE(ADDRESSMAP)
 ADDRESS('10.4.127.*')  MCAUSER('TSMUSER') +
 DESCR('Allow MQ Explorer users from 10.4.127.*')
```


Figure 3-18. Getting started with CHLAUTH (3 of 3) WM3121.1

Notes:

This slide shows the rules that are implemented for the rejected applications. CHLAUTH records are a good place to use the DESCR field to document what a rule is for.

A few months after the records are created, or if different IBM MQ administrators need to maintain the queue manager, the description of why the rule was created might be indispensable.

WebSphere Education

Rule checking with MATCH RUNCHECK

- Use of **DIS CHLAUTH** and **MATCH(RUNCHECK)** for rule checking:

```
DIS CHLAUTH('BILLING.MASTER') MATCH(RUNCHECK) +
 ADDRESS('10.4.127.184') +
 CLNTUSER('nomad789')
CSQN205I   COUNT=       3, RETURN=00000000, REASON=00000000
CSQM454I MQ0A
 CHLAUTH(BILLING.MASTER)
 TYPE(USERMAP)
 ADDRESS(10.4.127.184)
 CLNTUSER(nomad789)
 MCAUSER(TSMUSER)
 USERSRC(MAP)
```

- Reverse DNS can be disabled at the queue manager level
  ALTER QMGR REVDNS(DISABLED)

Figure 3-19. Rule checking with MATCH RUNCHECK WM3121.1

Notes:

Use of the DIS CHLAUTH with MATCH(RUNCHECK) option returns the record that is matched by an inbound channel at run time if it does connect to this queue manager. The command can be used for the following purpose:

- Check if DNS is properly set up
- Determine whether hostname-based rules work

Reverse DNS lookup can be disabled at the queue manager level by setting REVDNS(DISABLED).

The MATCH(RUNCHECK) command needs the IP address. The queue manager calls the domain name server (DNS) as if it was an actual inbound connection to find a host name.

- If it finds a host name, CHLAUTH works with host name rules.
- If RUNCHECK does not return a host name, hostname-based rules do not work.

If reverse DNS lookup is disabled with REVDNS(DISABLED) in the queue manager object, an error is displayed if a rule that uses host name runs.

WebSphere Education IBM

CHLAUTH best practices

- Implement the back-stop rule; use warning mode if you need to minimize impact.
- Monitor the outcome of the back-stop rule
  - Check the system log for blocked channels if WARN(NO).
  - Enable channel events and review event messages.
  - Blocked connection messages are *not* generated on the system log for rules with WARN(YES). Must enable channel events to view rejected connections.
- Add one **`CHLAUTH`** rule at a time and test it.
- Count the number of rules before and after you set a rule with ACTION(REPLACE).
- Ensure that you understand the attributes that are required with each type of CHLAUTH rule. Use correct attributes for all ACTIONs.
- Use the **`DESCR`** attribute of the **`CHLAUTH`** definition to identify why the rule is set.
- After you implement and test your rules, run a backup of the rules with CSQUTIL MAKEDEF for the CHLAUTH objects.
- Confirm that your organization scheduled definition backup includes CHLAUTH rules.

Figure 3-20. CHLAUTH best practices WM3121.1

Notes:

This slide contains hints and tips on the preferred approach to implement channel authentication rules. It also contains details that you might need to confirm, such as the automatic inclusion of channel authentication rules in the scheduled definition backups.

WebSphere Education

Use of IBM MQ Explorer for CHLAUTH (1 of 2)

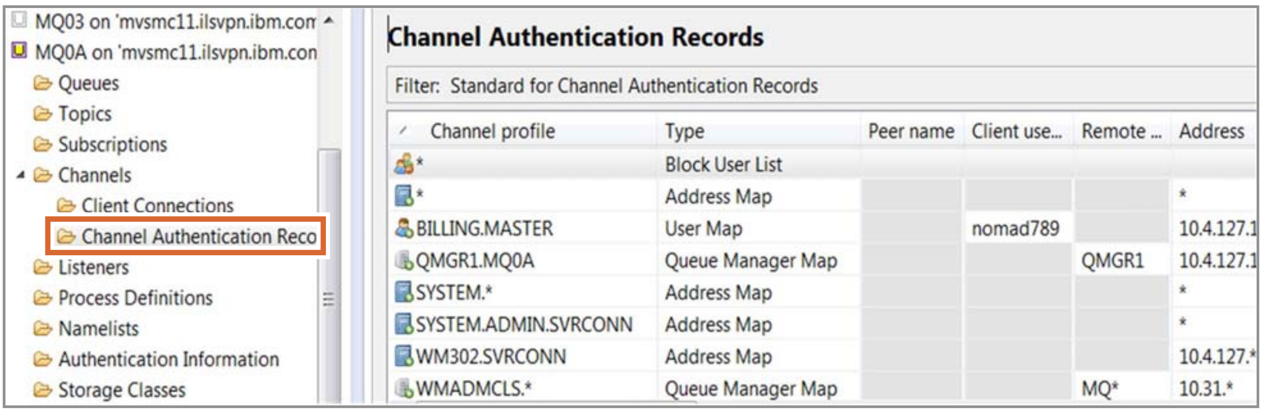


Figure 3-21. Use of IBM MQ Explorer for CHLAUTH (1 of 2) WM3121.1

Notes:

IBM MQ Explorer is updated to process CHLAUTH rules. The CHLAUTH rules are found in the **Channels** folder.

Click the **Channel Authentication** folder for a list of the channel authentication records.

Right-click **Channel Authentication Records** for the CHLAUTH wizard.

WebSphere Education

Use of IBM MQ Explorer for CHLAUTH (2 of 2)

- The panels walk through the definition and present a preview of the corresponding MQSC **`SET CHLAUTH`** command

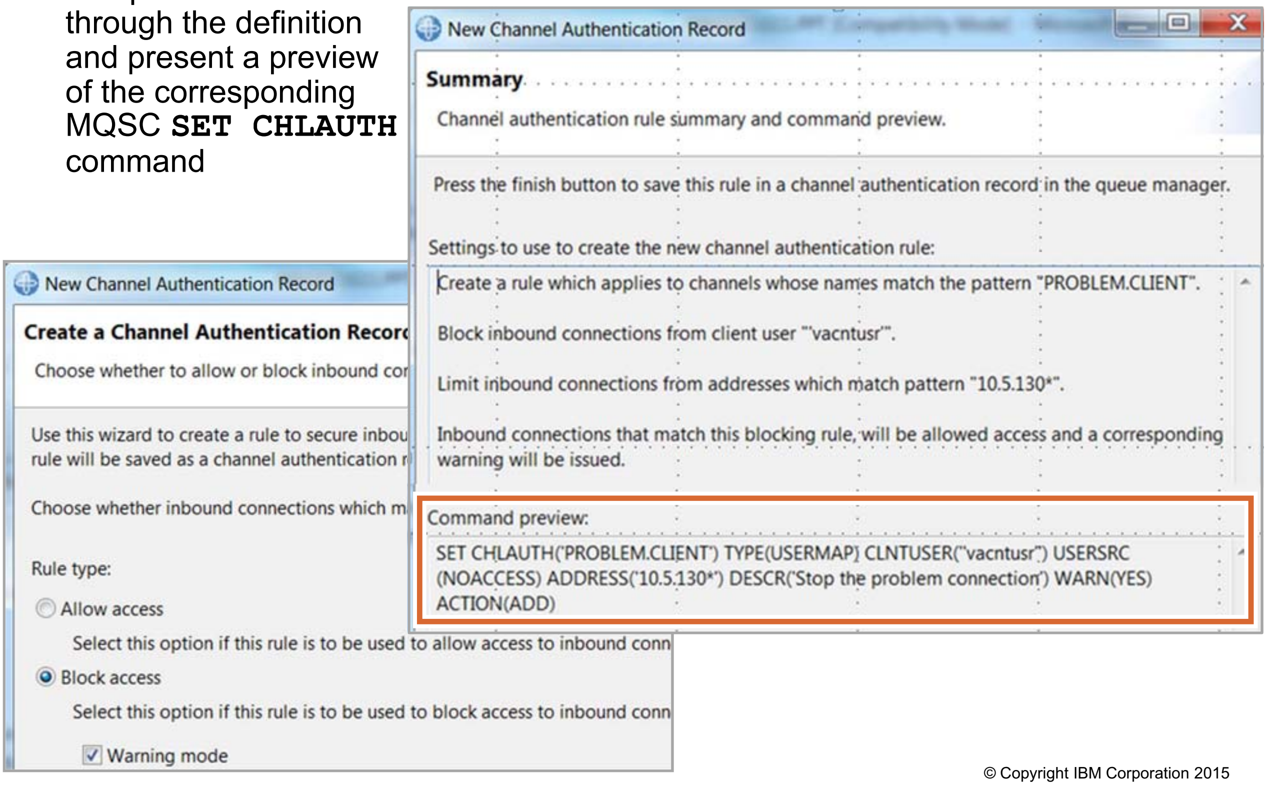


Figure 3-22. Use of IBM MQ Explorer for CHLAUTH (2 of 2) WM3121.1

Notes:

The channel authentication wizard presents the necessary pop-up panels to create a CHLAUTH record.

On the last pane, the wizard displays the command in its MQSC form as it would be run in CSQUTIL.

WebSphere Education

Unit summary

- Review the use of channel authentication (CHLAUTH) in IBM MQ
- Distinguish between the roles of a firewall, SSL/TLS, and channel authentication
- Describe the various types of channel authentication rules
- Describe channel authentication precedence behavior
- Describe how to set channel authentication rules
- Identify channel authentication best practices
- Review IBM MQ Explorer capabilities to administer channel authentication

Figure 3-23. Unit summary WM3121.1

Notes:

WebSphere Education IBM

Checkpoint questions (1 of 2)

1. Which CHLAUTH rule type can be used as a *temporary* workaround for a firewall?
   a. BLOCKUSER
   b. ADDRESSMAP
   c. QMGRMAP
   d. BLOCKADDR

2. True or false. Rule type BLOCKADDR requires use of a host name.

3. When CHLAUTH is enabled, which rule type is used to allow access to administrative users?
   a. BLOCKUSER
   b. USERMAP
   c. SSLPEERMAP
   d. QMGRMAP

Figure 3-24. Checkpoint questions (1 of 2) WM3121.1

Notes:

Write your answers here:

1.

2.

3.

WebSphere Education

Checkpoint questions (2 of 2)

4. Select the best answer. The three initial CHLAUTH rules:
   a. Disallow connections to the administrative queue, the channel initiator, and IBM MQ Explorer
   b. Disallow connections to all SYSTEM.* channels, allow use of SYSTEM.ADMIN.SVRCONN, and block privileged users
   c. Block privileged users, allow use of SYSTEM.ADMIN.SVRCONN, and protect the system command queue
   d. Block privileged users, allow use of SYSTEM.ADMIN.SVRCONN, and block connections at the listener level

5. What attribute of the SET CHLAUTH command is used to identify connection information without blocking connections?
   a. REVDNS(DISABLE)
   b. CHCKCLNT(ASKQMGR)
   c. WARN(YES)
   d. ACTION(WARN)

Figure 3-25. Checkpoint questions (2 of 2) WM3121.1

Notes:

Write your answers here:

4.

5.

WebSphere Education

Checkpoint answers (1 of 2)

1. Which CHLAUTH rule type can be used as a *temporary* workaround for a firewall?
   a. BLOCKUSER
   b. ADDRESSMAP
   c. QMGRMAP
   d. BLOCKADDR

   Answer: d, BLOCKADDR.
2. True or false. Rule type BLOCKADDR requires use of a host name
   Answer: False, BLOCKADDR requires an ADDRLIST parameter; also, the rule or channel name must be generic, '*'.
3. When CHLAUTH is enabled, which rule type is used to allow access to administrative users?
   a. BLOCKUSER
   b. USERMAP
   c. SSLPEERMAP
   d. QMGRMAP

   Answer: a, BLOCKUSER, with a USERLIST attribute that contains a bogus user ID.

Figure 3-26. Checkpoint answers (1 of 2) WM3121.1

Notes:

WebSphere Education

Checkpoint answers (2 of 2)

4. Select the best answer. The three initial CHLAUTH rules:
   a. Disallow connections to the administrative queue, the channel initiator, and IBM MQ Explorer
   b. Disallow connections to all SYSTEM.* channels, allow use of SYSTEM.ADMIN.SVRCONN, and block privileged users
   c. Block privileged users, allow use of SYSTEM.ADMIN.SVRCONN, and protect the system command queue
   d. Block privileged users, allow use of SYSTEM.ADMIN.SVRCONN, and block connections at the listener level

   Answer: b. Block all SYSTEM.* channels except SYSTEM.ADMIN.SVRCONN used for IBM MQ Explorer and disallow privileged users

5. What attribute of the SET CHLAUTH command is used to identify connection information without blocking connections?
   a. REVDNS(DISABLE)
   b. CHCKCLNT(ASKQMGR)
   c. WARN(YES)
   d. ACTION(WARN)

   Answer: c, WARN(YES). REVDNS is not a CHLAUTH attribute; the rest is incorrect.

Figure 3-27. Checkpoint answers (2 of 2) WM3121.1

Notes:

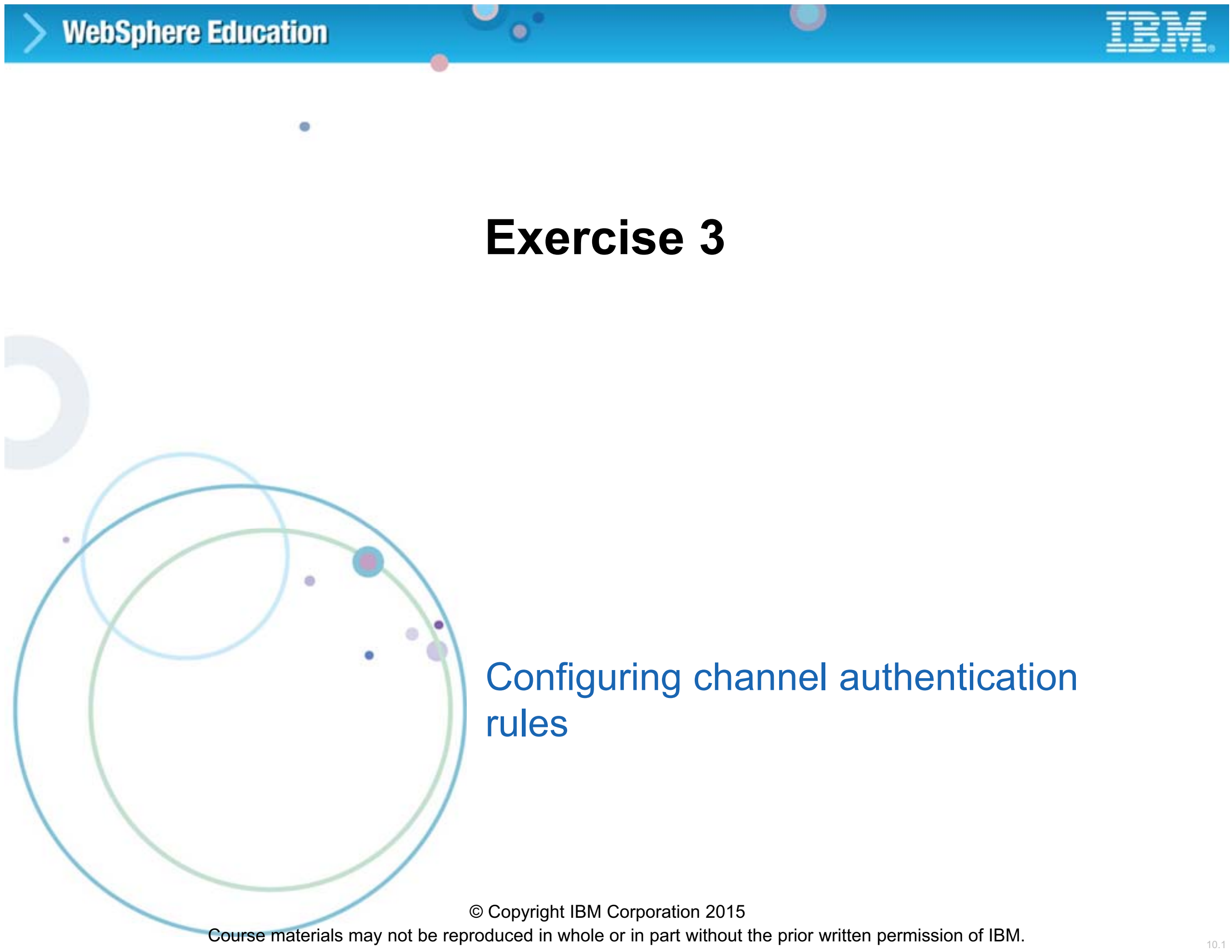


Figure 3-28. Exercise 3 WM3121.1

Notes:

WebSphere Education

Exercise objectives

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

Figure 3-29. Exercise objectives WM3121.1

Notes:

Unit 4. Queue-sharing groups

What this unit is about

This unit describes how queue-sharing groups can be used in the infrastructure and the components that are required to implement shared queues. You also learn how to implement, back up, and restore queue-sharing groups and what overflow or offload options can be used.

What you should be able to do

After completing this unit, you should be able to:

- Describe shared queues and their architectural role
- List the components that are used by queue-sharing groups
- Explain the relationship of the coupling facility capabilities and the CFLEVEL
- Describe queue-sharing group offload and overflow options
- Describe scenarios that are applicable to use of shared message data sets (SMDS) or storage class memory (SCM)
- Explain how messages are stored in a coupling facility application structure
- Contrast how the coupling facility application and administrative structures are used
- Describe the coupling facility resource management (CFRM) attributes to consider in queue-sharing groups
- Explain queue manager and structure failure, persistence, connectivity, and recovery as they apply to queue-sharing groups
- List IBM MQ administrative commands that are related to queue-sharing groups
- Explain how to configure and test a queue-sharing group

How you will check your progress

- Checkpoint questions
- Lab Exercises

WebSphere Education

Unit objectives

- Describe shared queues and their architectural role
- List the components that are used by queue-sharing groups
- Explain the relationship of the coupling facility capabilities and the CFLEVEL
- Describe queue-sharing group offload and overflow options
- Describe scenarios that are applicable to use of shared message data sets (SMDS) or storage class memory (SCM)
- Explain how messages are stored in a coupling facility application structure
- Contrast how the coupling facility application and administrative structures are used
- Describe the coupling facility resource management (CFRM) attributes to consider in queue-sharing groups
- Explain queue manager and structure failure, persistence, connectivity, and recovery as they apply to queue-sharing groups
- List IBM MQ administrative commands that are related to queue-sharing groups
- Explain how to configure and test a queue-sharing group

Figure 4-1. Unit objectives WM3121.1

Notes:

WebSphere Education

What are shared queues

- Queues whose messages are kept in a coupling facility
- Queue managers in the same z/OS sysplex access the same shared queues by using coupling facility connectivity
- Queue managers configured to share the same queues by using a coupling facility are called a queue-sharing group
- Provide high availability in an IBM MQ infrastructure

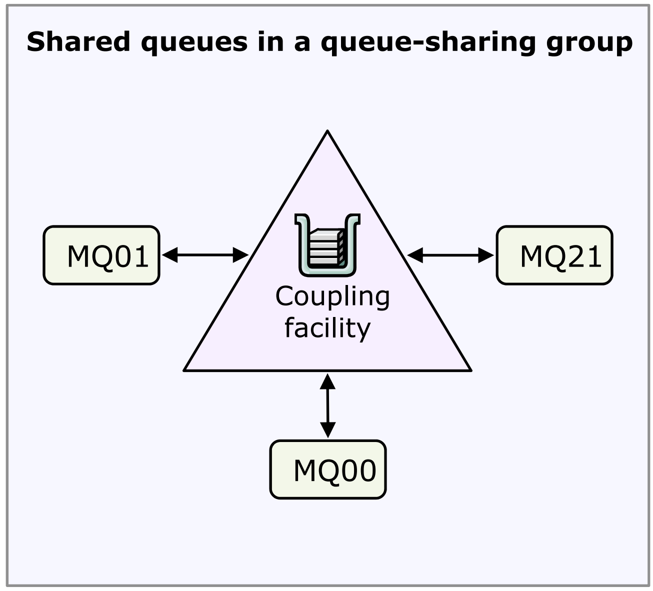


Figure 4-2. What are shared queues WM3121.1

Notes:

A shared queue is a special type of a local queue. The messages on a shared queue are stored in a list structure of the coupling facility (CF). These messages can be accessed from any queue manager in the sysplex, even if the originating queue manager fails.

The definition objects for the shared queues, together with other shareable definition objects, are stored in a DB2 shared database. Non-shared, regular local queues are still there and continue to be used by the queue manage.

Note

Developing applications for shared queues is similar to developing any other IBM MQ application. Developers must review section *Application Programming with shared queues* in the IBM Knowledge Center as part of a plan to use shared queues in a new or existing application.

Two key items to review are:

- Special details on use of correlation or message ID
- Dynamic queue consideration, where only permanent dynamic queues are allowed

However, the entire *Application Programming with shared queues* section should be reviewed.

WebSphere Education

Queue-sharing group components

- DB2 data sharing group
- Coupling facility (CF)
  - Administrative structures
  - Application structures
- IBM MQ CFSTRUCT objects
- Shared message data sets (SMDS)
- Flash express cards
  - Also referred to as storage class memory (SCM)
- IBM MQ object definitions such as queues and channels

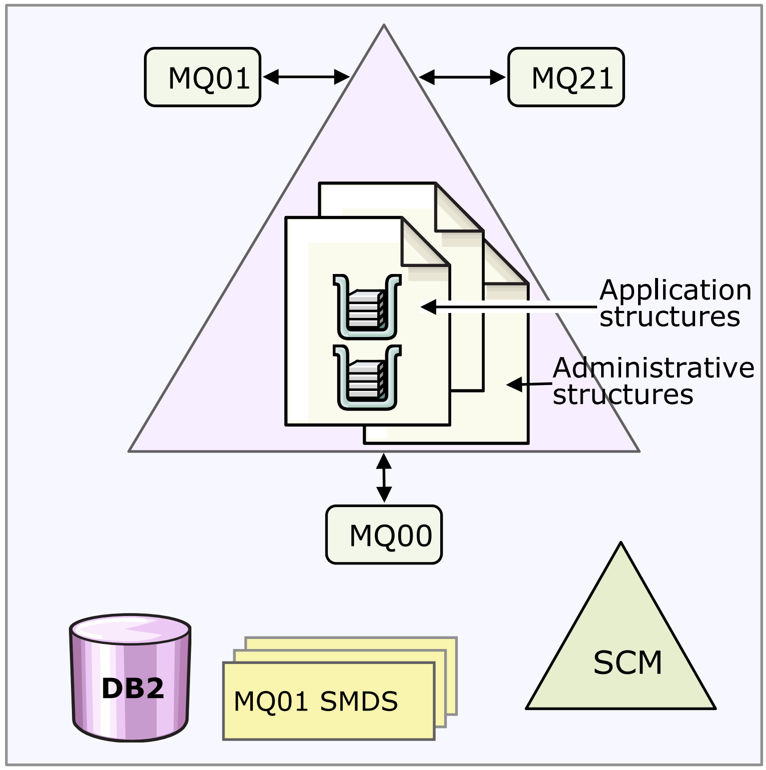


Figure 4-3. Queue-sharing group components WM3121.1

Notes:

The basic components of a queue-sharing group are:

1. The coupling facility and the structures that hold administrative information and messages.
2. DB2, which holds key information including shared IBM MQ object definitions.
3. IBM MQ object definitions.
4. A message offload or overflow option. SCM is considered overflow as it is part of the CF.
   a. SCM overflow
   b. SMDS offload
   c. SMDS + SCM
   d. DB2 offload - *not recommended* for IBM MQ versions that use CFLEVEL(5) and later

An application inside a queue-sharing group can put a message to a shared queue. If another queue manager in the queue-sharing group created the queue definition, the application does not need to go through the channel initiator, or mover, to place messages in this queue. Applications that are connected to any queue manager in the queue-sharing group can access the messages.

The main caveat about using shared queues is that space in the coupling facility, when compared to DASD, can be limited.

WebSphere Education

Queue-sharing groups and DB2

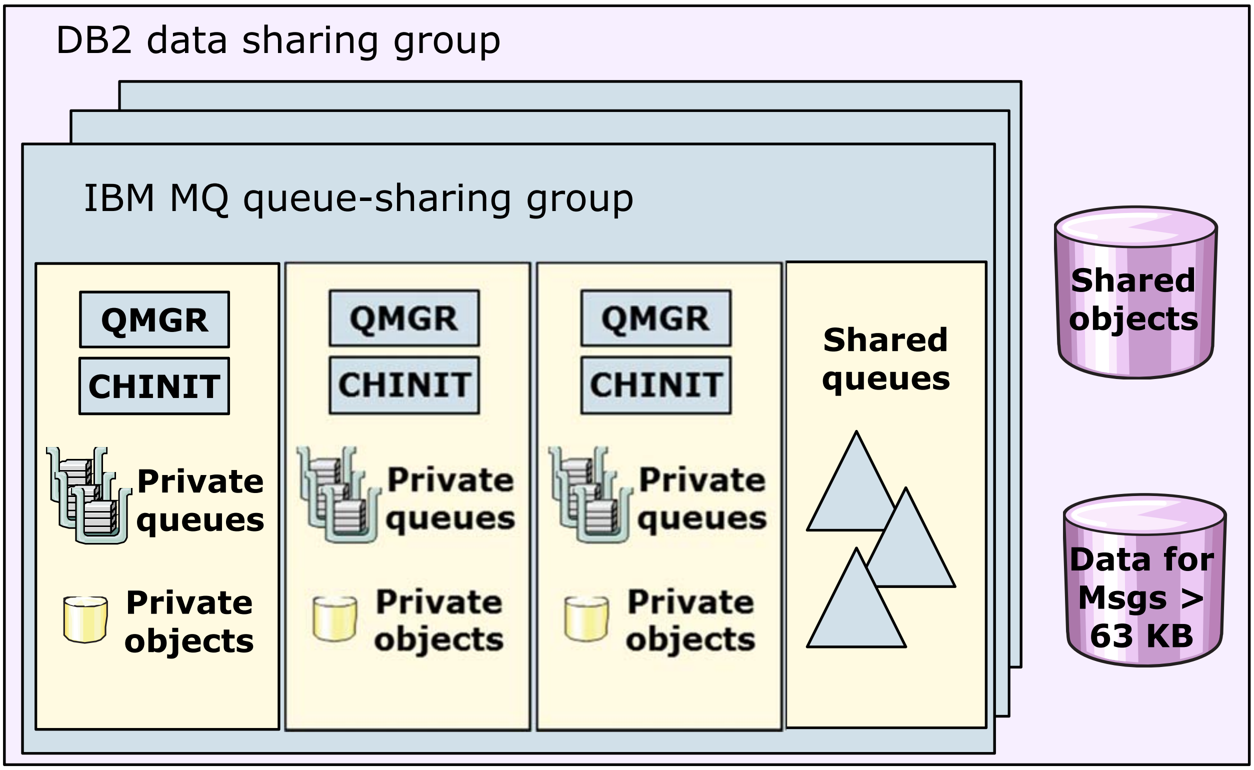


Figure 4-4. Queue-sharing groups and DB2 WM3121.1

Notes:

A queue-sharing group is added as an entry to a DB2 table. An entry is also added to DB2 to identify any queue manager that needs to be a member of the queue-sharing group. The queue manager continues to behave mostly like a stand-alone queue manager and keep its own object definitions; however, it also has access to shared queues.

Queue managers that belong to the same queue-sharing group must belong to the same data sharing group.

When shared queues are used:

- A queue manager can belong to one queue-sharing group.
- A queue manager holds its own “private” definitions.
- A queue manager can access definitions that are shared in the queue-sharing group, even if they were defined in another queue manager member of the same queue-sharing group.
- IBM MQ object definitions for queue-sharing groups are kept on DB2.
- Messages less than 63 KB can be stored in the coupling facility. Note the wording “can be stored”.

Messages over 63 KB can be stored in a shared message data set for IBM MQ V7.1 and later.

WebSphere Education

Coupling facility

- A coupling facility is a special type of LPAR available in z/OS
- Located outside the z/OS images in the sysplex that is usually configured on a separate power supply
- Managed by the coupling facility resource management (CFRM) policy
- Each message is referenced in the coupling facility's list structure that is dedicated to the specific queue-sharing group
- Each coupling facility:
  - Can hold list structures for more than one queue-sharing group
  - Up to 32 queue managers can concurrently connect to a coupling facility list structure
- A coupling facility list structure can contain 512 shared queues

> The real memory space that is used by the coupling facility is limited by the defined CF capacity. Keeping CF storage space available is a key objective when using shared queues.

Figure 4-5. Coupling facility WM3121.1

Notes:

You learn about "list structures" in the next slide.

Reminder

The amount of data that can be stored in the coupling facility for a single message is limited to 63 KB.

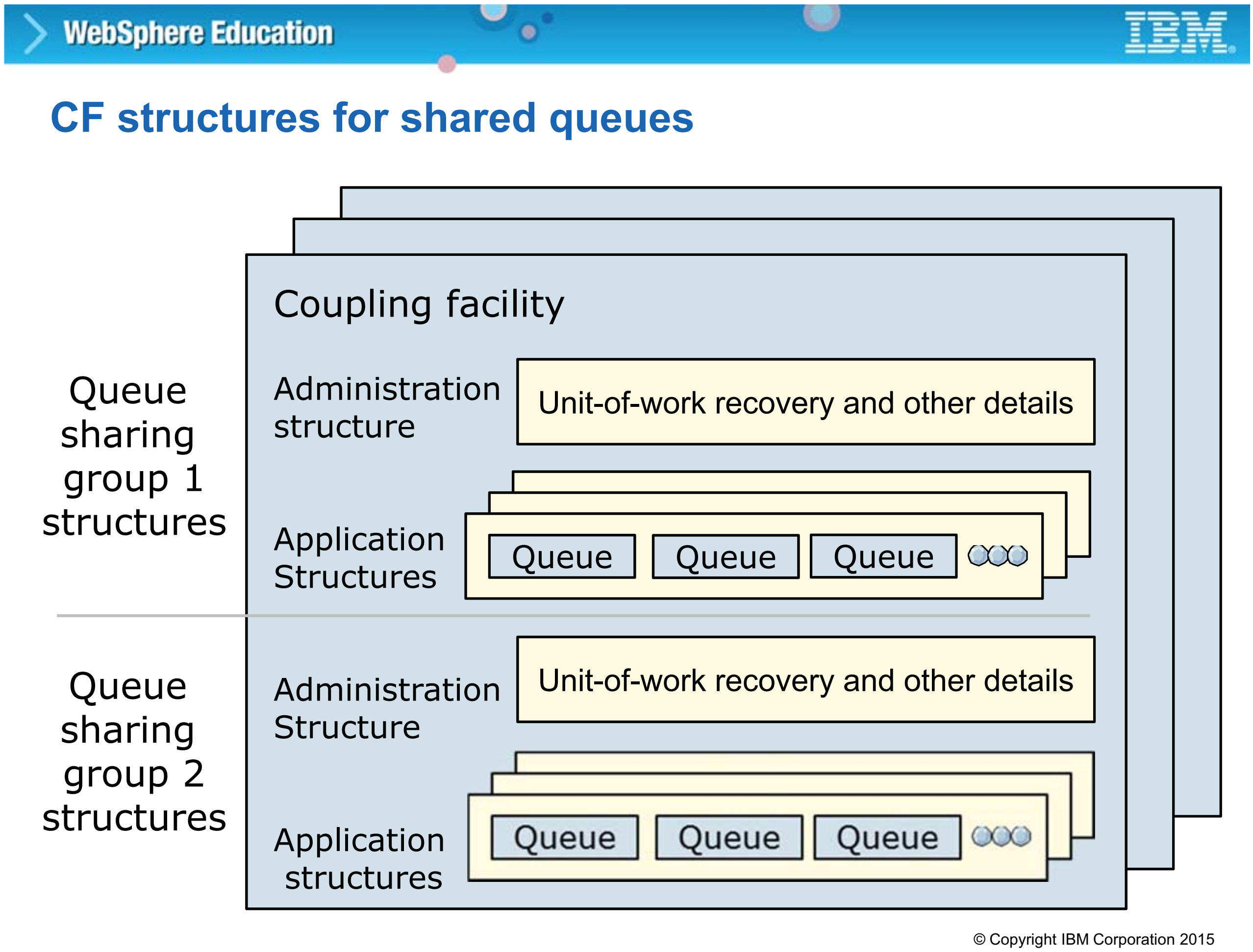


Figure 4-6. CF structures for shared queues WM3121.1

Notes:

IBM MQ data in a coupling facility is held in coupling facility structures. The term “list structures” is also used to refer to the IBM MQ structures.

The slide shows all structures in one coupling facility.

Coupling facilities are not exclusive to IBM MQ; other subsystems can have their own structures in the same coupling facility.

For each queue-sharing group, there is a maximum of 64 structures. These structures are:

- One administrative structure for IBM MQ needs, such as to manage unit-of-work recovery.
- One or more application structures to hold shared queues and the messages. Maximum is 63 application structures.
- If your applications require tracking for group units of recovery, one of the application structures is dedicated to the CSQSYSAPPL structure.

The structures are known to z/OS by its (up to) 16-character name, for example SG03APPL1 for the application structure, and SG03CSQ_ADMIN for the administrative structure.

The structures are known to IBM MQ by its (up to) 12-character name, for example APPL1 for the application structure, or CSQ_ADMIN for the administrative structure.

WebSphere Education

Partial coupling facility and IBM MQ application structures

```
/D XCF,STR,STRNAME=SG03APPL1
... ... ...
ACTIVE STRUCTURE
----------------
 ALLOCATION TIME: 06/19/2015 16:39:30
 CFNAME          : CF11
 COUPLING FACILITY:
 SIMDEV.IBM.EN.0000CFMVM
 PARTITION: 00   CPCID:
 STORAGE CONFIGURATION  ALLOCATED
  ACTUAL SIZE:                 49 M
  SPACE USAGE      IN-USE      TOTAL
   ENTRIES:          9034      17103
   ELEMENTS:        50756     102932
   EMCS:                5       7882
   LOCKS:                       1024
```

```
MQ03 DIS CFSTATUS(APPL1)
CSQM293I MQ03 CSQMDRTC 1
CSQM201I MQ03 CSQMDRTC  D
CFSTATUS(APPL1)
TYPE(SUMMARY)
CFTYPE(APPL)
STATUS(ACTIVE)
OFFLDUSE(SMDS)
SIZEMAX(50176)
SIZEUSED(52)
ENTSMAX(17103)
ENTSUSED(9034)
FAILTIME()
FAILDATE()
 END CFSTATUS DETAILS
```


Figure 4-7. Partial coupling facility and IBM MQ application structures WM3121.1

Notes:

If you look at the information that is generated from the coupling facility and IBM MQ commands for an application structure, you can see common fields of information in some of the information displayed. One example of this information is the ENTRIES data under SPACE USAGE in the XCF structure display; you see how it is related to the ENTSMAX and ENTSUSED fields in the IBM MQ CFSTATUS display.

Notice how the commands are entered. XCF, or coupling facility commands use the full structure name, including the queue-sharing group name, in this example SG03APPL1. Contrast how the IBM MQ command uses the shorter name, without the queue-sharing group name; in this example, APPL1.

You look at entries and elements in the capacity considerations topic. Notice how the ENTRIES numbers for the coupling facility application structure display match the IBM MQ coupling facility status attributes ENTSMAX and ENTSUSED.

One key piece of information in the DIS CFSTATUS command is in the SIZEUSED field; it shows the percentage that is used in the structure.

WebSphere Education

Partial coupling facility and IBM MQ administrative structures

```
/D XCF,STR,STRNAME=SG03CSQ_ADMIN
 ...  ...  ...
ACTIVE STRUCTURE
----------------
 ALLOCATION TIME: 06/18/2015 16:23:56
 CFNAME          : CF11
 COUPLING FACILITY:
 SIMDEV.IBM.EN.0000CFMVM
 PARTITION: 00   CPCID:
 STORAGE CONFIGURATION   ALLOCATED
  ACTUAL SIZE:                  49 M
  SPACE USAGE      IN-USE        TOTAL
   ENTRIES:              6         8804
   ELEMENTS:            96        17850
   LOCKS:                           256
```

```
/MQ03
 DIS CFSTATUS(CSQ_ADMIN)
CSQM293I MQ03 CSQMDRTC 1
CSQM201I MQ03 CSQMDRTC
 DCFSTATUS(CSQ_ADMIN)
TYPE(SUMMARY)
CFTYPE(ADMIN)
STATUS(ACTIVE)
OFFLDUSE()
SIZEMAX(20480)
SIZEUSED(1)
ENTSMAX(8804)
ENTSUSED(6)
FAILTIME()
FAILDATE()
 END CFSTATUS DETAILS
```


Figure 4-8. Partial coupling facility and IBM MQ administrative structures WM3121.1

Notes:

If you look at the information that is generated from the coupling facility and IBM MQ commands for an administrative structure, you can also see similarities in some of the coupling facility and IBM MQ data that is displayed.

The same syntax applies to the commands for the administrative structure. XCF, or coupling facility commands use the full structure name, including the queue-sharing group name, in this example SG03CSQ_ADMIN. Contrast how the IBM MQ command uses the shorter name, without the queue-sharing group name; in this example, CSQ_ADMIN.

You also see how the ENTRIES numbers for the coupling facility administrative structure display match the IBM MQ coupling facility status attributes ENTSMAX and ENTSUSED.

WebSphere Education IBM

The IBM MQ CFSTRUCT object

```
DEFINE CFSTRUCT (structure-name)
  -CFCONLOS:  ASQMGR,TERMINATE,TOLERATE
  -CFLEVEL:   Must be at least 5 for offload to SMDS
  -OFFLOAD:   SMDS (preferred) or DB2
 Offload rules.  Different values for SMDS and DB2.
  SMDS default values shown:
  - OFFLD1SZ(32K/64K) OFFLD1TH(70)
  - OFFLD2SZ(4K/64K)  OFFLD2TH(80)
  - OFFLD3SZ(0K/64K)  OFFLD3TH(90)
   SMDS database allocation attributes
   -DSGROUP, DSBLOCK, DSBUFFS, DSEXPAND(YES/NO)
   Recovery attributes
   -RECOVER(YES/NO): states if CF recovery is supported in
  application
   -RECAUTO: determines if auto recovery upon detecting
  certain error conditions
```


Figure 4-9. The IBM MQ CFSTRUCT object WM3121.1

Notes:

The IBM MQ CFSTRUCT object is used to define queue manager CF level capability, message offload environment, and backup and recovery parameters for a coupling facility structure.

Significant changes were made to queue-sharing groups over the different IBM MQ versions. The IBM MQ CFSTRUCT object was added with IBM MQ V5.3. The structure contains important attributes that determine the behavior of IBM MQ in different situations. You look at some of these attributes later in the unit. The definition for the CFSTRUCT object is included in SCSQPROC member CSQ4INSS.

The value of the CFLEVEL attribute in your CFSTRUCT definition determines what new capabilities can be enabled in your queue-sharing group. The capability to offload messages to a shared message data set is available to queue managers with a CFLEVEL of 5 or above in the CFSTRUCT object. The DSGROUP attribute of the CFSTRUCT object identifies the SMDS.

It is important to distinguish between the IBM MQ CFSTRUCT object and the z/OS coupling facility (CF) structure, as these terms might cause confusion.

Before IBM MQ V5.3, the IBM MQ CFSTRUCT existed, but was automatically defined. If you overlook defining the IBM MQ CFSTRUCT object before you place non-persistent messages in the queue, for IBM MQ V8 a structure is still automatically created, but it uses older version initial attributes. You learn more about this topic in the configuration unit. You cannot place persistent messages in this self-defined structure; you learn why later in this unit.

In the next few slides, you learn more about shared message data sets (SMDS). The IBM MQ CFSTRUCT object is where you tell IBM MQ the name of the SMDS associated with your queue manager.

You revisit the IBM MQ CFSTRUCT object in the configuration slides.

WebSphere Education IBM

Differences between non-shared and shared local queues

Storage	Non-shared local queues	Shared queues
Object definition storage	Stored on queue manager page set zero	Stored in DB2 OBJ_B_QUEUE table
Message storage	Page sets and buffer pools	Represented by an entry in a coupling facility structure. Stored, or "offloaded" according to message size: • Small messages: Usually kept in the coupling facility • Large messages: Message data can be stored in a shared message data set (SMDS) or DB2 • Messages over 63 K: Always offloaded • Can also be set up to overflow to flash express cards, also called storage class memory (SCM)

Figure 4-10. Differences between non-shared and shared local queues WM3121.1

Notes:

Basic differences between regular local queues and shared queues are shown on this slide.

Logging: For persistent messages, MQPUT and MQGET are logged to the log of the queue manager where the MQPUT or MQGET is done.

WebSphere Education

Architecture: How do shared queues compare with clusters

- Channel initiator usage
  - When messages are sent across shared queues, no channels are used
  - This scenario requires the queue managers to be on the same QSG
- Availability
  - If a clustered queue manager fails the message, is not accessible unless configured for failover with shared disks
  - Other queue managers in the queue-sharing group can access messages in a shared queue
- Capacity
  - Care must be taken to maintain space available in the CF
  - Storage in a coupling facility is more expensive than disk
  - CF storage can be mitigated by using the overflow and offload techniques
  - It is less expensive to use a local queue
- Workload balancing
  - A coupling facility does its workload balancing based on actual load
  - Cluster weighing distributes according to its configuration, regardless of load

Figure 4-11. Architecture: How do shared queues compare with clusters WM3121.1

Notes:

This slide *is not* intended as a one-to-one comparison between shared queues and clusters, but rather to emphasize the strength that each technology brings to the infrastructure.

Although clusters "help" with high availability, clusters provide better scalability without the concern to fill up the coupling facility structures. Clusters aim to ensure that a queue is always available in a queue manager. However, if a queue manager goes down cluster messages in the failed queue manager are "marooned" until the failed queue manager restarts or a backup queue manager that is configured with a failover solution can access the messages.

Shared queues are accessible by all queue managers in the queue-sharing group. If the queue manager that created or received a message fails, the messages are accessible by other queue managers in the sharing group, so the message is not marooned. However, with queue-sharing groups, you must ensure that IBM MQ and the coupling facility are configured in a way that can mitigate the possibility of having the coupling facility structures run out of space.

Clusters and shared queues can be used together to promote scalability and high availability, but each is better suited to a different role.

WebSphere Education

Coupling facility capabilities and the CFLEVEL

CFLEVEL	Capabilities
1-2	Non-persistent messages under 63 KB
3	Persistent and non-persistent under 63 KB
4	Persistent and non-persistent up to 100 MB
5	• Persistent and non-persistent up to 100 MB • Can be selectively offloaded to shared message data sets or DB2
19	• Persistent and non-persistent up to 100 MB • Ability to overflow to SCM

- How a queue manager uses different queue-sharing group resources is specified in the IBM MQ **`CFSTRUCT`** IBM MQ object
- IBM MQ **`CFSTRUCT`** objects are stored in DB2

Figure 4-12. Coupling facility capabilities and the CFLEVEL WM3121.1

Notes:

Ability to use SCM is not dependent on the version of IBM MQ but rather in the ability to support a `CF 19 CFLEVEL` coupling facility.

SCM is part of the CF structure, not a true “offload”. When using SCM, you are using part of the coupling facility as an overflow area.

Contrast the CFLEVELs to the next slide to see how the evolution in CF capabilities provides required support for queue-sharing group enhancements across the IBM MQ versions.

You learned that if you overlooked defining the IBM MQ CFSTRUCT object explicitly, the object would be defined dynamically the first time you tried to put non-persistent messages in the queue. If you notice the IBM MQ versions, it was not possible to use persistent messages before CFLEVEL(3).

When you see the next slide with IBM MQ versions, you can see a parallel between the CFLEVEL and the capabilities that are introduced in the different IBM MQ releases.

WebSphere Education

Large message support and CF structure use history

- Preservation of available space in the CF structure is always the key objective
- Need to offload messages to keep space available in the CF was enabled
- Faster offload or overflow and retrieval of messages incorporated

Release	V5.2	V5.3	V6.0	V7.1	V8.0
Maximum message size	63 K	63 K	100 Mb	100 Mb	100 Mb
Persistence	NP	Yes	Yes	Yes	Yes
Message store	CF	CF	CF, DB2	CF, DB2, SMDS offload	CF, DB2, SMDS offload, SCM overflow
CF Level		CF 3	CF4	CF5	CF19 for SCM

Figure 4-13. Large message support and CF structure use history WM3121.1

Notes:

This slide shows a historical perspective, from the IBM MQ view, at how mitigating space constraints and other attributes in the coupling facility hanged across IBM MQ releases.

You learned that if you overlooked defining the IBM MQ CFSTRUCT object explicitly, the object would be defined dynamically the first time you tried to put non-persistent messages in the queue. If you notice the IBM MQ versions, it was not possible to have persistent messages before V5.3.

IBM MQ V8 needs CFLEVEL(5) to use SMDS and CFLEVEL(19) to use SCM.

If you use 100 Mb messages, you must check that SYSTEM.QSG.TRANSMIT.QUEUE is large enough to accommodate the larger messages. You learn about the SYSTEM.QSG.TRANSMIT.QUEUE in the shared queue distributed unit.

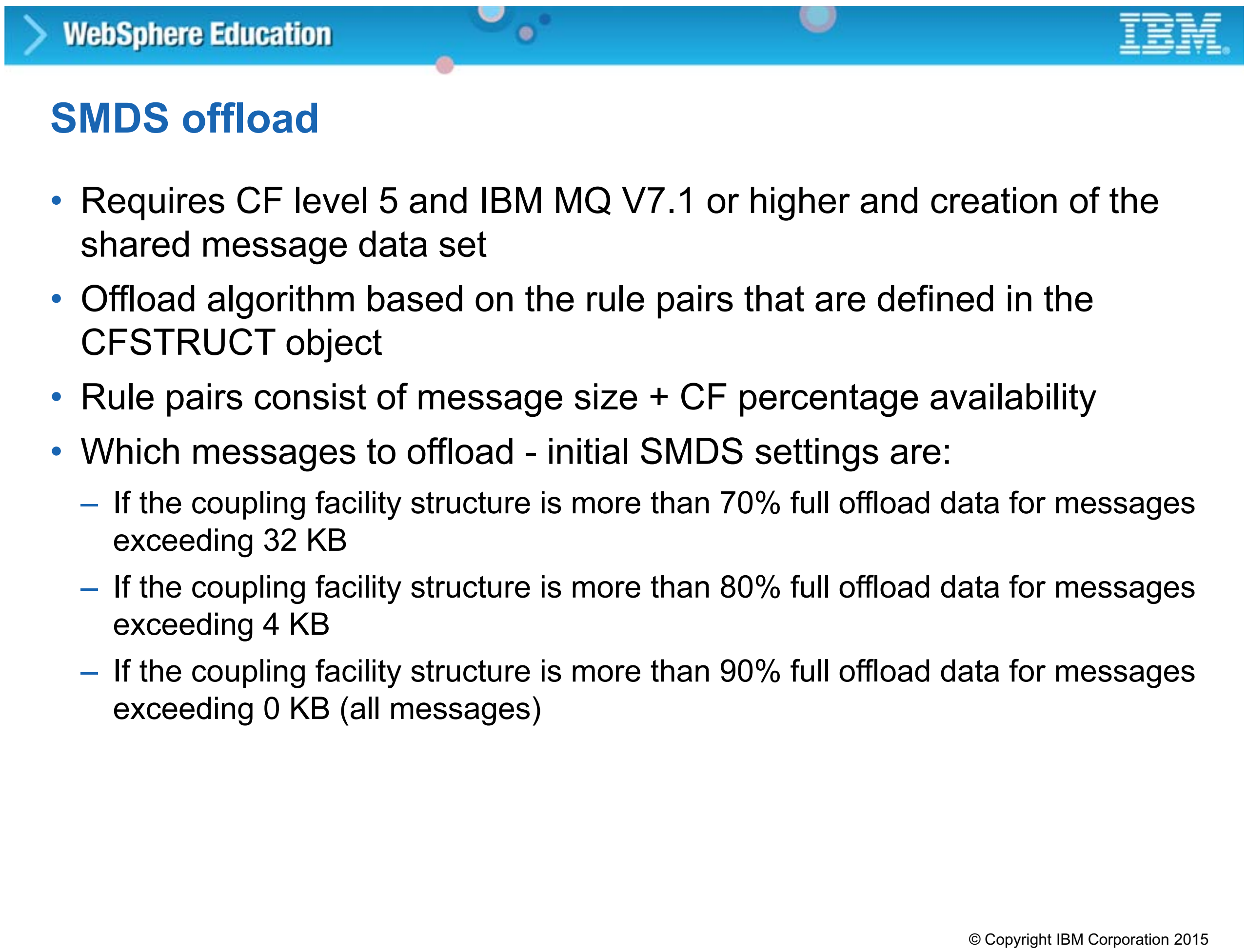


Figure 4-14. SMDS offload WM3121.1

Notes:

Each application structure is associated with a group of shared message data sets.

There is one shared message data set for each queue manager. The name of the SMDS data set and the SMDS offload rules are specified in the IBM MQ CFSTRUCT command.

A queue manager:

- Writes large messages and other messages according to the offload rules to its own data set; must have read/write access
- Can open and read the data sets for the other queue managers in the queue-sharing group; needs at least read-only access
- Needs read/write access to other queue managers data sets in the group for recovery

If a message with offloaded data needs to be deleted, it is passed back to the queue manager that wrote the message originally. When the message is deleted, the data set space can be released.

The overflow rules can be changed depending on the requirements of the application. The intended use of the initial default rules is:

- First rule is meant to save space for 32 K and larger messages by offloading them, even when ample space left, at 70% full. Performance impact is little. Figures are initial values.
- Second rule is intended to save more space by starting to offload messages 4 K and larger as the structure fills to 70%. The second rule is an intermediate step between the first and third rules. The performance impact is expected to be minor. Figures are initial values.
- The third rule is active when the structure is 90% full, at which point, if possible, all messages are offloaded. 90% is the initial full percentage value.

WebSphere Education

CF structure overflow with SCM

- Requires CF level 19
- Configured by the **`SCMALGORITHM`** and **`SCMMAXSIZE`** attributes in the CFRM policy
- Offload uses CF KEYPAIR1 algorithm based on message age and priority:
  - Pre-staging: When CF is 90% full, messages least likely to be accessed are moved to SCM
  - Pre-fetching: When 70% or more of the CF space is available, messages are brought back to CF

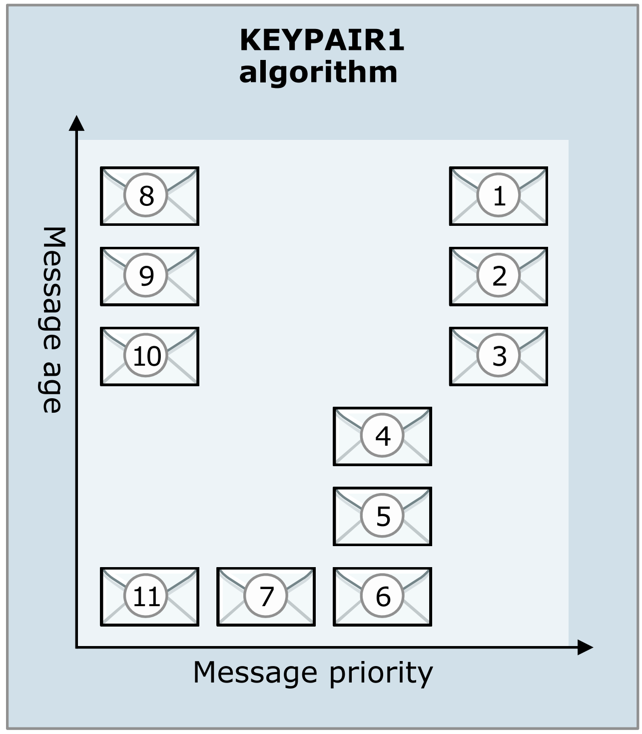


Figure 4-15. CF structure overflow with SCM WM3121.1

Notes:

The CFRM policy controls the SCM overflow. As of the time this course is written, KEYPAIR1 is the only overflow algorithm for CF19 level.

You learn more about SMMAXSIZE in the capacity consideration slides.

As noted in this slide:

- When the SCM algorithm detects a CF 90% full algorithm, it starts a process termed “pre-staging” to move messages less likely to be accessed to SCM. Messages with lower priorities that were placed in the queue more recently are pre-staged first.
- When the SCM algorithm detects the CF back to 70% or more available, it starts the “pre-fetch” process to return messages to the coupling facility. Older messages with higher priorities are pre-fetched first.

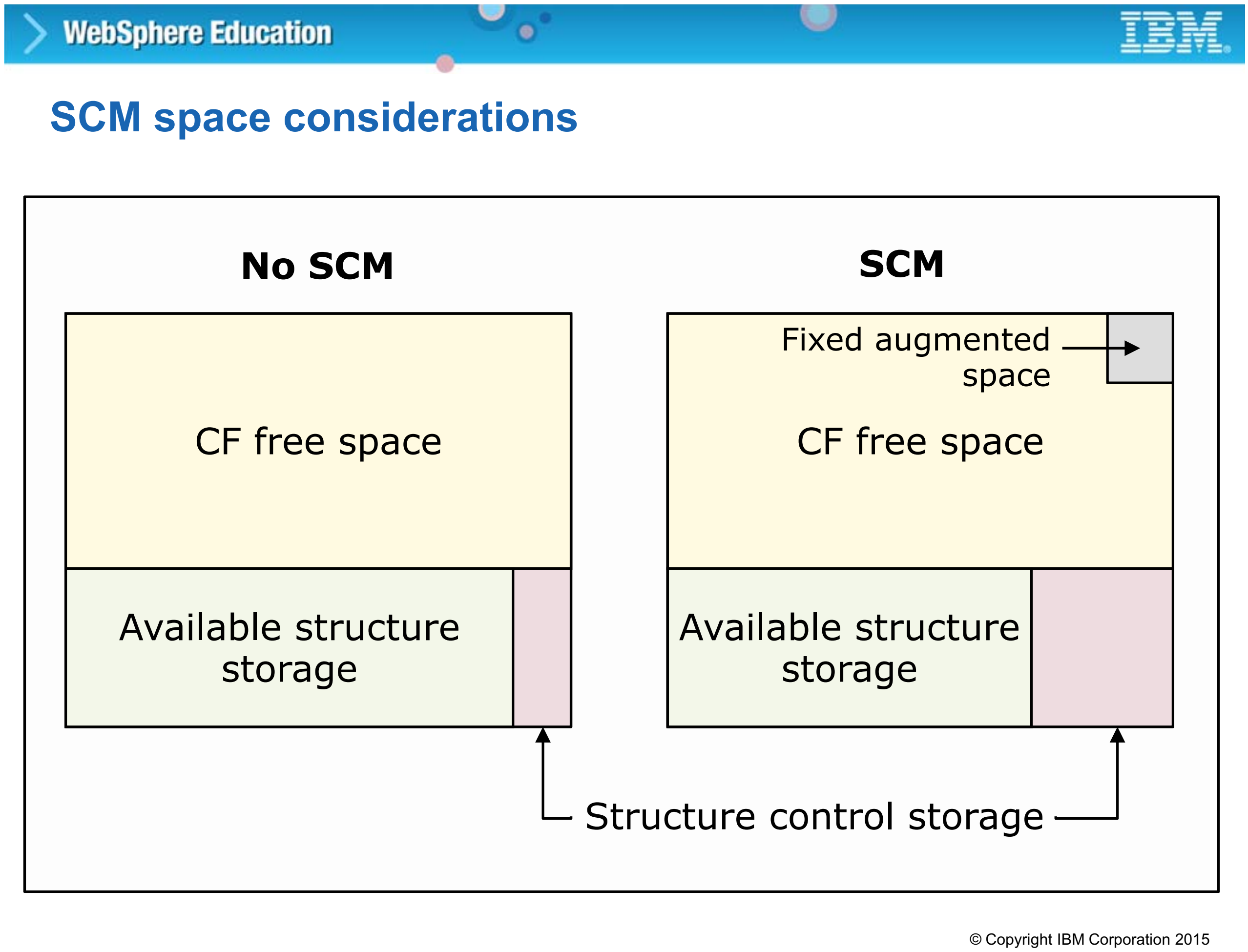


Figure 4-16. SCM space considerations WM3121.1

Notes:

Various data structures in the coupling facility track the use of SCM. These data structures, which are known as *augmented space*, exist in the real storage that is allocated to the coupling facility, and can reduce the amount of storage available to other structures.

Storage is allocated to SCM in the following ways:

- A structure that is configured with SCM reserves a small space of real space named *fixed augmented space*. This space is allocated regardless if the structure uses SCM.
- More *dynamic augmented space* is allocated from the spare real storage as data from a structure is stored in SCM.
- Dynamic augmented space is returned to the coupling facility when the data is removed from SCM.
- In addition to augmented storage, the amount of control storage increases when a structure is configured to use SCM, so there is less space for data in structures.

You see two indications of dynamic augmented space use in the forthcoming slides that show tested scenarios.

WebSphere Education IBM

What is the best option to mitigate CF space constraints?

- Increase the size of the CF structure?
- Use SMDS to offload messages?
- Extend the CF structure with SCM?
- Offload to DB2?
- Combination of SMDS offload and SCM overflow?

Always benchmark!

	DB2 offload	SMDS offload	SCM overflow	Bigger CF
Cost	Not considered	Less than SCM *	Less than CF	Most expensive
Performance	Very slow	Slower than SCM	High perf. option	Best
Tradeoff	No longer advised for offload	Offload performance IBM MQ version must be MQV7.1 >	Requires CF 19	Real storage availability?

No "one size fits all" answer. While in some cases SCM hardware costs might be higher, application requirements and the entire solution cost must be analyzed.

Figure 4-17. What is the best option to mitigate CF space constraints? WM3121.1

Notes:

As with normal, non-shared IBM MQ configuration, a successful implementation is based on adequate research of the requirements of the work that is processed in the environment.

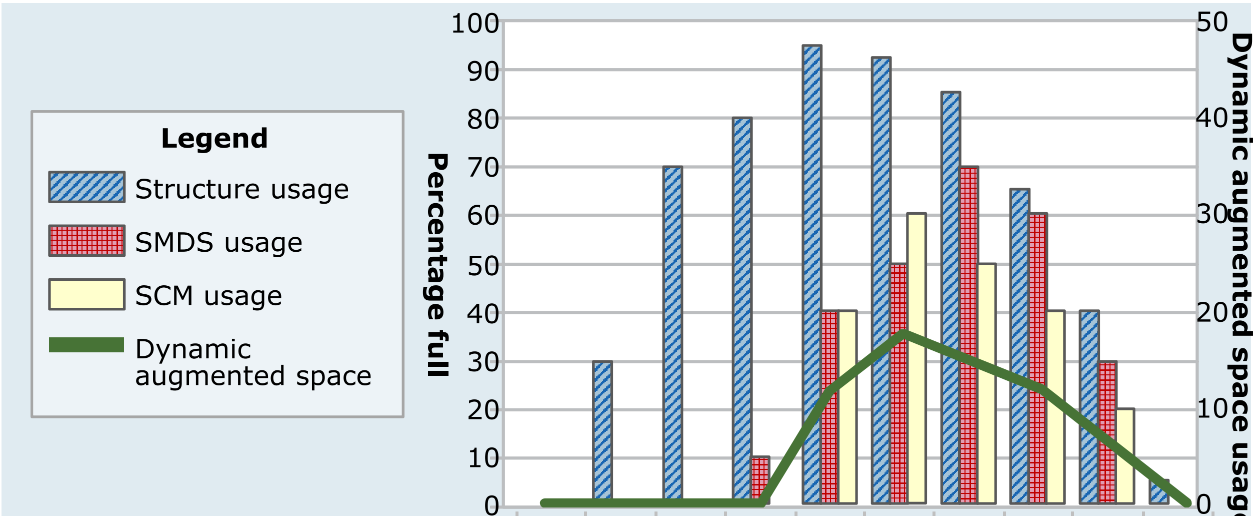
SMDS and SCM work together according to the corresponding rules and algorithms of each technology.

In the next two slides you review two scenarios and see why each technology was selected:

- Requirement for the first scenario is the ability to tolerate a two-hour outage of the getting application. This scenario takes advantage of both SMDS offload and SCM overflow.
- In the second scenario, the ability to handle volume spikes is the requirement. This scenario uses SCM.

WebSphere Education

SMDS+SCM: Tolerate outage of getting application



Time	0	1	2	3	4	5	6	7	8	9
Putting application active	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Getting application active	✓	X	X	X	X	✓	✓	✓	✓	✓
Offload rule 1 active	X	X	✓	✓	✓	✓	X	X	X	X
Offload rule 2 active	X	X	X	✓	✓	✓	✓	X	X	X
Offload rule 3 active	X	X	X	X	✓	✓	✓	X	X	X
Pre-stage active	X	X	X	X	✓	✓	X	X	X	X
Pre-fetch active	X	X	X	X	X	X	X	✓	✓	✓

Figure 4-18. SMDS+SCM: Tolerate outage of getting application WM3121.1

Notes:

The requirement for this scenario is the ability to handle a two-hour outage of the application that gets these messages. Green checks indicate an active process and red Xs a stopped process.

The measure for dynamic storage on the right margin refers to the percentage of 4 GB of pre-allocated SCM storage used.

- IBM MQ CFTRUCT offload rules control if messages are written to SMDS. In this case, as the application structure starts to fill, the first two offload rules activated with larger, then smaller messages.
- Even when a message is stored in SMDS, a pointer or reference to that message is still kept in the coupling facility structure.
- When the 90% full threshold is reached for the structure, the queue manager tries to place as many messages references as possible in the application structure.
- Also, at the 90% threshold, the SCM algorithm becomes active. It causes messages to move to SCM storage, keeping messages most likely to be retrieved next, in the faster “real” storage.

As the “getting application” resumes, you can see discontinuation of the SMDS offload rules and the SCM pre-fetch becoming active.

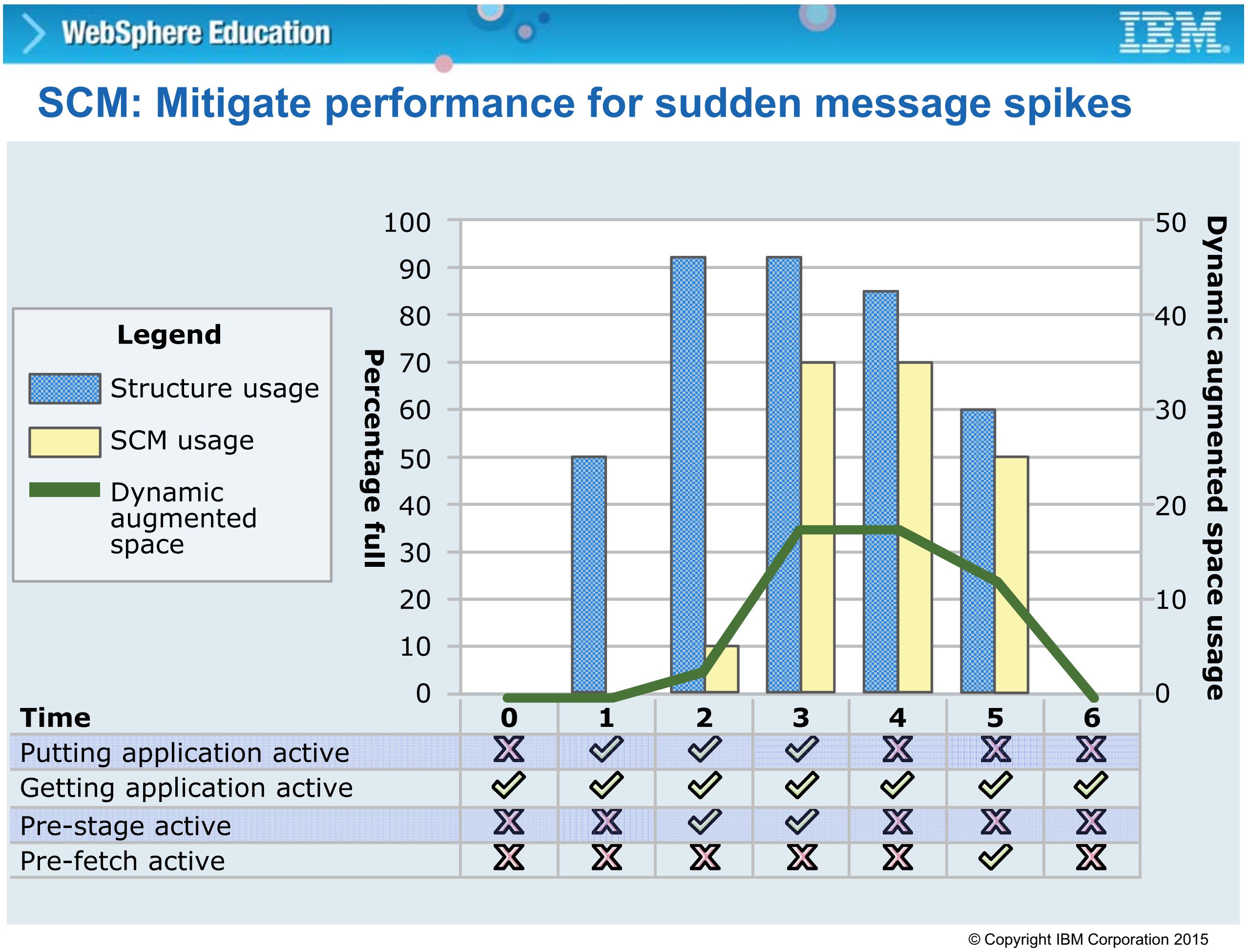


Figure 4-19. SCM: Mitigate performance for sudden message spikes WM3121.1

Notes:

The requirement for this scenario is to handle sudden spikes in incoming messages. An example of this scenario might be a holiday sale, or special event.

The measure for dynamic storage on the right margin refers to the percentage of 4 GB of pre-allocated SCM storage used.

- Except for messages larger than 63 KB, the SMDS offload rules were disabled by specifying 64 K in two of the “offload size” attributes. Although not represented in the figure, these larger messages are always offloaded.
- When the 90% threshold is reached, the SCM pre-staging algorithm starts moving newer messages to SCM, allowing the faster, “real” application storage for older messages that are expected to be retrieved next.
- Although the putting application continues to put messages, an MQRC_STORAGE_MEDIUM_FULL condition is mitigated because messages that do not fit in the application structure are being placed in SCM.
- The putting application has no more messages. After the structure falls below the 90% full threshold, the pre-staging algorithm stops, and the pre-fetch algorithm starts moving messages from SCM back to the structure.

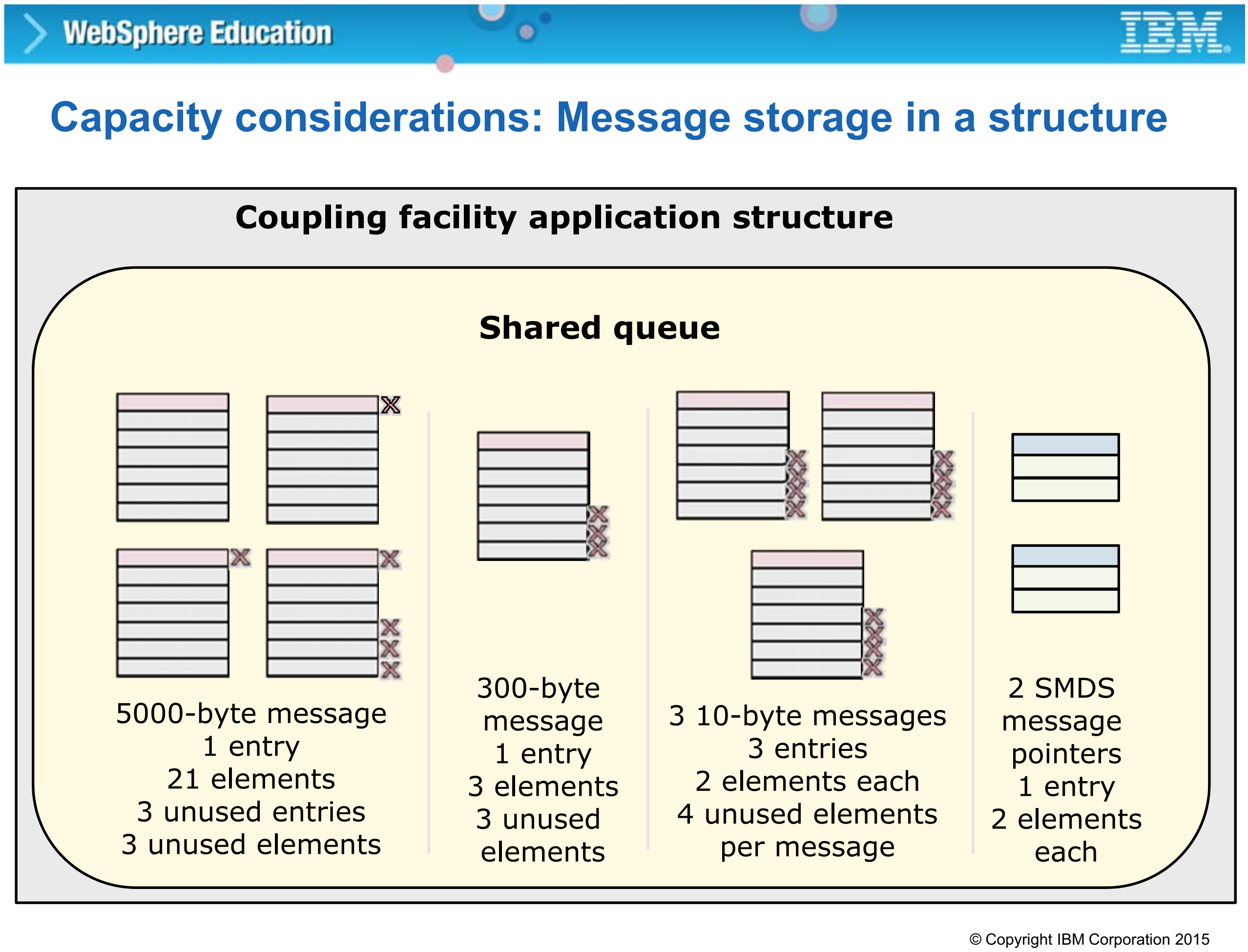


Figure 4-20. Capacity considerations: Message storage in a structure WM3121.1

Notes:

There is substantial interaction between XCF and IBM MQ. It is important to understand how messages are handled in the coupling facility to adequately plan configuration details.

Messages in a structure are arranged in a series of an ENTRY and ELEMENTS.

Important

The structure is considered full when it runs out of ELEMENTs or ENTRYs.

The ENTRY and ELEMENTs maintain a ratio that is initially set by IBM MQ to 1:6; that is 1 ENTRY to 6 ELEMENTs.

The figure illustrates how messages are stored according to this ratio. It also shows wasted ENTRY or ELEMENTs, indicated by an ‘x’ on the right, depending on the size of the message.

- Each message must have one ENTRY, and enough ELEMENTs to hold the message and headers.

- For purposes of this explanation, you add approximately 372 bytes to the size of the message. Next, you determine how many ELEMENTs are required to store a message.
- Each ELEMENT is 256 bytes, so you calculate (Message length + 372)/256 to derive the number of elements that a message requires.
- Messages that are offloaded to SMDS keep one ENTRY and two ELEMENTs in the coupling facility.

When you look at the slide, you can see how different messages use the coupling facility ENTRY and ELEMENTs.

It is possible for z/OS to alter the 1:6 ratio to fit more messages in a structure. In the coming slides, you see some of the attributes to consider for this process.

WebSphere Education

CFRM list structure attributes (1 of 2)

```
D XCF,STR,STRNAME=SG03APPL1
… … … … …
STRNAME: SG03APPL1
 STATUS: ALLOCATED
 EVENT MANAGEMENT: POLICY-BASED
 TYPE: SERIALIZED LIST
 POLICY INFORMATION:
  POLICY SIZE     : 200000 K
  POLICY INITSIZE: 50000 K
  POLICY MINSIZE : 50000 K
  FULLTHRESHOLD  : 85
  ALLOWAUTOALT   : YES
  REBUILD PERCENT: N/A
  DUPLEX         : DISABLED
  ALLOWREALLOCATE: YES
  PREFERENCE LIST: CF11     CF12
  ENFORCEORDER   : NO
  EXCLUSION LIST IS EMPTY
```

Policy information

- SIZE
- INITSIZE
- MINSIZE
- FULLTHRESHOLD
- ALLOWAUTOALT
- DUPLEX
- PREFERENCE LIST
- ENFORCEORDER

If using SCM

- SCMMAXALLOC

Consult:

– *Support pac MP16: Capacity Planning and Tuning for IBM MQ for z/OS*

– CFSizer tool

Figure 4-21. CFRM list structure attributes (1 of 2) WM3121.1

Notes:

The structure display is shown across two slides. There are some attributes in the CFRM policy to keep in mind when configuring the structure. These attributes are presented as a starting point.

There are numerous variables that interact across some of these attributes. It is imperative to consult the z/OS information in IBM Knowledge Center to understand all interdependencies:

- ALLOWAUTOALT. If this attribute is set to (YES) in the CFRM policy, it allows the system, on a need basis, to automatically *alter the size and ratio attributes of the structure.*
- FULLTHRESHOLD is an attribute that controls the frequency that the structure is monitored for space. The initial value is 80%. If a structure has the initial monitoring value, when the structure approaches 80% full, if ALLOWAUTOALT is YES, then the structure would be eligible for the automatic alteration of size and ration attributes.
- INITSIZE is the size that you initially want the structure to be when started.
- SIZE denotes the maximum size of the structure. It is suggested that SIZE should be 1.5 to two times the amount of INITSIZE.

- MINSIZE is the smallest size that the structure can be. It is important that MINSIZE is not left at 0, otherwise the ALLOWAUTOALT(YES) process might result in excessive space that is removed from the structure.
- SCMMAXSIZE. If using SCM, it is important not to over commit SCM. SCMMAXSIZE should not be allocated in a way that it exceeds the available coupling facility space.

Note

Setting ALLOWAUTOALT(YES) helps mitigate space shortages since it adjusts the 1:6 ratio when the system detects a different optimal ratio.

If ALLOWAUTOALT(YES) is used with SCM, and the pre-fetching algorithm starts, the adjustment of entry to ratio is stopped. If ALLOWAUTOALT(YES) stops before it reaches an optimal situation, it might result in an inefficient entry to element ratio reached.

Be aware of augmented storage use when using SCM; always benchmark.

For more information, see:

- *SupportPac MP16: Capacity Planning and Tuning for IBM MQ for z/OS*
- z/OS documentation in the IBM Knowledge Center: *z/OS 2.1.0 > z/OS MVS > z/OS MVS Setting Up a Sysplex > Managing coupling facility resources > Planning a coupling facility policy > Identifying the coupling facility structures > Requesting structure size > How MVS initially allocates the structure*
- CFSizer tool at `http://www.ibm.com/systems/support/z/cfsizer/mqseries/`

CFRM list structure attributes (2 of 2)

```
ACTIVE STRUCTURE
----------------
 ALLOCATION TIME: 06/19/2015 16:39:30
 CFNAME         : CF11
 COUPLING FACILITY: SIMDEV.IBM.EN.0000CFMVMB11
                    PARTITION: 00   CPCID: 00
 STORAGE CONFIGURATION     ALLOCATED          MAXIMUM      %
  ACTUAL SIZE:                   49 M           196 M   25

  SPACE USAGE      IN-USE        TOTAL    %
   ENTRIES:          9034        17103   52
   ELEMENTS:        50756       102932   49
   EMCS:                5         7882    0
   LOCKS:                         1024

  102932/17103 = ~ 6.018359
```


Figure 4-22. CFRM list structure attributes (2 of 2) WM3121.1

Notes:

This slide shows the second half of the structure display output. You can see the ENTRIES and ELEMENTS. If you calculate the entry to element ratio, you see the initial ratio set by IBM MQ, 1 ENTRY per 6 ELEMENTs.

Depending on application requirements, you might need to change the 1:6 ratio. This ratio can change if ALLOWAUTOALT(YES) is set.

Information

The displays that follow compare storage usage in the two scenarios that were presented earlier. These comparisons show how the entry to element ratio can change when manipulating CFRM to allow automatic alteration of structure storage. It also shows manipulation of the SMDS offload rules to force all messages to be offloaded when the structure reached 70% full.

Observe the changes in ENTRYs and ELEMENTs after SCM is configured with the adjusted CFRM and SMDS offload rules.

SCM storage for the example display in these notes was configured with 4 GB in the SCMMAXSIZE parameter.

Structure display (D, XCF, ...) of sample environment before use of SCM. EMCS is event monitoring controls:

```
SPACE USAGE IN-USE TOTAL %
ENTRIES: 205041 205041 100
ELEMENTS: 2947918 3224661 91
EMCS: 2 386218 0
LOCKS: 1024
```

Structure display after enabling SCM. You see the SCM labels added to the display:

```
STORAGE CONFIGURATION ALLOCATED MAXIMUM %
   ACTUAL SIZE: 1024 M 1024 M 100
   AUGMENTED SPACE: 3 M 142 M 2
   STORAGE-CLASS MEMORY: 88 M 4096 M 2
      ENTRIES: 120120 1089536 11
      ELEMENTS: 240240 15664556 1
   SPACE USAGE IN-USE TOTAL %
      ENTRIES: 84921 219439 38
      ELEMENTS: 2707678 3149050 85
      EMCS: 2 282044 0
      LOCKS: 1024
```

If you compare the environment before and after SCM, there were some changes in the storage distribution:

- Gained 14,398 entries
- Lost 75,611 elements
- Lost 104,174 EMCS

Source of example SCM configuration and scenarios: *IBM MQ V8 Features and Enhancements Redbook*.

WebSphere Education

Administrative storage in a coupling facility: CSQ_ADMIN

- Factors that affect the administrative structure
  - Number of queue managers in the queue-sharing group
  - Long running UOW and number of messages in the UOW
- Avoid UOWs with large number of messages
- A queue manager in a queue-sharing group is not allowed to start unless a minimum of 1000 entries per queue manager are available
- Consult *Performance report – SupportPac MP1H WebSphere MQ for z/OS version 7.1.0*

```
MQ03 DIS CFSTATUS(CSQ_ADMIN)
… … …
SIZEMAX(20480)
SIZEUSED(12)
ENTSMAX(8804)
ENTSUSED(146)
… … …
```


Figure 4-23. Administrative storage in a coupling facility: CSQ_ADMIN WM3121.1

Notes:

Administrative structures are treated different than application structures.

- With application structures, you consider the volume and size of messages.
- With the administrative structure, you are concerned in applications that long running units of work (UOWs), and with the number of queue managers in the queue-sharing group.

The SIZEUSED denotes the percentage of the structure used.

ENTSMAX shows the maximum number of entries possible for the structure, less any structures that are held in SCM.

ENTSUSED shows the maximum number of entries in use for the structure, less any structures that are held in SCM.

Note

When a new capability is made available for IBM MQ, more performance information might be published in a SupportPac corresponding to the IBM MQ version for which the capability was released. Since shared queues had significant functions added for IBM MQ v7.1, SupportPac MP1H is still applicable to SMDS, although it refers an older IBM MQ version than 8.0.

Always cross-reference the SupportPac material with IBM Knowledge Center, for details on any new or superseded information.

Queue-sharing group considerations: Support pack MP16

The Capacity Planning and Tuning for IBM MQ for z/OS support pack MP16 contains critical details that must be reviewed.
Examples of the topics in MP16 are:

- How many CF structures should be defined?
- What attributes in the CFRM policy?
  - FULLTHRESHOLD
  - ALLOWAUTOALT
  - SIZE, INITSIZE, MINSIZE
  - SCMMAXSIZE
  - DUPLEX, PREFERENCE LIST, ENFORCEORDER
- What impact does the size of the administrative CSQ_ADMIN structure have on the queue managers?
- When the first message for an empty shared queue arrives, is there any extra work incurred by the queue manager?

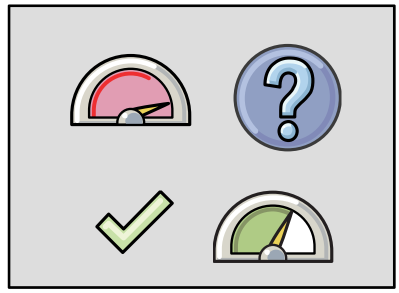


Figure 4-24. Queue-sharing group considerations: Support pack MP16 WM3121.1

Notes:

This slide reiterates the need to review the referenced documentation for details that might apply to your specific environment.

There are many variables when sizing and configuring the coupling facility and IBM MQ for queue-sharing groups. You looked at some of the initial attributes to consider as a starting point. For more information, see the SupportPac MP16.

WebSphere Education IBM

SMDS space considerations

- How message size is calculated
  - A message is stored as one or more message blocks
  - Each message includes standard headers – usually 352 bytes
  - Each block is contained in a range of 4 K pages on the data set plus a header
  - To determine the space to allocate per large message, in bytes, use the size of the message, plus header, and round up to the next 4 K
- After estimating message size, multiply by the anticipated volume, including a conservative margin for unanticipated messages
- Expansion:
  - Set DSEXPAND YES or NO in IBM MQ CFSTRUCT definition
  - Modify by using ALTER SMDS command
  - Expansion is automatically triggered when data set is 90% full
- Consult *Performance report – SupportPac MP1H WebSphere MQ for z/OS version 7.1.0*

Figure 4-25. SMDS space considerations WM3121.1

Notes:

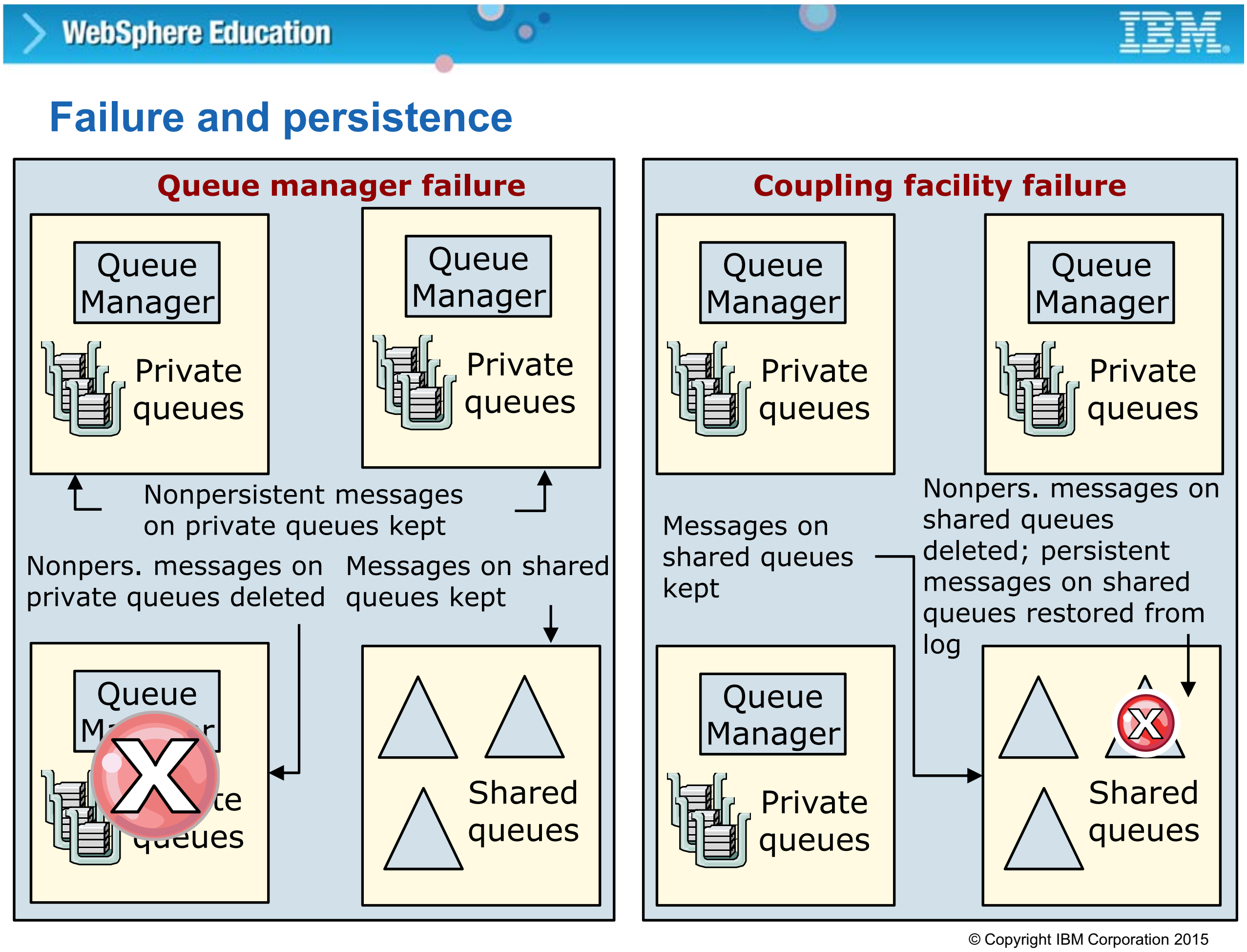


Figure 4-26. Failure and persistence WM3121.1

Notes:

When a queue-sharing group *queue manager* is stopped or fails:

- Non-persistent messages in non-shared queues (private to the failing queue manager) are lost or deleted
- Even if all queue managers in the queue-sharing group fail, messages on shared queues are not lost

If one or more coupling facilities fail:

- Non-persistent messages on queues in failing CF structure are lost
- Messages on queues in other non-failing structures are not lost
- Persistent messages on queues in failing CF structure are restored from backup and log information on the logs
- The “restoring” queue manager accesses the logs of all queue managers in the queue-sharing group

If the administration structure for a queue-sharing group fails, all queue managers in the queue-sharing group fail.

Active queue managers in queue-sharing groups at IBM MQ V7.0.1 and later detect a failure and initiate administration structure rebuild.

WebSphere Education

IBM MQ V7.1 and later loss of CF connectivity tolerance

- Queue managers tolerate loss of connectivity to administrative structure without ending
  - If QMGR CFCONLOS attribute is set to TOLERATE, all queue managers in the queue-sharing group are at IBM MQ V7.1 or later
  - Lack of available CF might cause some shared queue operations to remain
- Queue managers tolerate loss of connectivity to application structure if QMGR CFCONLOS is set to TOLERATE and they are at CFLEVEL 5
  - All queue managers that lose connectivity, disconnect
  - If loss of connectivity is total, structure might need to be recovered, and non-persistent messages are lost
- Applicable attributes
  - QMGR CFCONLOS(TERMINATE|TOLERATE)
  - CFSTRUCT CFCONLOS(TERMINATE|TOLERATE|ASKQMGR)
  - CFSTRUCT RECAUTO(YES|NO)

Figure 4-27. IBM MQ V7.1 and later loss of CF connectivity tolerance WM3121.1

Notes:

WebSphere Education

IBM MQ and coupling facility connectivity in the system log

```
... ... ... ... ...
IXL014I IXLCONN REQUEST FOR STRUCTURE SG03CSQ_ADMIN  892
WAS SUCCESSFUL.  JOBNAME: MQ03MSTR ASID: 0057
CONNECTOR NAME: CSQESG03MQ0301 CFNAME: CF11
CSQE005I MQ03 CSQECONN Structure CSQ_ADMIN connected  893
... ... ... ... ...
CSQE151I MQ03 CSQECKLC Loss of admin structure  918
connectivity toleration enabled
 ... ... ... ... ...
CSQE140I MQ03 CSQEENFR Started listening for ENF 35  941
events for structure APPL1
IXL014I IXLCONN REQUEST FOR STRUCTURE SG03APPL1  942
WAS SUCCESSFUL.  JOBNAME: MQ03MSTR ASID: 0057
CONNECTOR NAME: CSQESG03MQ0301 CFNAME: CF11
CSQE141I MQ03 CSQEENFR Stopped listening for ENF 35  943
events for structure APPL1
CSQE005I MQ03 CSQECONN Structure APPL1 connected as  944
CSQESG03MQ0301, version=CF1F60B1471FD34C 00010006
```


Figure 4-28. IBM MQ and coupling facility connectivity in the system log WM3121.1

Notes:

This slide sows some of the messages that you see in the system log regarding connectivity between IBM MQ and the coupling facility.

- First, you see the connection to the administrative structure, in this case **SG03CSQ_ADMIN**.
- Next, you see message **CSQE151I**, indicating connectivity toleration, or the CFCONLOS attribute setting in the IBM MQ CFSTRUCT object.
- Last in this display, you see the connection for the application structure, in this case **SG03APPL1**.

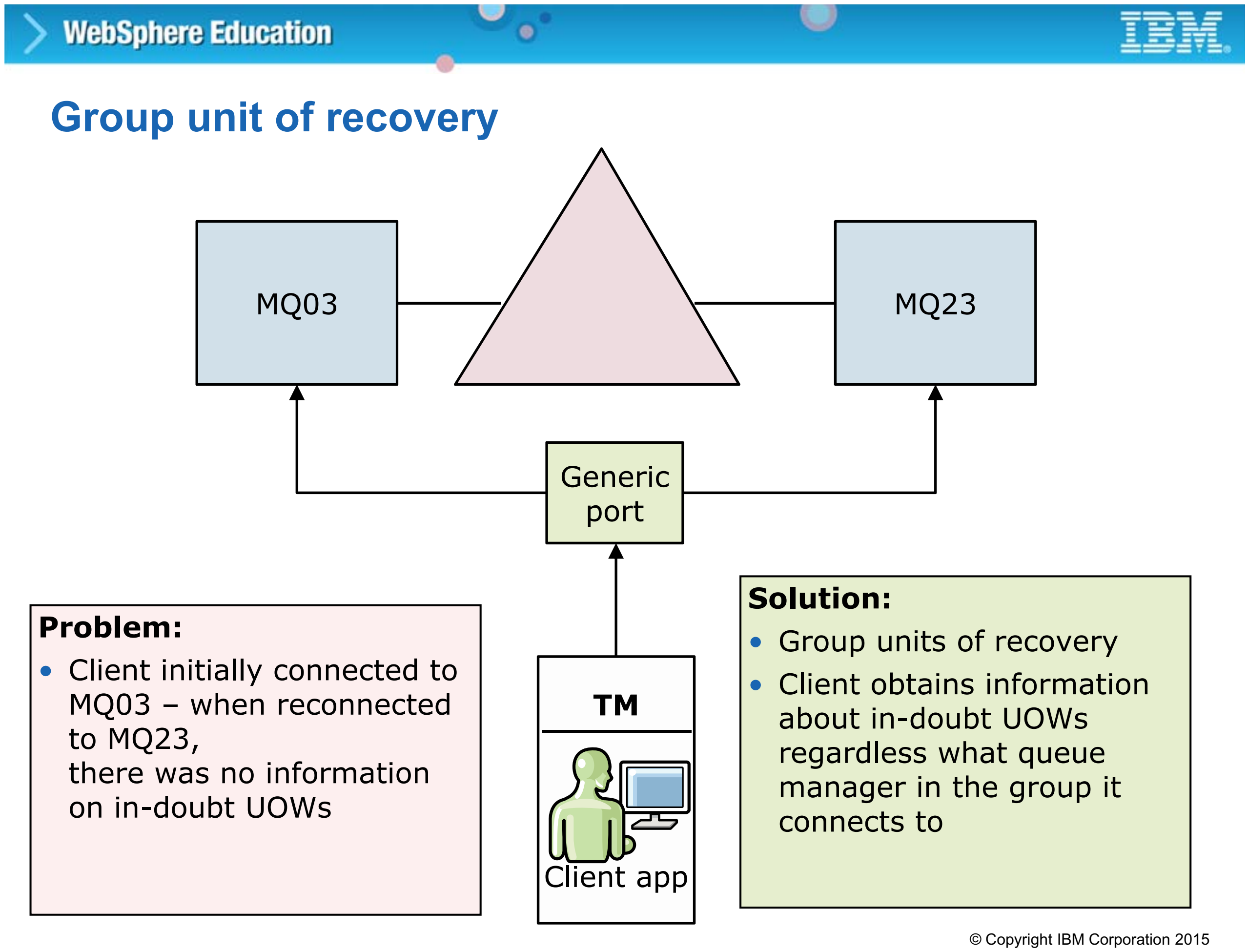


Figure 4-29. Group unit of recovery WM3121.1

Notes:

To implement GROUPUR:

- If not already present, define an IBM MQ CFSTRUCT object for CSQSYSAPPL
- SYSTEM.QSG.UR.RESOLUTION.QUEUE must be defined to use the CSQSYSAPPL structure
- Alter the queue manager GROUPUR to ENABLED for selected queue managers
- OPMODE parameter must not inhibit use of GROUPUR – queue manager must be at IBM MQ V7.0.1 or later with OPMODE set to enable new functions

Applications use the queue-sharing group name instead of the queue manager name in the xa_open string.

WebSphere Education

MQSC commands to manage shared queue environment

Display group information

```
DISPLAY GROUP
```

Manage CFSTRUCT options

```
DISPLAY CFSTRUCT
DEFINE CFSTRUCT
ALTER CFSTRUCT
DELETE CFSTRUCT
```

Check CFSTRUCT status

```
DISPLAY CFSTATUS
```

Manage backup and recover of shared messages

```
BACKUP CFSTRUCT
RECOVER CFSTRUCT
```

Manage SMDS data set options

```
DISPLAY SMDS
ALTER SMDS
```

Manage status and availability of QSG data sets

```
DISPLAY CFSTATUS TYPE(SMDS)
RESET SMDS
```

Review SMDS data set details:

```
DISPLAY USAGE TYPE(SMDS)
```

Manage status and availability of data set connections:

```
DISPLAY SMDSCONN
START SMDSCONN
STOP SMDSCONN
```


Figure 4-30. MQSC commands to manage shared queue environment WM3121.1

Notes:

The MQSC commands provide options to manage, display, and when possible manipulate details with the coupling facility structures and the shared message data sets.

- The DISPLAY GROUP command shows the queue managers in a queue-sharing group and the status of the queue managers. It also shows any obsolete DB2 messages, if any.
- The CFSTRUCT commands provide a way to view and alter the structure. To alter a structure, it must not contain messages or be actively allocated to the queue-sharing group.
- DISPLAY CFSTATUS can be used with the structure, and with the SMDS by using the TYPE(SMDS) attribute. The SIZEUSED attribute shows the percentage of space that is used in the structure, and status information, such as an ACTIVE status for the structure.
- The SMDS option of DISPLAY CFSTATUS indicates whether SMDS files are active and enabled for each queue manager.
- Structures should be backed up often, not only to preserve messages but also to decrease the amount of time that it would take to recover a sizable backup. The BACKUP CFSTRUCT command, if entered with an asterisk as the structure name, initiates a backup of all known

recoverable structures. RECOVER CFSTRUCT as implied, recovers a structure from a prior backup.

- The DIS SMDSCONN command provides access information about an SMDS data set, such as whether the data set is open, and in which mode, such as read or update. This command might specify a generic SMDS value, but requires a structure name.
- Contrast DIS SMDS with DIS SMDSCONN. DIS SMDS provides information about the way the SMDS was defined as far as the number of buffers and expansion option. You must use the "ALL" option to obtain all information. If the command returns "DEFAULT" in the buffers and recovery value, it indicates that the group value for these attributes is taken from the CFSTRUCT definition.
- If you need to do any work with the SMDS, it might be necessary to connect or disconnect the queue manager from the SMDS by entering the START SMDSCONN or STOP SMDSCONN.
- The RESET SMDS command might be necessary to reset the status of an SMDS as required. An example might be when an SMDS is not properly defined and comes up in FAILED state.
- The SMDS status might be any of:
  - ACTIVE
  - FAILED
  - NOTFOUND. Structure shows on DB2, but not allocated in CF.
  - INBACKUP means that structure is in the process of being backed-up.
  - INRECOVER means that the structure is being recovered.
  - UNKNOWN

IBM MQ Explorer has capabilities to manage queue-sharing groups. From the IBM MQ Explorer Navigator pane, the *Queue-sharing Groups* menu is found below the *Queue Managers* menu.

WebSphere Education IBM

DIS CFSTATUS to check health of queue-sharing group

- Component status:
  – Structure status should show ACTIVE; SIZEUSED shows structure % available
  – TYPE(SMDS) status should have STATUS(ACTIVE) ACESS(ENABLED)

```
MQ05 DIS CFSTATUS(*)
… … …
CSQM201I MQ05 CSQMDRTC  DIS
 CFSTATUS DETAILS
CFSTATUS(APPL1)
TYPE(SUMMARY)
CFTYPE(APPL)
STATUS(ACTIVE)
OFFLDUSE(SMDS)
SIZEMAX(50176)
SIZEUSED(27)
ENTSMAX(17103)
ENTSUSED(3043)
```

```
MQ05 DIS CFSTATUS(*) TYPE(SMDS)
 … … …
CSQM201I MQ05 CSQMDRTC  DIS
  CFSTAT
 CFSTATUS(APPL1)
 TYPE(SMDS)
 SMDS(MQ05)
 STATUS(ACTIVE)
 ACCESS(ENABLED)
 RCVTIME(17.46.32)
 RCVDATE(2015-07-24)
 FAILTIME()
 … … …
```


Figure 4-31. DIS CFSTATUS to check health of queue-sharing group WM3121.1

Notes:

Two statuses should be checked when you work with queue-sharing groups.

The CF structure status, which shows you percentage that is used (SIZEUSED), and the number of entries that are used and available (ENTSUSED, ENTSMAX).

- The SMDS status, which shows the ability to use the shared message data set for offload. Use the asterisk as shown in the slide to view the SMDS status per application structure per queue manager in the queue-sharing group.

Example

If you enter **`DIS CFSTATUS(*) TYPE(SMDS)`** for a queue-sharing group that contains two queue managers, MQ05, and MQ25, and has three application structures, APPL1, APPL2, and CSQSYSAPPL, the results are six separate displays:

- `APPL1, APPL2, and CSQSYSAPPL for MQ05`
- `APPL1, APPL2, and CSQSYSAPPL for MQ25`

If you enter **DIS CFSTATUS(*)**, which defaults to TYPE(SUMMARY), for a queue-sharing group with the same configuration described earlier, four results are displayed:

- Application structure CFTATUS for APPL1
- Application structure CFSTATUS for APPL2
- Application structure CFSTATUS for CSQSYSAPPL
- Administration structure CFSTATUS for CSQ_ADMIN

WebSphere Education

Coupling facility commands

- Display commands to obtain structure information

```
/D XCF,STR,STRNAME=SG05*
IXC360I  16.00.35  DISPLAY XCF
STRNAME: SG05APPL1
... ... ...
SPACE USAGE      IN-USE       TOTAL    %
 ENTRIES:          3043       17103   17
 ELEMENTS:        27796      102932   27
 ... ... ...
```

- Configuration commands to manager coupling facility structure

```
/SETXCF START,POL,TYPE=CFRM,POLNAME=CFDBPOL1
IXC511I START ADMINISTRATIVE POLICY CFDBPOL1 FOR CFRM ACCEPTED
IXC513I COMPLETED POLICY CHANGE FOR CFRM. 439
```


Figure 4-32. Coupling facility commands WM3121.1

Notes:

To display the structures by using z/OS coupling facility commands, you use the `/D XCF,STR` command. You can specify a generic or specific structure name.

For a queue-sharing group called SG05 with two queue managers and application structures APPL1, APPL2, and CSQSYSAPPL, if you enter the command as shown on the slide:

`/D XCF,STR,STRNAME=SG05*`, four CF structures are displayed:

- SG05APPL1
- SG05APPL2
- SG05CSQ_ADMIN
- SG05CSQSYSAPPL

The SETXCF utility can process different commands such as enabling definitions in the CFRM policy or adding space to a structure.

Outside the individual commands that can be entered in a z/OS SDSF screen, PDS member CSQ4CFRM is available as a template to define the structures that are used for IBM MQ. After CSQ4CFRM is customized, it can be added to your IXCMIAPU jobs to define the coupling facility structures.

WebSphere Education IBM

Queue-sharing groups implementation checklist (1 of 2)

- Before you start:
  - Ensure the objectives to be accomplished with an offload scheme are clear
  - Obtain expected volumes and arrival patterns, such as peak times
  - Confirm CF levels
- Review capacity and tuning documents for recommendations and information pertinent to your environment:
  - SupportPac MP16: Capacity Planning and Tuning for IBM MQ for z/OS
  - SupportPac MP1H: IBM MQ for z/OS V7.1 performance report
- Set up the DB2 environment and APF authorize the DB2 HLQ.SDSNLOAD library
- Add CF structure definition to CFRM policy – see SCSQPROC member **`CSQ4CFRM`** for templates for the z/OS **`IXCMIAPU`** jobs
- If using SCM, CF level previously checked:
  - Configure CFRM to use **`SCMALGORITHM`**
  - Determine on setting of **`ALLOWAUTOALT`** and **`SCMMAXSIZE`** attributes
- Implement required security
  - Refer to Task 11: Implement your **`ESM`** security controls in the IBM Knowledge Center
  - Check that all queue managers in the group have adequate access to the SMDS

Figure 4-33. Queue-sharing groups implementation checklist (1 of 2) WM3121.1

Notes:

Depending on your environment, you might need to work with different resources to plan the environment. Some of the resources you might need to interface with are:

- The z/OS team to coordinate details of the CFRM policy and DB2 configuration.
- Application analysts who have information on the message volume and arrival patters and overall application requirements.
- As you see in the next unit, you also need the network team to set up dynamic virtual IP addresses (DVIPA) to promote shared channel high availability.

You might need to explain how SMDS and SCM are used for queue-sharing groups and share the SupportPacs cited in this unit with the team you interface, particularly the z/OS team.

Reminder

The following structures must be in the CFRM policy:

- One administrative structure that is called XXXXCSQ_ADMIN, where XXXX is the queue-sharing group name.

- If your application requires group units of recovery, one application structure that is called XXXXCSQSYSAPPL, where XXXX is the queue-sharing group name. If your application does not use group units of recovery, you do not need this structure.
- One or more application structures for your queues. You might want to use different structures under different circumstances. For example, you might have a structure to hold queues that are involved in long units of work (UOWs), separate from a structure for queues that are not used in long UOWs.

WebSphere Education

Queue-sharing groups implementation checklist (2 of 2)

- Create the queue-sharing group in DB2 – CSQ45AQS
- Add the queue manager to the DB2 queue-sharing group – CSQ45AQM
- If using SMDS, use CSQ4SMDS to create one queue manager SMDS for each application structure in the queue-sharing group, and for CSQSYSAPPL if used
- Customize and add the CSQ4INSS member to the CSQINP2 concatenation of the queue manager MSTR started task. CSQ4INSS contains:
  - The IBM MQ CFSTRUCT definition
  - The SYSTEM.QSG.UR.RESOLUTION.QUEUE and CSQSYSAPPL CFSTRUCT – these objects allow enablement of groups unit of recovery (GROUPUR) in the queue manager
  - Set the CFLEVEL structure to 5 to use new features
  - You might need to add the new IBM MQ CFSTRUCT attributes to CSQ4INSS – use the IBM Knowledge Center DEFINE CFSTRUCT documentation for reference
- Add the DB2 HLQ.SDSNLOAD library to the STEPLIB concatenation of the queue manager MSTR started task
- Add QSG information to the CSQ6SYSP parameter macro and reassemble

Figure 4-34. Queue-sharing groups implementation checklist (2 of 2) WM3121.1

Notes:

When you complete the configuration:

- Restart the queue manager.
- Review the start messages to confirm all queue-sharing group-related messages; these messages are prefixed with ILX and CSQE.
- Issue the IBM MQ DIS GROUP command. This example uses queue manager MQ03:

```
MQ03 DIS GROUP
CSQ5100I MQ03 DISPLAY GROUP report ...
CSQ5102I MQ03 Queue managers in group SG03 110
-----------------------------------------------------------------
Name  Num  Prefix    Status      Ver  DB2   Connection
----  ---  --------  ----------  ---  ----  ----------
MQ03    1  MQ03      ACTIVE      800  DBA1  ACTIVE
-----------------------------------------------------------------
End of queue managers report
CSQ9022I MQ03 CSQ5DQSG ' DIS GROUP' NORMAL COMPLETION
```

You complete the tasks in this slide during the lab exercises for this unit.

WebSphere Education IBM

Create the queue-sharing group in DB2: CSQ45AQS

```
//MQ03AQS1 JOB ... ... ...
//ADDQSG   EXEC PGM=CSQ5PQSG,REGION=4M,
//         PARM='ADD QSG,SG03,DSNDBA0,DBA1'
//SYSPRINT DD SYSOUT=*
//STEPLIB  DD DSN=SYS2.MQ8000.SCSQANLE,DISP=SHR
//         DD DSN=SYS2.MQ8000.SCSQAUTH,DISP=SHR
//         DD DSN=DSNDBA0.DB2B10.SDSNLOAD,DISP=SHR
```

```
 CSQ5PQSG Queue-sharing Group Utility - 2015-06-18 10:50:38
 ADD QSG function requested
 QSG=SG03, DB2 DSG=DSNDBA0, DB2 ssid=DBA1
 Connected to DB2 DBA1
 QSG SG03 entry successfully added to DB2 table
  CSQ.ADMIN_B_QSG
 QSG SG03 entry successfully added
 Disconnected from DB2 DBA1
CSQ5PQSG Utility completed, return code=0
```


Figure 4-35. Create the queue-sharing group in DB2: CSQ45AQS WM3121.1

Notes:

Queue-sharing groups require use of DB2 and the coupling facility. While many of the tasks required to configure your queue-sharing group environment can be done in any order, you must complete the DB2 tasks in the order given in this unit. That order is to first define the queue-sharing group to DB2; then, add the queue managers to the queue-sharing group.

The IBM MQ CSQ5PQSG utility creates the required definitions in DB2 by using parameters that you specify.

If you look at the parameter **(PARM)** statement in the example JCL, you can see the command to the CSQ5PQSG utility is “ADD QSG”.

WebSphere Education

Add queue manager to group in DB2: CSQ45AQM

```
//MQ03AQS1 JOB ... ... ...
//ADDQSG   EXEC PGM=CSQ5PQSG,REGION=4M,
// PARM='ADD QMGR,MQ03,SG03,DSNDBA0,DBA1'
//SYSPRINT DD SYSOUT=*
//STEPLIB  DD DSN=SYS2.MQ8000.SCSQANLE,DISP=SHR
//         DD DSN=SYS2.MQ8000.SCSQAUTH,DISP=SHR
//         DD DSN=DSNDBA0.DB2B10.SDSNLOAD,DISP=SHR
```

```
CSQ5PQSG Queue-sharing Group Utility - 2015-06-18 11:46:43
ADD QMGR function requested
Queue manager=MQ03
QSG=SG03, DB2 DSG=DSNDBA0, DB2 ssid=DBA1
Full name of admin structure is SG03CSQ_ADMIN
Unable to get attributes for admin structure, CF not found
 or not allocated
Connected to DB2 DBA1
QMGR MQ03 entry successfully added to QSG SG03
```


Figure 4-36. Add queue manager to group in DB2: CSQ45AQM WM3121.1

Notes:

After you add the queue-sharing group in the previous step, you add the queue manager to the queue-sharing group. The command in the CSQ5PQSG **PARM** statement is now ADD QMGR.

Do not be concerned if you receive the "*Unable to get attributes ...*" message. The administrative structure is not yet allocated; the message is normal until the structure is used.

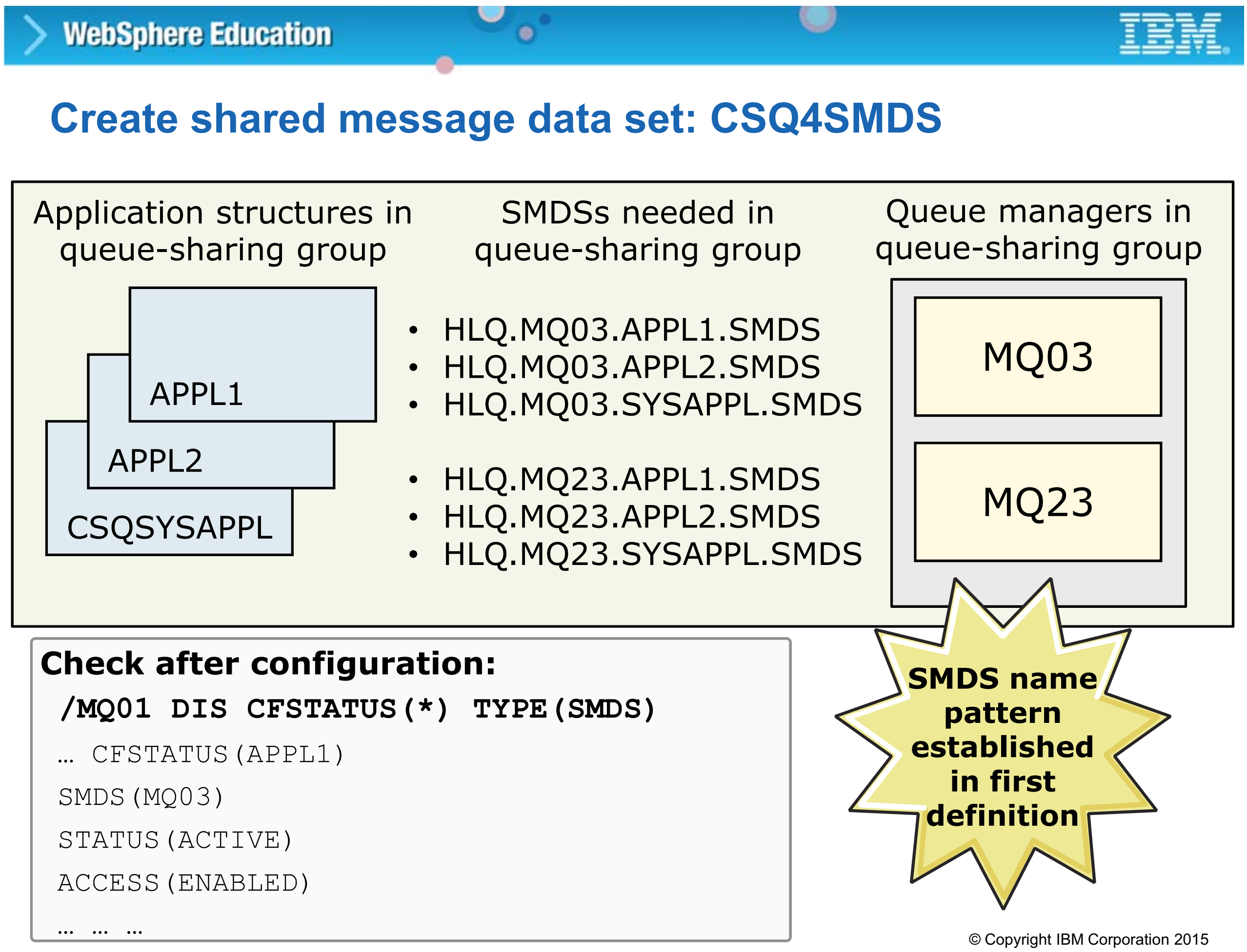


Figure 4-37. Create shared message data set: CSQ4SMDS WM3121.1

Notes:

Although configuration of the shared message data sets (SMDSs) seems simple, incorrect configuration can cause significant rework and frustration.

The number of SMDSs required by each queue manager in the queue-sharing group depends on the number of *application* structures in your queue-sharing group environment:

- If the applications in your environment are involved in group units of recovery, you must define an SMDS for the CSQSYSAPPL structure for each queue manager you add to the queue-sharing group. CSQSYSAPPL is too long to be used as a node in a data set name, so you might abbreviate it to SYSAPPL.
- Add one SMDS data set for every other application structure in the queue-sharing group.

When the first queue manager in the queue-sharing group starts, it “establishes” the naming pattern for subsequent SMDSs from the first data set.

As an example: If the first data set shown on the slide for queue manager MQ03 and structure APPL1 is created, subsequent SMDS data sets for the APPL1 structure in the other queue managers joining the queue-sharing group are expected to be named HLQ.XXXX.APPL1.SMDS. XXXX stands for the queue manager name.

The total number of SMDSs required by a queue-sharing group with two queue managers, which requires a CSQSYSAPPL structure and two other application structures is six, as depicted on the slide.

After you define the shared message data set, you must ensure that your corresponding IBM MQ CFSTRUCT definition or definitions are accurate. *This part of the configuration, which uses PDS member CSQ4INSS is critical.* Update of the CSQ4INSS PDS member, which includes the CFSTRUCT is explained in a subsequent slide.

The name of the SMDS data set is provided to the queue manager in the CFSTRUCT object DSGROUP attribute, in the CSQ4INSS member.

When you specify the SDMS name in the DSGROUP attribute, always use the generic character, or '*', instead of the queue manager name portion of the SMDS name. By continuing with the example in this slide, the DSGROUP attribute for the APPL1 CFSTRUCT in queue manager MQ03 is coded as follows:

`DSGROUP('HLQ.*.APPL1.SMDS')`

As part of the work that is involved in including more queue managers in the queue-sharing group, you make sure to add an SMDS for all structures that are used in the queue-sharing group.

IBM MQ looks for the SMDSs that it knows about as defined in the queue-sharing group. When the first queue manager is added, it looks for:

- HLQ.MQ03.APPL1.SMDS
- HLQ.MQ03.SYSAPPL.SMDS

After the queue manager starts, the SMDS status for all SMDS associated with the queue manager must show a status of ACTIVE and AVAILABLE. Any other status and access results indicate a problem with the configuration.

You use command **`DIS CFSTATUS(*) TYPE(SMDS)`**, for example:

```
CSQM201I MQ05 CSQMDRTC  DIS C
 CFSTATUS(APPL1)
 TYPE(SMDS)
 SMDS(MQ05)
 STATUS(ACTIVE)
 ACCESS(ENABLED)
... ... ...
CSQM201I MQ05 CSQMDRTC  DIS CFSTATUS DE
 CFSTATUS(CSQSYSAPPL)
 TYPE(SMDS)
 SMDS(MQ05)
 STATUS(ACTIVE)
 ACCESS(ENABLED)
 RCVTIME(09.17.13)
```

WebSphere Education

Customize SCSQPROC member CSQ4INSS (1 of 2)

- If you work with IBM MQ V7.1 or later
  - Check that the version of the CSQ4INSS member reflects CF5 and later attributes
  - Attributes might need to be manually added to this member
  - The queues defined in CSQ4INSS use specific CFSTRUCT values

CFSTRUCT objects in CSQ4INSS

```
CFSTRUCT CSQSYSAPPL
   CFLEVEL(5)
   OFFLOAD(SMDS)
  DSGROUP('HLQ.*.SYSAPPL.SMDS)
   RECOVER(YES)
    ...
CFSTRUCT APPL1
    CFLEVEL(5)
    OFFLOAD(SMDS)
    DSGROUP('HLQ.*.APPL1.SMDS)
    RECOVER(YES)
```

SYSTEM queues in CSQ4INSS

```
QLOCAL SYSTEM.QSG.CHANNEL.SYNCQ
    QSGDISP(SHARED)
    CFSTRUCT('CSQSYSAPPL')

QLOCAL SYSTEM.QSG.TRANSMIT.QUEUE
    QSGDISP(SHARED)
    CFSTRUCT('APPL1')

QLOCAL
   SYSTEM.QSG.UR.RESOLUTION.QUEUE
    QSGDISP(SHARED)
    CFSTRUCT('CSQSYSAPPL')
```


Figure 4-38. Customize SCSQPROC member CSQ4INSS (1 of 2) WM3121.1

Notes:

Completion of the CSQ4INSS updates might be a most challenging part of your configuration.

At the time the course was written, there are six IBM MQ object definitions in CSQ4INSS: two CFSTRUCT objects and three SYSTEM local queues. These local queues need the correct CFSTRUCT designated:

- The application structure, which uses APPL1 as an example, is specified for the group transmission queue, SYSTEM.QSG.TRANSMIT.QUEUE.
- The CSQSYSAPPL structure must be specified for the SYSTEM.QSG.CHANNEL.SYNCQ and SYSTEM.QSG.UR.RESOLUTION.QUEUE.

You must be careful with:

- The version of the CSQ4INSS member, that is, ensure that it contains required attributes and CFLEVEL values. It might be necessary for you to add attributes to this member.
- The number of CFSTRUCT statements you need.
- How **OFFLOAD** and **DSGROUP** attributes are specified.

- The timing, in the order of adding queue managers to the queue-sharing group, that you include or omit definitions.

If you overlook adding CSQ4INSS to your queue manager MSTR procedure, the queue manager starts with partial shared queue functions and a dynamic IBM MQ CFSTRUCT object is defined.

However, the CFSTRUCT object that is created uses an older CF level, which does not allow persistent messages. You are also missing IBM MQ system objects that are used for certain queue-sharing group processes. Overall, the shared queue manager has limited use of the queue-sharing group capabilities.

To remedy this situation, you correct the problem by customizing and including CSQ4INSS, and ensure that you use REPLACE at the end of the IBM MQ CFSTRUCT object definitions. You also need to create the IBM MQ system objects that are included in CSQ4INSS to support shared queues, so it is best to use CSQ4INSS and avoid a manual alteration of the IBM MQ CFSTRUCT.

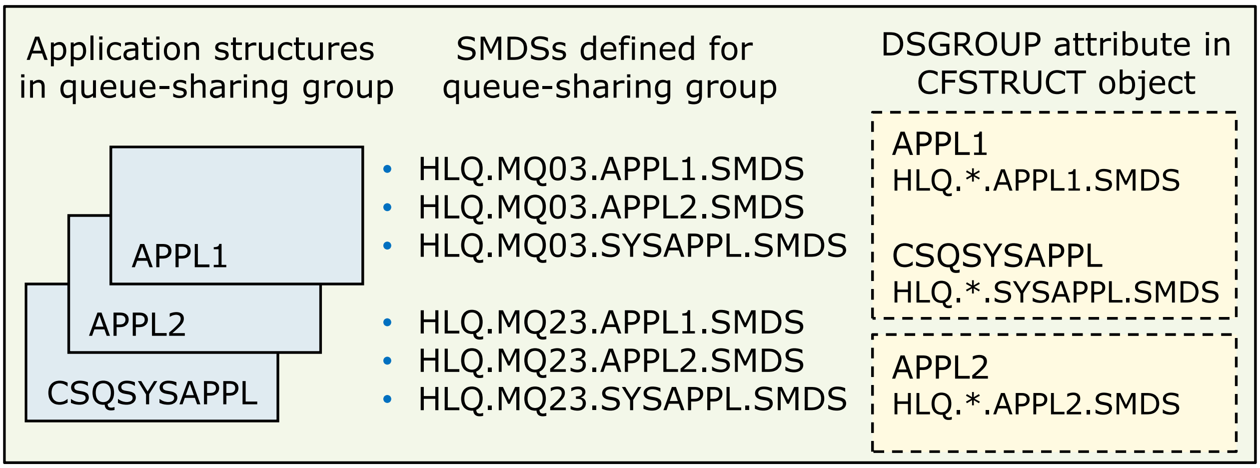
The IBM MQ CFSTRUCT object is not updated if there are any messages in the coupling facility, so you must empty all queues before you are able to alter the IBM MQ CFSTRUCT object.

Always display the updated IBM MQ CFSTRUCT object after update to ensure that the expected changes did take place.

WebSphere Education

Customize SCSQPROC member CSQ4INSS (2 of 2)

- SMDS configuration for each new structure added to the group
  - CFSTRUCT OFFLOAD attribute value set to SMDS
  - CFSTRUCT DSNAME set to SMDS name pattern for the application structure



It is assumed the CSQSYSAPPL and APPL1 objects are included in CSQ4INSS when MQ03 is added to the queue-sharing group, and APPL2 included when MQ23 is added

Figure 4-39. Customize SCSQPROC member CSQ4INSS (2 of 2) WM3121.1

Notes:

Warning

Some of the new attributes were not yet present in the IBM MQ V8 version of the CSQ4INSS member at the time this course was written. CSQ4INSS is under revision and should be updated in a subsequent software update.

If you find that your CSQ4INSS does not include the newer updates, you must manually add the new attributes. Earlier, IBM MQ V8 for z/OS CSQ4INSS member shows a maximum CFLEVEL 3, and does not include new attributes.

There are two IBM MQ CFSTRUCT definitions in CSQ4INSS. You must ensure that the CFLEVEL, recoverability, and other attributes are changed to reflect your requirements. If you are on IBM MQ V7.1 and later, check that CFLEVEL(5) is specified, and check all other recovery and toleration attributes. The structure definitions included in CSQ4INSS are:

- The queue manager application structure, such as ‘APPL1’

- The (optional) group unit of recovery structure CSQSYSAPPL
- System queues that are used by the queue-sharing group environment

The SYSTEM QLOCAL definitions in CSQ4INSS use the application structure names as an attribute to define these objects.

In the next few slides, you learn more about shared message data sets (SMDS). The DSGROUP attribute of the IBM MQ CFSTRUCT object is where you tell IBM MQ the name of the SMDS associated with your application structure.

Note

There must be a naming convention across all the shared message data sets for the same queue-sharing group. The name of the data set should include the queue manager name. When providing the queue manager name to the IBM MQ CFSTRUCT object by using the DSGROUP attribute, the queue manager name is substituted by a generic indicator, or ‘*’. For example:

HLQ.*.APPL1.SMDS

IBM MQ inserts the queue manager name in the space where the ‘*’ included as needed.

If you placed any messages in the structure and you need to make changes after the structure is created, you must empty the structure of any messages first.

WebSphere Education

Update the queue manager started task

- Add the DB2 library to STEPLIB
- Add the customized CSQ4INSS to the CSQINP2 data set
- Ensure that CSQ4INSS precedes CSQ4IYNG
- Ensure that the updated procedure is copied to the active procedure library

```
 BROWSE    SYS3.STUDENT.PROCLIB(MQ03MSTR) - 01.05
 Command ===>
... ... ...
//STEPLIB    DD DSN=SYS2.MQ8000.SCSQANLE,DISP=SHR
//           DD DSN=SYS3.STUDENT.SCSQAUTH,DISP=SHR
... ... ...
//           DD DSN=DSNDBA0.DB2B10.SDSNLOAD,DISP=SHR
============================================================
//CSQINP2    DD DSN=SYS2.MQ8000.SCSQPROC(CSQ4INYS),DISP=SHR
... ... ...
//           DD DSN=TSM0003.WM31.SCSQPROC(CSQ4INSS),DISP=SHR
//           DD DSN=TSM0003.WM31.SCSQPROC(CSQ4IYNG),DISP=SHR
   ... ... ...
```


Figure 4-40. Update the queue manager started task WM3121.1

Notes:

In most environments, there are two changes to the queue manager procedure for queue-sharing groups:

- Add the DB2 library to the STEPLIB concatenation
- Add the updated version of CSQ4INSS to the CSQINP2 concatenation

When you use group channel recovery, there is one other change to include start of a group listener in CSQINPX; this change is detailed in the *Distributed queuing and queue-sharing groups* unit.

Warning

There are two details in this part of the configuration that might cause time-consuming errors if they are omitted:

1. If the **`ALTER QMGR GROUPUR(ENABLED)`** runs before queue `SYSTEM.QSG.UR.RESOLUTION.QUEUE` is defined, the queue manager is changed back to **`GROUPUR(DISABLED)`**.

2. Ensure that the CFSTRUCT definitions are created by using your updated CSQ4INSS member. If there is no predefined CFSTRUCT object when the structure is allocated, IBM MQ creates a back-leveled structure, which might be challenging to rectify. Always ensure:
   - You update your copy of CSQ4INSS with the new attributes.
   - Include CSQ4INSS before CSQ4IYNG in the CSQINP2 concatenation of the **`MQ??MSTR`** procedure.
   - Remember to copy the updated **`MQ??MSTR`** procedure in the active procedure library.

WebSphere Education

Update CSQ6SYS parameter macro

- QSGDATA has positional parameters
  - First is queue-sharing group name. For this example, it is SG03.
  - Second is data sharing group. For this example, it is DSNDBA0.
  - Third is the DB2 SSID. For this example, it is DBA1.
  - Fourth is number of DB2 server tasks. Initial value is 4.
  - Fifth is the number of DB2 server tasks to access binary large objects (BLOBS). Initial value is 4.

```
CSQ6SYSP                                                    X
      CLCACHE=STATIC,       CLUSTER CACHE TYPE              X
      CMDUSER=TSM0003,      $WM302 DEFAULT USERID FOR CMDS  X
      ... ... ...                                           X
      OPMODE=(NEWFUNC,800), IBM MQ V8 OPMODE                X
      ... ... ...                                           X
      QSGDATA=(SG03,DSNDBA0,DBA1,4,4), QSG GROUP DATA       X
      ... ... ...                                           X
```

- Optional: Rename the updated parameter load module

Figure 4-41. Update CSQ6SYS parameter macro WM3121.1

Notes:

Update the QSGDATA attribute in the CSQ6SYS macro with the positional values required to enable queue-sharing groups.

Take care to check that the ‘X’ continuation characters are lined up in column 72 after you make changes. Failure to have the ‘X’ in the correct column causes assembly errors in the parameter macro.

You might want to:

- Create a copy of your CSQ4ZPRM member to make the queue-sharing group updates.
- Rename the new parameter module, for example, MQ03ZQSG, until you are sure the queue-sharing group configuration works. You start your queue manager with the renamed parameter module for queue-sharing group testing.

WebSphere Education IBM

Restart the queue manager and define shared queues

- `DEFINE QLOCAL(queue-name) QSGDISP(SHARED) CFSTRUCT(str-name)`
- `DEFINE QLOCAL(FIRST.SHARED.QUEUE) QSGDISP(SHARED) CFSTRUCT(APPL1)`
- Use application or sample MQPUT to send messages to the queue
- Review messages in the system log to confirm that structure was allocated

Figure 4-42. Restart the queue manager and define shared queues WM3121.1

Notes:

After you start the queue manager with the queue-sharing group configuration and the updated parameter module, carefully check all messages in the system log or IBM MQ MSTR output for any errors.

WebSphere Education

IBM MQ log messages upon first put to structure (1 of 2)

```
$HASP373 MQ03JPUT STARTED - INIT 3     - CLASS W          - SY
CSQE140I MQ03 CSQEENFR Started listening for ENF 35
events for structure APPL1
IXC582I STRUCTURE SG03APPL1 ALLOCATED BY SIZE/RATIOS.
  PHYSICAL STRUCTURE VERSION: CF1F60B1 471FD34C
  STRUCTURE TYPE:             SERIALIZED LIST
  CFNAME:                     CF11
  ALLOCATION SIZE:               49 M
  MAX STRUCTURE SIZE:           196 M
  POLICY SIZE:               200000 K
  POLICY INITSIZE:            50000 K
  ENTRY COUNT:                17103
  ELEMENT COUNT:             102932
  EMC COUNT:                   7882
  LOCKS:                       1024
```


Figure 4-43. IBM MQ log messages upon first put to structure (1 of 2) WM3121.1

Notes:

Check the system log after you complete the first PUT of messages to a shared queue to confirm that the structure was correctly allocated.

WebSphere Education

IBM MQ log messages upon first put to structure (2 of 2)

```
 EMC COUNT:                     7882
  LOCKS:                         1024
  ENTRY:ELEMENT RATIO:              1 : 6
  EMC STORAGE PERCENTAGE:        5.00 %
ALLOCATION SIZE IS WITHIN CFRM POLICY DEFINITIONS
IXL014I IXLCONN REQUEST FOR STRUCTURE SG03APPL1 513
WAS SUCCESSFUL.  JOBNAME: MQ03MSTR ASID: 004C
CONNECTOR NAME: CSQESG03MQ0301 CFNAME: CF11
IXL015I STRUCTURE ALLOCATION INFORMATION FOR 514
STRUCTURE SG03APPL1, CONNECTOR NAME CSQESG03MQ0301,
CONNECTIVITY=DEFAULT
 CFNAME     ALLOCATION STATUS/FAILURE REASON
 --------   ----------------------------------------
 CF11       STRUCTURE ALLOCATED AC001800
... ... ...
CSQE141I MQ03 CSQEENFR Stopped listening for ENF 35 515
events for structure
```


Figure 4-44. IBM MQ log messages upon first put to structure (2 of 2) WM3121.1

Notes:

In this second part of the output that is created after your first PUT to a shared queue, you see the IBM MQ entry to element ratio as 1:6.

WebSphere Education

Unit summary

- Describe shared queues and their architectural role
- List the components that are used by queue-sharing groups
- Explain the relationship of the coupling facility capabilities and the CFLEVEL
- Describe queue-sharing group offload and overflow options
- Describe scenarios that are applicable to use of shared message data sets (SMDS) or storage class memory (SCM)
- Explain how messages are stored in a coupling facility application structure
- Contrast how the coupling facility application and administrative structures are used
- Describe the coupling facility resource management (CFRM) attributes to consider in queue-sharing groups
- Explain queue manager and structure failure, persistence, connectivity, and recovery as they apply to queue-sharing groups
- List IBM MQ administrative commands that are related to queue-sharing groups
- Explain how to configure and test a queue-sharing group

Figure 4-45. Unit summary WM3121.1

Notes:

WebSphere Education

Checkpoint questions (1 of 2)

1. Select the best answer. What is the best reason to use shared queues in the infrastructure?
   a. High availability. Messages in a shared queue are available even if the originating or receiving queue manager is down.
   b. Reliability. Messages in shared queues tolerate outages and are never lost.
   c. Performance. Shared queues provide faster access to messages.
   d. All of the above.
2. The minimum IBM MQ version and CF level that is required to be able to offload messages to a shared message data set (SMDS) is:
   a. IBM MQ V7.1 CFLEVEL(5)
   b. IBM MQ V5.3 CFLEVEL(3)
   c. All supported IBM MQ versions at CFLEVEL(4) and higher
   d. IBM MQ V8 CFLEVEL19
3. True or false. Messages in a coupling facility are held in application structures.

Figure 4-46. Checkpoint questions (1 of 2) WM3121.1

Notes:

Write your answers here:

1.

2.

3.

WebSphere Education

Checkpoint questions (2 of 2)

4. IBM MQ checks the following resource to determine the name of the SMDS:
   a. SMDS1 data set in the queue manager master procedure JCL
   b. DSNAME attribute of the queue manager object
   c. DSNAME attribute of the CSQ6SYSP macro
   d. DSGROUP attribute of the IBM MQ CFSTRUCT object
5. True or false: When planning to implement queue-sharing groups, it is imperative to review *support pac MQ16: Capacity Planning and Tuning for IBM MQ for z/OS* and *support pac MP1H: IBM MQ for z/OS V7.1 performance report.*

Figure 4-47. Checkpoint questions (2 of 2) WM3121.1

Notes:

Write your answers here:

4.

5.

WebSphere Education

Checkpoint answers (1 of 2)

1. Select the best answer. What is the best reason to use shared queues in the infrastructure?
   a. High availability. Messages in a shared queue are available even if the originating or receiving queue manager is down.
   b. Reliability. Messages in shared queues tolerate outages and are never lost.
   c. Performance. Shared queues provide faster access to messages.
   d. All of the above.

   Answer: a. Non-persistent messages are lost, and performance depends on many factors. The best answer is a, particularly if the coupling facility is duplexed.
2. The minimum IBM MQ version and CF level that is required to be able to offload messages to a shared message data set (SMDS) is:
   a. IBM MQ V5.3 CFLEVEL(3)
   b. All supported IBM MQ versions at CFLEVEL(4) and higher
   c. IBM MQ V7.1 CFLEVEL(5)
   d. IBM MQ V8 CFLEVEL19

   Answer: c. SMDS was introduced in IBM MQ V7.1 at CFLEVEL(5)
3. True or false. Messages in a coupling facility are held in application structures.

   Answer: True. Administrative structures hold UOW and other details.

Figure 4-48. Checkpoint answers (1 of 2) WM3121.1

Notes:

WebSphere Education

Checkpoint answers (2 of 2)

4. IBM MQ checks the following resource to determine the name of the SMDS:
   a. SMDS1 data set in the queue manager master procedure JCL
   b. DSNAME attribute of the queue manager object
   c. DSNAME attribute of the CSQ6SYSP macro
   d. DSGROUP attribute of the IBM MQ CFSTRUCT object

   Answer: d. Use same naming convention for all queue managers in the queue-sharing group and use an '*' for the queue manager name. Specify OFFLOAD=SMDS, and define the SMDS data set with member CSQ4SMDS.

5. True or false: When planning to implement queue-sharing groups, it is imperative to review *support pac MQ16: Capacity Planning and Tuning for IBM MQ for z/OS* and *support pac MP1H: IBM MQ for z/OS V7.1 performance report.*

   Answer: True. These publications contain critical information that address different variables in a queue-sharing group environment.

Figure 4-49. Checkpoint answers (2 of 2) WM3121.1

Notes:

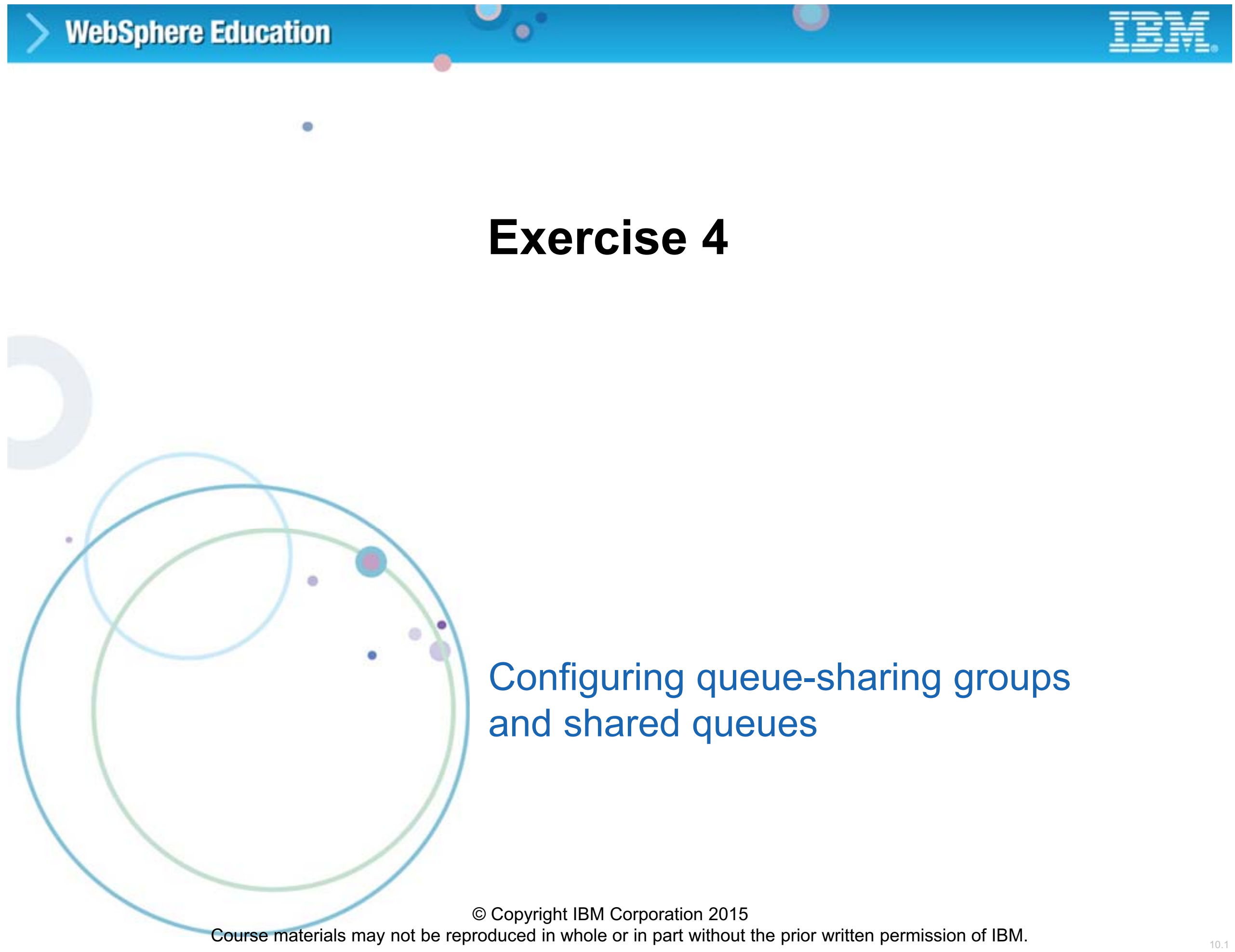


Figure 4-50. Exercise 4 WM3121.1

Notes:

WebSphere Education

Exercise objectives

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

Figure 4-51. Exercise objectives WM3121.1

Notes:

Unit 5. Intra-group, distributed queuing, and clustering with queue-sharing groups

What this unit is about

This unit describes the extra capabilities that are available for queue manager communication with queue-sharing groups.

What you should be able to do

After completing this unit, you should be able to:

- Explain the advantages of intra-group and distributed queuing with queue-sharing groups
- Describe how to set up peer channel recovery
- Explain intra-group queuing
- Explore various queue-sharing group configurations
- Describe how to add a queue-sharing group to an existing cluster
- Describe the optimal client connection configuration
- Explain queue-sharing group differences with START CHANNEL and DISPLAY CHSTATUS commands

How you will check your progress

- Checkpoint questions
- Lab exercise

WebSphere Education

Unit objectives

- Explain the advantages of intra-group and distributed queuing with queue-sharing groups
- Describe how to set up peer channel recovery
- Explain intra-group queuing
- Explore various queue-sharing group configurations
- Describe how to add a queue-sharing group to an existing cluster
- Describe the optimal client connection configuration
- Explain queue-sharing group differences with START CHANNEL and DISPLAY CHSTATUS commands

Figure 5-1. Unit objectives WM3121.1

Notes:

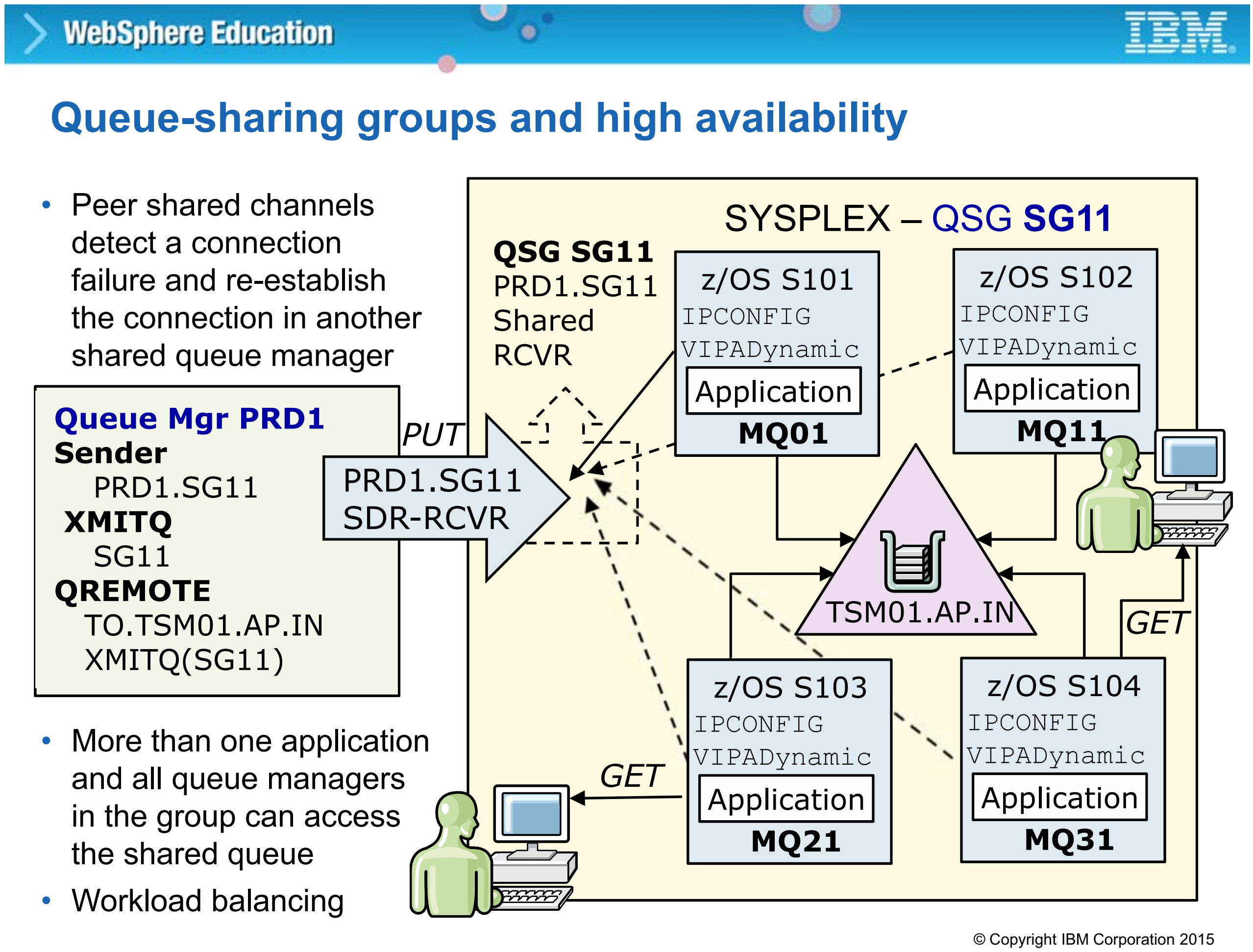


Figure 5-2. Queue-sharing groups and high availability WM3121.1

Notes:

Queue-sharing groups increase availability to the network and allow any queue manager in the queue-sharing group to access shared queues. When coupled with a network product such as sysplex distributor, channels can be set up for peer recovery.

- Peer shared channels reconnect to another queue manager if a problem is detected
  - Another queue manager in the queue-sharing group detects the problem
  - Transactions are backed out and the message is returned to be processed by another queue manager
  - Each queue manager in the queue-sharing group should have a clone of the application that gets messages
- More than one queue manager can have a connection from a requesting application
- Applications in more than one queue manager can process messages from the shared queue simultaneously

- When a channel is triggered, starts with the START CHANNEL command, or goes through channel peer recovery, IBM MQ load balances by selecting the best channel initiator candidate to process the request.

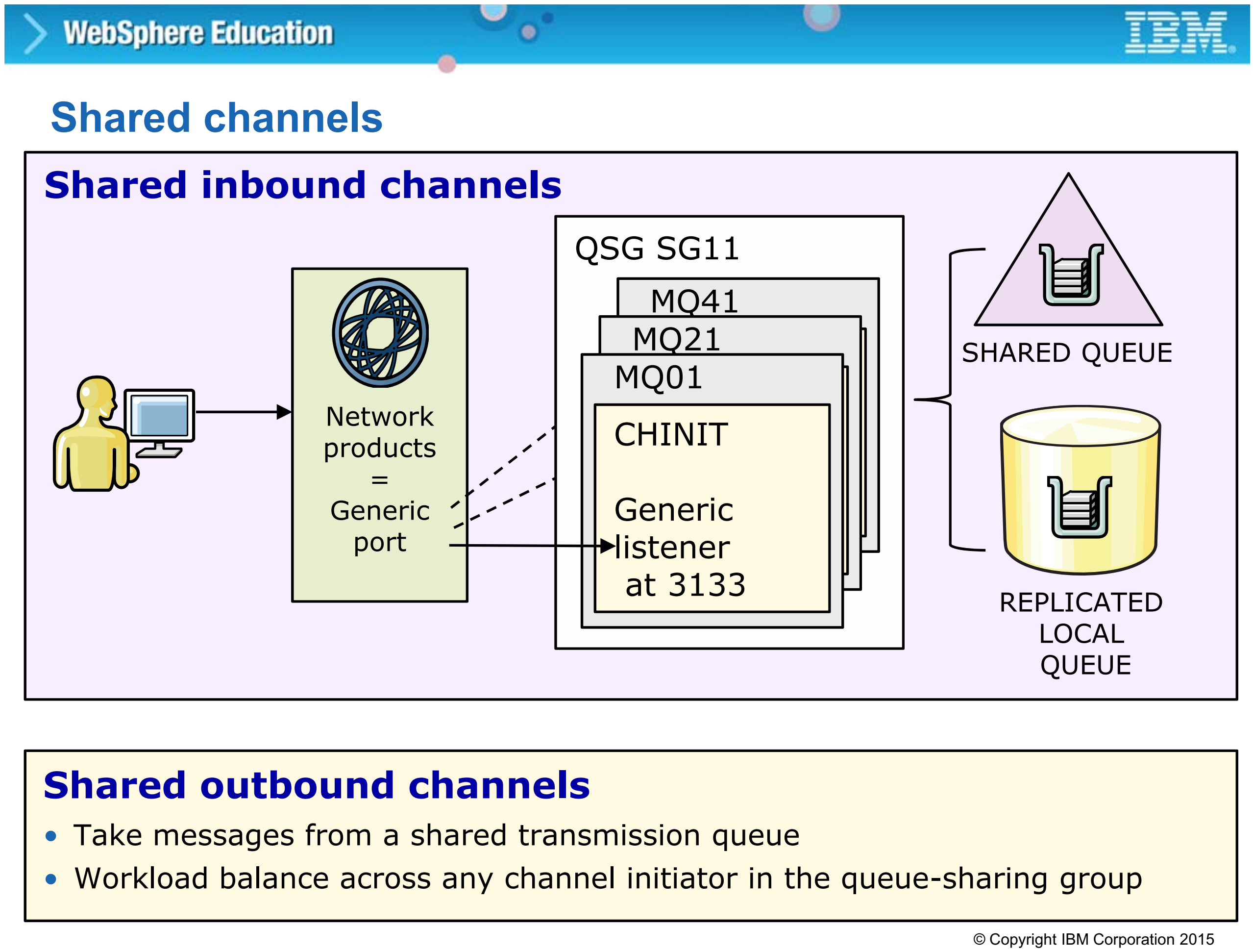


Figure 5-3. Shared channels WM3121.1

Notes:

The concept of shared channels is slightly different than shared queues in that a channel defined with QSGDISP(GROUP) is not a shared channel unless it meets certain criteria, although its definition is stored in DB2. This criteria is different for inbound or outbound channels.

Shared channels can access private local queues or shared queues. To access private queues, the private queues must be defined in each queue manager in the queue-sharing group. These queues are referred to as *replicated* queues.

An *inbound* shared channel uses a dynamic virtual IP address (DVIPA) and listens on a generic port. Each channel initiator in the queue-sharing group starts a generic listener. A receiving channel is shared if it started in response to an inbound message directed to the queue-sharing group.

Channel synchronization for a channel setup for peer channel recovery is with the queue-sharing group, not with the queue manager.

The channel initiators in a queue-sharing group maintain a shared channel-status table in DB2. If a channel failure is detected, IBM MQ uses the information in this table to determine what other peer channel is available to restart the connection. The ability to reconnect by using another available channel is referred to as *peer channel recovery.*

Use of the DB2 does have performance considerations, but the benefits that are derived usually outweigh performance concerns.

In cases such as *client channels*, which do not require to keep channel state, the generic port does not need to be used. A separate port can be created, different than the port used for distributed queuing to use the channels without the performance overhead. In this case, the client connection uses the DVIPA IP address and the second non-generic listener for the connection. This second listener must be started in all queue managers in the queue-sharing group. The server connection channel would have the QSGDISP attribute as GROUP, so a definition of the channel is available to all queue managers in the queue-sharing group.

Outbound shared channels use a shared transmission queue and get started by using the CHLDISP(SHARED) attribute of the START CHANNEL command. A shared outbound channel also holds status information at the queue-sharing group level, and is eligible for *peer channel recovery*.

If an outbound shared channel is started without specifying on which queue manager to start, workload balancing takes place to determine the most suitable queue manager where the channel should start. The decision is made based on:

- Which channel initiator in the queue-sharing group has the lightest load?
- Is there a connection to DB2?
- Is the required communication subsystem available to the channel initiator?

IBM MQ load balances when a channel's shared transmission queue is triggered so the channel starts automatically, and when a channel is manually started without specifying a queue manager name. Workload balancing for the channels is based on the *channel load* and *headroom* of the channel initiator.

- *Channel load* is the number of active channels, which are shown as a percentage of the maximum number of available channels that are allowed according to the channel initiator parameters.
- *Headroom* refers to the difference between the number of active channels and the maximum number of channels allowed.

WebSphere Education IBM

Shared versus private channel comparison

Resources	SHARED	PRIVATE
Transmission queue	Uses a shared transmission queue	Uses a private transmission queue
Port	Start inbound channel through a generic port	Start inbound channel through a private port
Synchronization Information	With queue-sharing group SYSTEM.QSG.CHANNEL.SYNCQ	With queue manager SYSTEM.CHANNEL.SYNCQ
Channel status	When generic listener is used there is a channel status table in DB2 for channel peer recovery	No use of DB2

Figure 5-4. Shared versus private channel comparison WM3121.1

Notes:

Summarize the differences between shared and private channels:

- A shared outbound channel must use a shared transmission queue available in the coupling facility, while a private channel uses a private (non-shared) queue.
- A shared inbound channel uses the generic port that is defined for the queue-sharing group. A private channel uses the queue manager's private port.
- Shared channels keep synchronization information in the SYSTEM.QSG.CHANNEL.SYNCQ, while private channels keep synchronization information in the SYSTEM.CHANNEL.SYNCQ.
- The channel status for shared channels is kept in DB2, which allows for channel peer recovery with a performance cost. Private channels do not use DB2 and do not incur the performance cost but they cannot participate in channel peer recovery, although they can reconnect to another available non-generic listener in the queue-sharing group.

WebSphere Education

Peer channel recovery configuration topics

- Example of a generic port sysplex distributor definition.
- Display generic port configuration.
- Start of generic listener at port 3135. Private listeners are on 3105, not in diagram.
- Shared inbound channel and related queue configuration.
- Start of shared inbound channel.
- Review of peer channel recovery outputs when channel initiators alternate restarts.

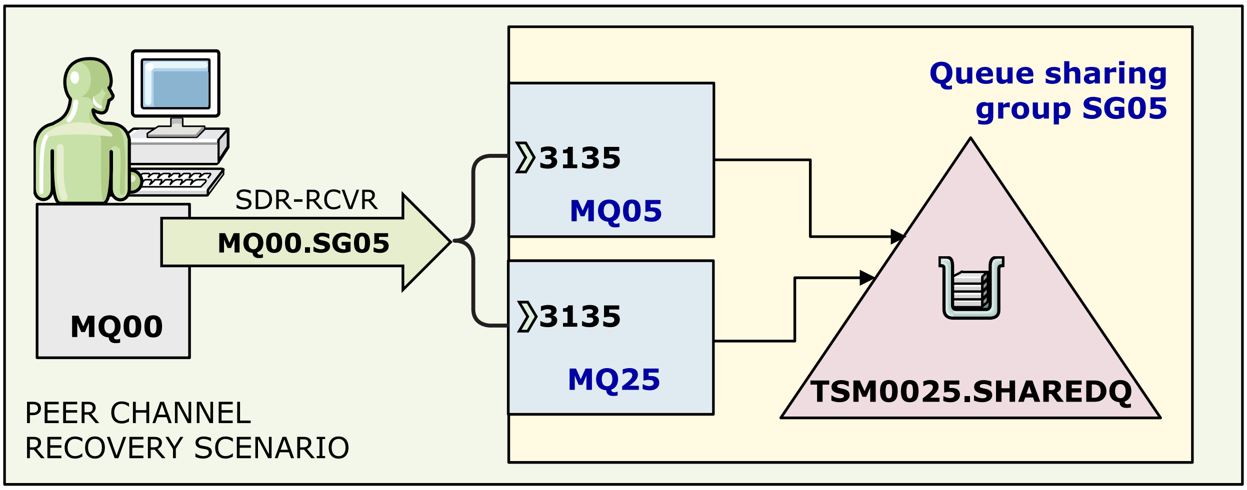


Figure 5-5. Peer channel recovery configuration topics WM3121.1

Notes:

This slide introduces implementation of peer channel recovery. The scenario that is used as an example shows one application that connects with a sender channel by using the DVIPA IP address and generic port. Queue-sharing group SG05 consists of two queue managers, MQ05 and MQ25. The chevron icon signifies the generic port that is used, 3135, which is started in each queue manager's channel initiator in addition to the queue manager's private port.

The application uses a remote queue called TO.TSM0025.SHAREDQ (not shown) which resolves to shared queue TSM0025.SHAREDQ.

The channel definition for MQ00 is with the queue-sharing group, SG05.

In the next slides, you look at:

- An example of a dynamic VIPA (DVIPA) configuration similar to the configuration used in the course.
- How to display/confirm the DVIPA and generic port configuration.
- How the start of the generic listener is incorporated in the channel initiator start.
- The configuration of IBM MQ objects and start of shared inbound channel

- The channel peer recovery process when alternating starts and stops of the channel initiators of each queue manager.

There are different details to consider when working with shared channels and peer recovery.

Table 5-1: Types of shared channel failure

Failure	Results
Channel initiator communication subsystem fails	Channels that use the communication start retry; then, are restarted in the best candidate of the other available channel initiators in the queue-sharing group.
Channel initiator fails	The associated queue manager monitors the failure and initiates recovery.
Queue manager fails	Other queue managers in the queue-sharing group detect failure and initiate peer recovery.
Shared status (DB2) failure	Loss of connectivity to DB2 results in the inability to complete peer channel recovery. There is no impact to running channels, but when it becomes necessary to consult the status on DB2 for recovery and access to DB2 fails, the channel goes into retry.

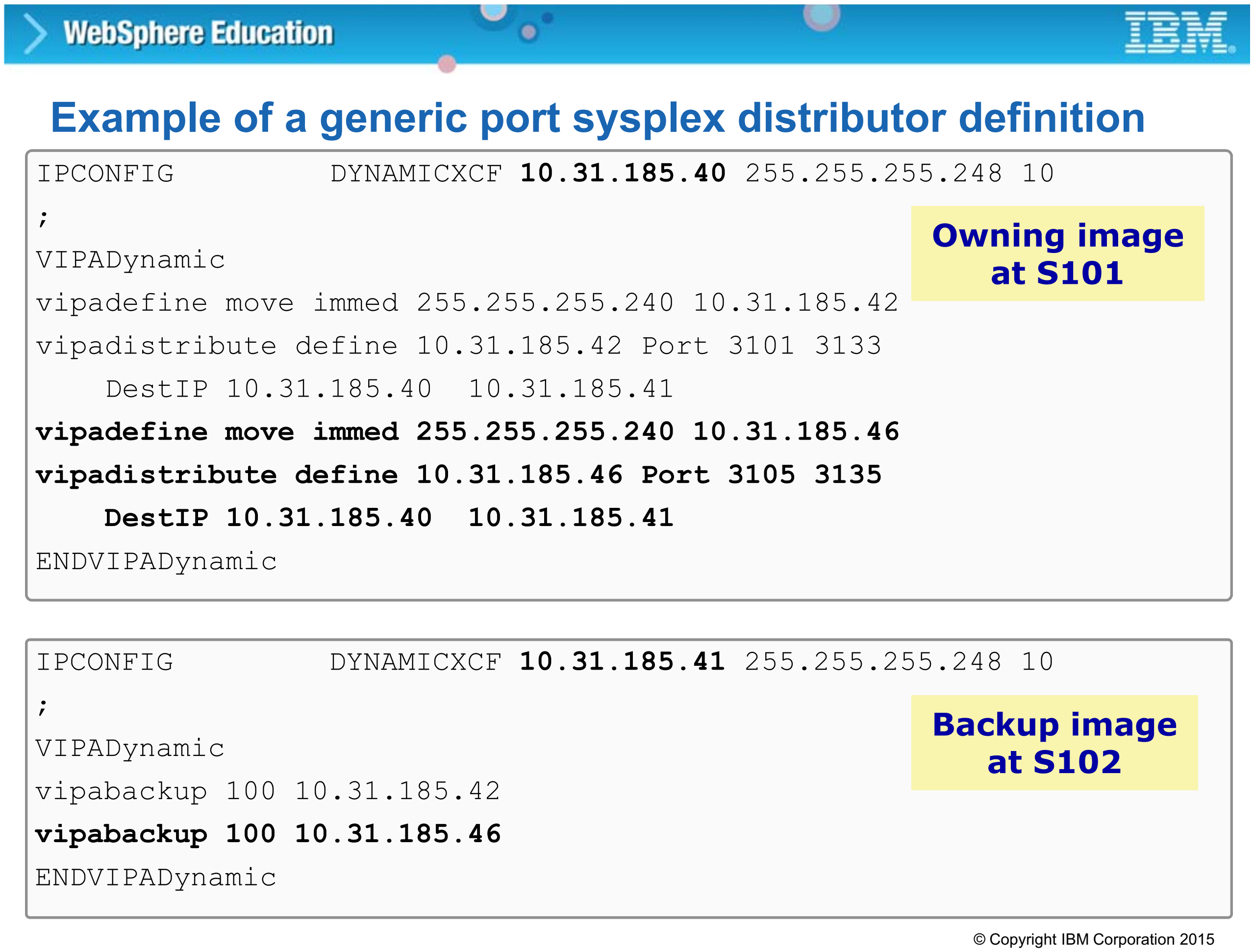


Figure 5-6. Example of a generic port sysplex distributor definition WM3121.1

Notes:

Since in most organizations the networking setup is deferred to a separate group from IBM MQ administration, you might need to explain the IBM MQ peer channel recovery capabilities before sending your request for DVIPA definitions. If the networking team does not understand the process, they might provide definitions to have a queue manager come up on other images, which is *not* what is required.

It is important that the resources understand that IBM MQ itself is restarting the connection on another available channel for a different queue manager member of the queue-sharing group. An IBM MQ queue-sharing group *does not* start the same failed queue manager on a different LPAR.

The IP to be used for the DVIPA definitions is usually obtained from the network team.

In the two screens in this slide, the first IP address following the DYNAMICXCF key word denotes the LPAR’s IP address: S101 is at 10.31.184.40 and S102 is at 10.31.185.41.

Follow the definitions applicable to the SG05 queue-sharing group by looking at the *second set* of **VIPADEFINE** and **VIPADISTRIBUTE** statements, which are in bold.

In the **VIPADEFINE** statement, **10.31.185.46** was selected for queue-sharing group SG05. The queue manager port, 3105, and the generic port, 3135 are added to the **PORT** statement. *All the* *queue managers in the queue-sharing group use private port 3105,* while 3135 is the queue-sharing group port.

The destination IP (**DestIP**) parameter contains the IPs for both LPARs. Some practitioners might choose to use **"DestIP ALL"** instead of specifying specific IP addresses. The use of **"DestIP ALL"** might result in problems that are difficult to rectify as documented in the warning box that follows.

Warning

IBM MQ multi-stack is not supported in a VIPADISTRIBUTE DESTIP ALL environment and results in an error as documented in the following technote:
http://www.ibm.com/support/docview.wss?uid=swg21239625

The example DVIPA configuration that is provided follows the "owning" and "backup image" setup as presented in the *Establishing a TCP connection Using Sysplex Distributor* section of the IBM Knowledge Center. In this section, one queue manager in the queue-sharing group "owns" the VIPADEFINE and **VIPADISTRIBUTE** statements, and the backup images have the **VIPABACKUP** statement.

When there is more than one channel initiator in the same LPAR, the **SHAREPORT** option must be added so that the port is shared in the **PORT** reservation list in the **PROFILE** data set.

WebSphere Education IBM

Display generic port configuration

```
D TCPIP,,N,VIPADCFG
5:40:52 PM: RESPONSE=S101
 EZZ2500I NETSTAT CS V2R1 TCPIP 279
 DYNAMIC VIPA INFORMATION:
   VIPA DEFINE:
     IP ADDRESS       ADDRESSMASK      MOVEABLE  SRVMGR FLG
    ------------     -------------    ---------- ------- ---
     10.31.185.42     255.255.255.240  IMMEDIATE  NO
     10.31.185.46     255.255.255.240  IMMEDIATE  NO
   VIPA DISTRIBUTE:
     IP ADDRESS       PORT   XCF ADDRESS      SYSPT  TIMAFF  FLG
    ------------     -----  -------------    -----  ------   ---
     10.31.185.42     3101   10.31.185.40     NO     NO
     10.31.185.42     3101   10.31.185.41     NO     NO
     10.31.185.42     3133   10.31.185.40     NO     NO
     10.31.185.42     3133   10.31.185.41     NO     NO
     10.31.185.46     3105   10.31.185.40     NO     NO
     10.31.185.46     3105   10.31.185.41     NO     NO
     10.31.185.46     3135   10.31.185.40     NO     NO
     10.31.185.46     3135   10.31.185.41     NO     NO
```


Figure 5-7. Display generic port configuration WM3121.1

Notes:

Confirm that the DVIPA configuration is available by typing the **/D TCPIP,,N,VIPADCFG** command as shown. Take care to include the commas and space.

For SG05, you see that the DVIPA, 10.31.185.46 is in the **VIPA DEFINE** section. 10.31.185.46 is also present, starting in row 5 of the **VIPA DISTRIBUTE** section, and available for both LPAR IP addresses, 10.31.184.40 and 10.31.185.41. The **VIPA DISTRIBUTE** entries also show the queue manager and queue-sharing group ports.

WebSphere Education IBM

Start of generic listeners

- Include the command to start the listener in the CSQINPX SCSQPROC member

```
EDIT   TSM0005.WM31.SCSQPROC(CSQINPX) - 01.06    Columns 00001
 00072
 Command ===>                                          Scroll ===> CSR
 ****** ********************** Top of Data **********************
 000001 START LISTENER PORT(3105) <=== initial listener start
 000002 START LSTR TRPTYPE(TCP) PORT(3135) IPADDR(10.31.185.46) +
 000003    INDISP(GROUP)  <=== generic port listener start
 ****** ********************* Bottom of Data *********************
```

- The command can be typed manually for test purposes, but the generic listener needs to start automatically when the channel initiator is recycled

Figure 5-8. Start of generic listeners WM3121.1

Notes:

Include the START statement for the generic port in the CSQINPX SCSQPROC member. Examine the extra parameters in the START LISTENER statement:

- The DVIPA IP is included in the IPADDR attribute
- The port is started with attribute INDISP set to GROUP disposition

WebSphere Education

Shared inbound channel and related queue configuration

Definitions outside QSG

Queue manager MQ00

```
DEF QL(SG05) USAGE(XMITQ) +
  TRIGGER +
  INITQ(SYSTEM.CHANNEL.INITQ) +
  TRIGDATA(MQ00.SG05)

DEF CHL(MQ00.SG05) +
  CHLTYPE(SDR) +
  TRPTYPE(TCP) XMITQ(SG05) +
  CONNAME('10.31.185.46(3135)')

DEF QR(TO.TSM0025.SHAREDQ) +
   XMITQ(SG05) RQMNAME(SG05) +
   RNAME(TSM0025.SHAREDQ)
```

SG05 definitions

Queue manager MQ05

```
DEFINE CHANNEL(MQ00.SG05)+
  CHLTYPE(RCVR) +
  TRPTYPE(TCP) +
  QSGDISP(GROUP)
```

Queue manager MQ25

```
DEFINE QL(TSM0025.SHAREDQ) +
CFSTRUCT('APPL2')
QSGDISP(SHARED)
```


Figure 5-9. Shared inbound channel and related queue configuration WM3121.1

Notes:

IBM MQ object definitions for a queue-sharing group are similar to non-shared definitions, except for:

- The inclusion of the QSGDISP parameter
- Use of the queue-sharing group name, SG05
- Use of the DVIPA and generic port in the CONNAME channel attribute

The triggered transmission queue can generate one message for each local initiation queue that is defined on a queue manager in the queue-sharing group that is associated with the triggered shared queue. Only one channel initiator processes the channel starts, and the other trigger messages adequately fail. The triggered channel is started in a load-balanced manner, as determined by IBM MQ.

The messages that are sent from MQ00 queue remote **TO.TSM0025.SHAREDQ** are directed to the queue-sharing group. Receiver channel MQ00.SG05, since it is receiving a connection that is directed at the queue-sharing group, is a shared channel. MQ00.SG05, defined with QSGDISP(GROUP), is “copied” to all queue managers in the queue-sharing group. The availability of the channel and the use of the DVIPA and generic port play a role if this channel requires peer recovery.

WebSphere Education IBM

Peer channel recovery view (1 of 2)

```
… on S101 … both MQ05 and MQ25 up, MQ05 has RCVR side of connection
17:43:11.55 STC03744 00000090  +CSQX500I MQ00 CSQXRCTL Channel MQ00.SG05
   started
… on S101 … stop MQ05 CHINIT … after MQ05 confirms channel failure, MQ00
   confirms channel start
17:44:41.73 STC04049 00000090  +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
… on S102, confirmation that MQ25 picked up RCVR side of connection
17:45:22.26 STC04051 00000090  +CSQX500I MQ25 CSQXRESP Channel MQ00.SG05
   started
                 292 00000090   connection 10.31.185.46
```


Figure 5-10. Peer channel recovery view (1 of 2) WM3121.1

Notes:

You now see how this configuration behaves in a peer channel recovery scenario. You start with both MQ05 and MQ25 queue managers and channel initiators active. S101 is LPAR1, which hosts MQ00 and MQ05. S102 is LPAR2, which hosts MQ25.

1. Channel MQ00.SG05 is started in queue manager MQ00 at 17:43:11.55. Not shown, it was confirmed that the MQ05 queue manager picked up the receiving part of the channel.
2. On S101, MQ05 channel initiator is stopped. MQ05 detects the error at 17:44:41.73, and confirms abnormal ending at 17:44:41.74.
3. At 17:45:2219 on S101, MQ00 announces that channel MQ00.SG05 started.
4. On S102, you look for messages for channel MQ00.SG05 and at 17:45:22.26 you see confirmation from queue manager MQ25 that it picked up its peer connection. You did not define channel MQ00.SG05 on MQ25, the definition was “copied”.
5. The MQ05 channel initiator is started. Channel MQ00.SG05 stays with the receiver end in queue manager MQ25. Both queue managers and channel initiators in the queue-sharing group are running.

WebSphere Education IBM

Peer channel recovery view (2 of 2)

```
...MQ05 CHINIT is restarted on S101. MQ25 CHINIT is stopped on S102
... on S101 the channel failure is noted
17:47:21.76 STC03744 00000090  +CSQX209E MQ00 CSQXRCTL Connection
   unexpectedly t
                    696 00000090   channel MQ00.SG05
                    696 00000090   connection 10.31.185.46
                    696 00000090   (queue manager SG05)
                    696 00000090   TRPTYPE=TCP RC=00000000
17:47:21.76 STC03744 00000090  +CSQX599E MQ00 CSQXRCTL Channel
   MQ00.SG05 ended abnormally
...
... on S101 MQ00 shows restart; MQ05 shows RCVR side of connection
17:47:31.76 STC03744 00000090  +CSQX500I MQ00 CSQXRCTL Channel
   MQ00.SG05 started
17:47:31.78 STC04055 00000090  +CSQX500I MQ05 CSQXRESP Channel
   MQ00.SG05 started
```


Figure 5-11. Peer channel recovery view (2 of 2) WM3121.1

Notes:

6. Not shown, on S102 the MQ25 channel initiator is stopped.
7. At 17:47:21:76, on S101, MQ00 informs of the channel failure. MQ25 channel initiator is stopped on S102. MQ05 channel initiator is started on S101.
8. At 17:47:31.76, MQ00 announces the start or restart of channel MQ00.SG05.
9. At 17:47:31.78, MQ05 confirms the restart of its receiver side of the channel also on S101.

Peer channel recovery took place as the channel initiator availability in the queue-sharing group was alternated.

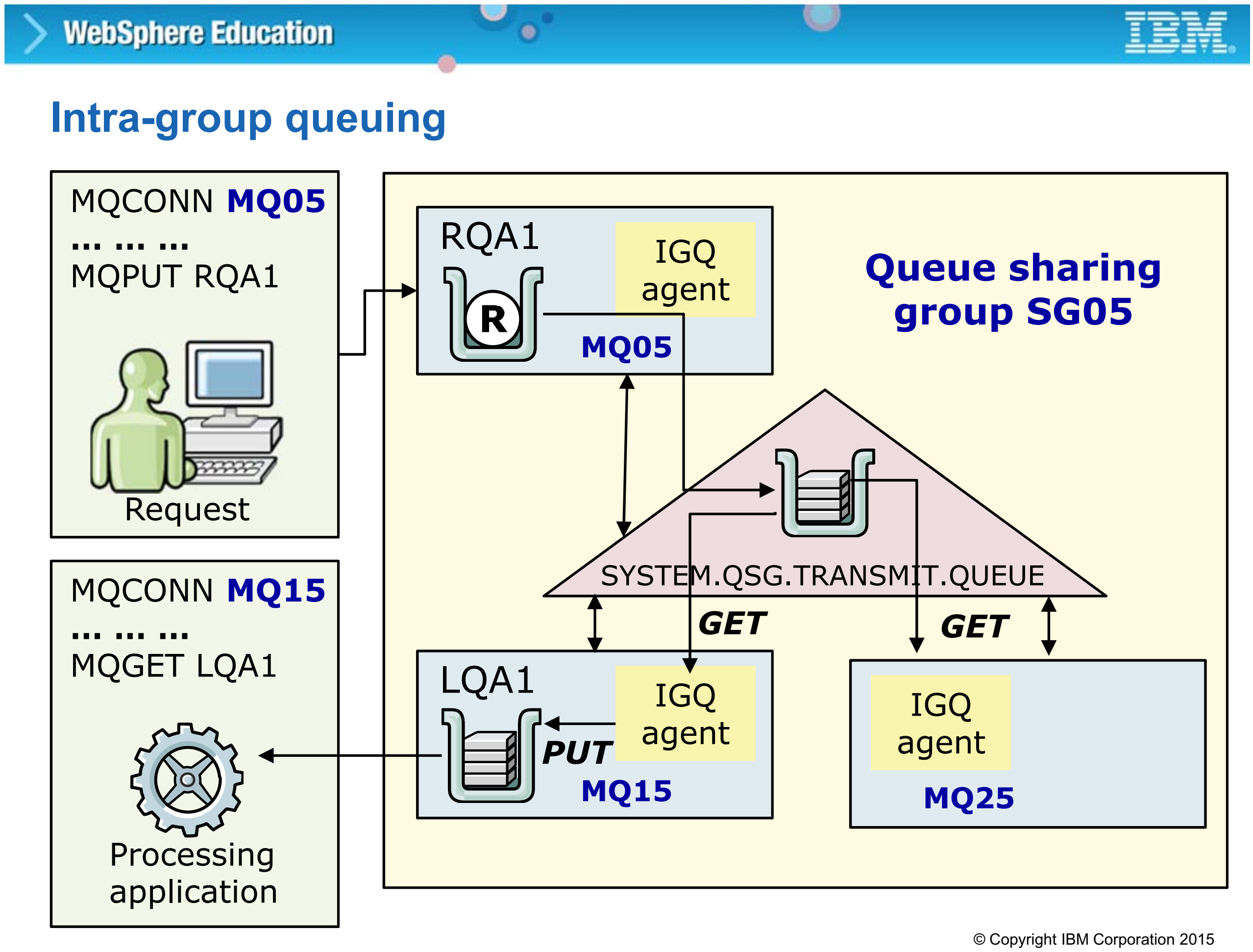


Figure 5-12. Intra-group queuing WM3121.1

Notes:

Intra-group queuing is a z/OS only IBM MQ function available to queue managers in a queue-sharing group. When messages are directed to queues to a queue manager member of a queue-sharing group, intra-group queuing can efficiently transfer smaller messages across the shared queue managers without the need to define channels.

There are two extra components to intra-group queuing:

- The intra-group queuing agent, or IGQ
- The local shared SYSTEM.QSG.TRANSMIT.QUEUE

The figure shows:

- Intra-group queuing agents that are run on three queue managers that compose queue-sharing group SG05: MQ05, MQ15, and MQ25.
- SG05 queue-sharing group transmission queue SYSTEM.QSG.TRANSMIT.QUEUE.
- Remote queue RQA1 defined on queue-sharing group SG05 queue manager MQ15, which resolves to local queue LQA1 on shared queue manager MQ15.
- Private (unshared) local queue LQA1, defined on SG05 member queue manager MQ15.

- An application that is connected to queue manager MQ05. The application might be a message channel agent. This application placed a moderately sized message to remote queue RQA1, which results in the message being placed in the SYSTEM.QSG.TRANSMIT.QUEUE.
- The intra-group queuing agent retrieves the message from the SYSTEM.QSG.TRANSMIT.QUEUE and places it in the target local queue, LQA1.
- An application that gets messages and is connected to queue manager MQ15 retrieves the message from the LQA1 queue.

When a message is put to a remote queue, the local queue manager decides whether intra-group queuing is used. Since the SYSTEM.QSG.TRANSMIT.QUEUE is in the coupling facility, the decision to use intra-group queuing is usually made according to the message’s size being of an adequate size to be placed in the coupling facility.

The key advantages of intra-group queuing are reduced IBM MQ objects to define and increased performance for messages less than 63 K.

Some caveats for intra-group queuing:

- Limitation on the size of the message that is processed to the maximum allowable length, minus the length of the transmission header (MQXQH).
- There is one intra-group queuing agent started per queue manager. Under normal situations, the agent is self-recoverable.
- When enabled, the intra-group agent starts when the queue manager starts. If the agent stops due to an error while the queue manager is running; it can be restarted again by re-enabling the agent with the **`ALTER QMQR IGQ(ENABLED)`** command without the need to recycle the queue manager.

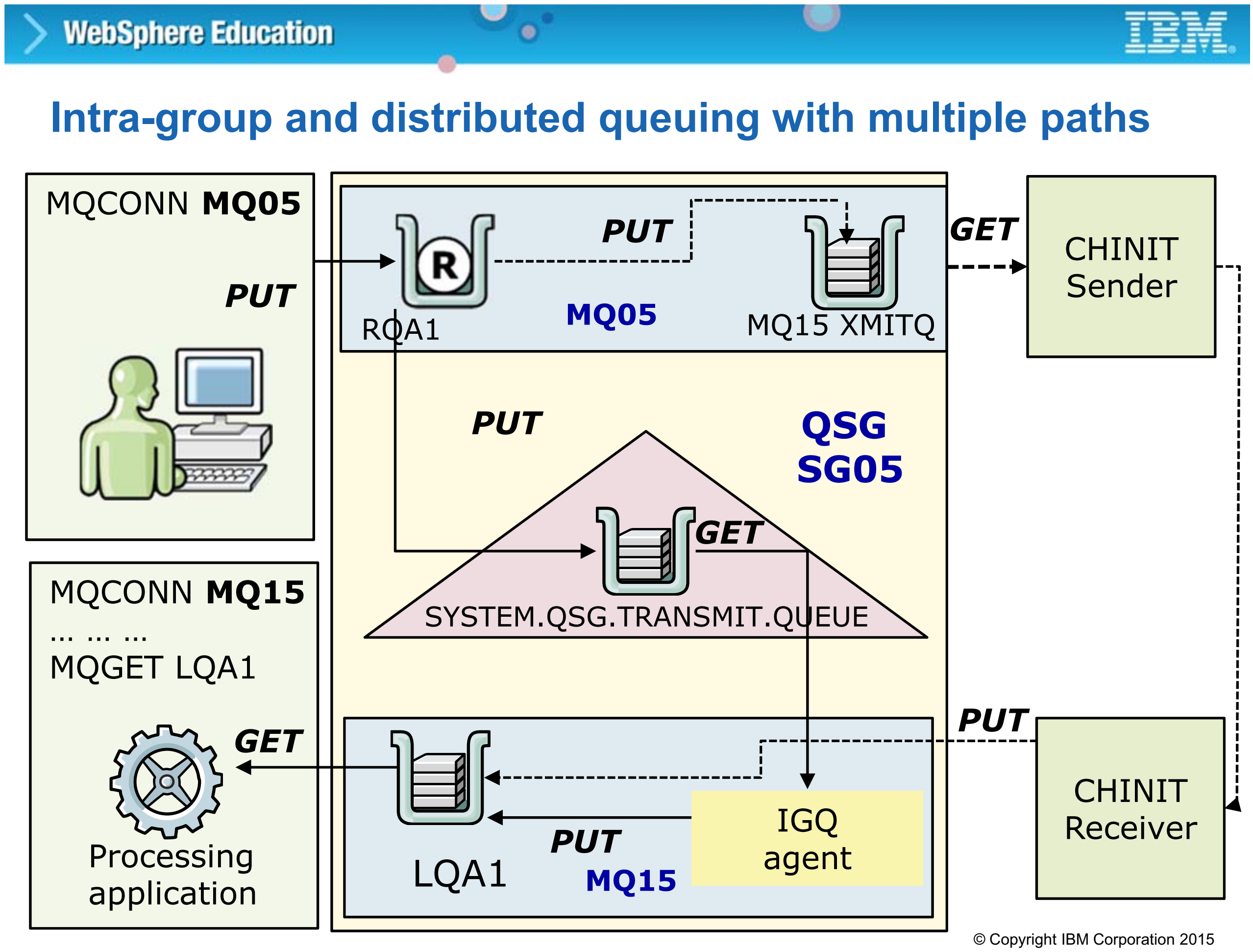


Figure 5-13. Intra-group and distributed queuing with multiple paths WM3121.1

Notes:

This figure expands to show a configuration to allow for larger messages that are mixed with small messages. The figure assumes intra-group queuing is enabled.

- The application connects to queue manager MQ05 and puts a message to remote queue RQA1.
- IBM MQ does name resolution for both message paths: distributed by using transmission queue MQ15 or intra-group by using SYSTEM.QSG.TRANSMIT.QUEUE.
- IBM MQ determines the most efficient way to route the message. If the message does not exceed the allowed size for shared queues, the path uses intra-group queuing. The solid arrows illustrate the intra-group queuing path. The dotted arrows indicate the distributed queuing path.
- After a message is placed in either the intra-group or the distributed transmission queue, no attempt is made to alter the path if a failure occurs. That is, if SYSTEM.QSG.TRANSMIT.QUEUE is full, no attempt is made to place the message to the MQ15 transmission queue.
- Applications do not need to know the path.
- The configuration in the figure is expected to provide faster message delivery for smaller messages.

WebSphere Education

Clusters and queue-sharing groups

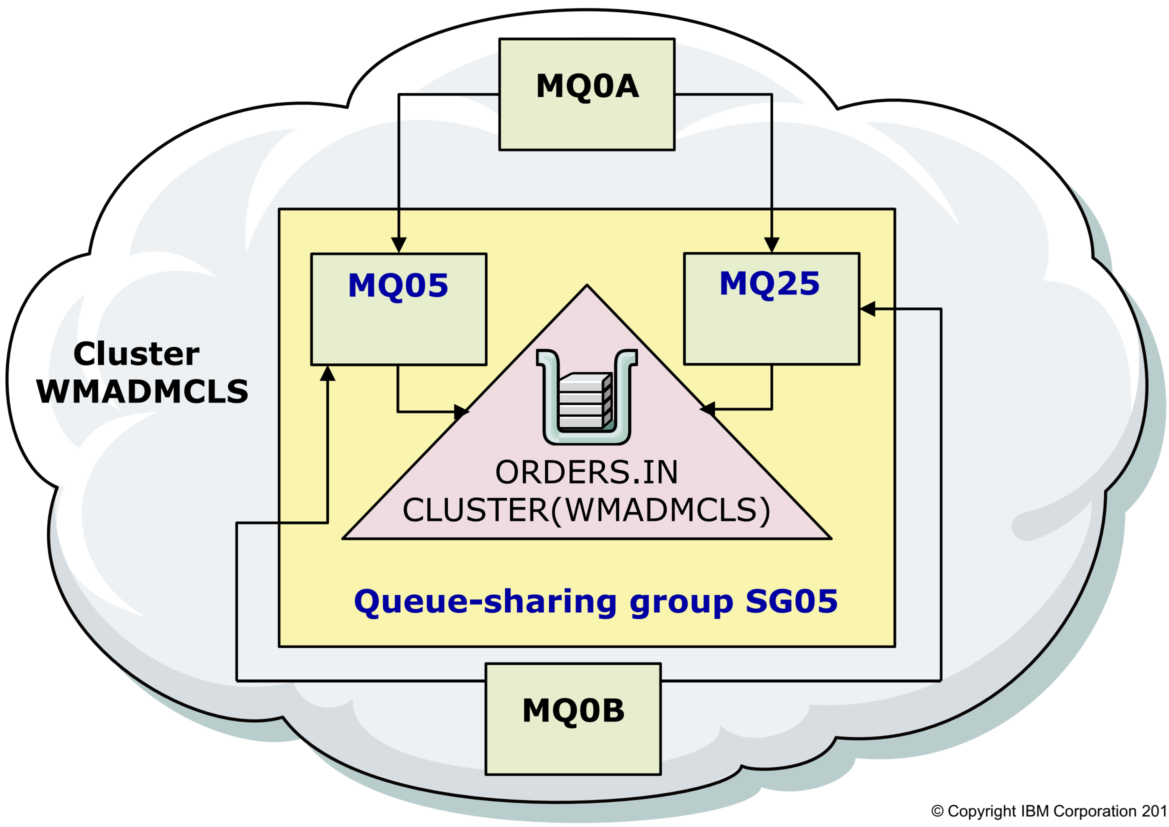


Figure 5-14. Clusters and queue-sharing groups WM3121.1

Notes:

Clusters can be combined with shared queues to process messages.

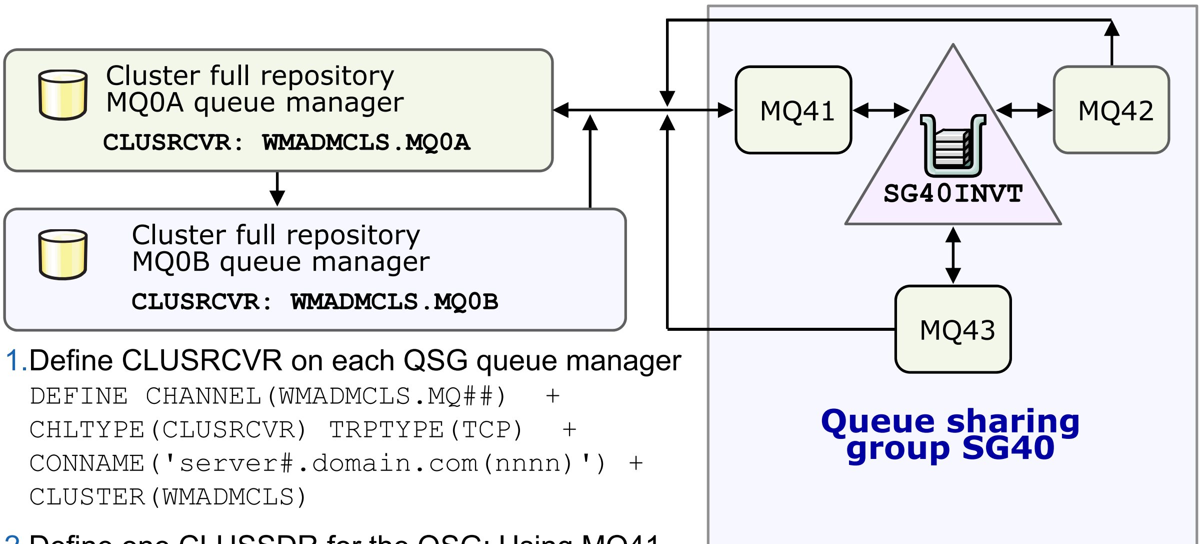
The figure shows:

- Queue-sharing group SG05 composed of queue managers MQ05 and MQ25.
- Cluster WMADMCLS consisting of queue managers MQ05, MQ25, MQ0A, and MQ0B.
- Clustered, shared queue ORDERS.IN.

Applications in the network see queue ORDERS.IN as hosted by all queue managers in the queue-sharing group. Members of the cluster can put messages to the queue ORDERS.IN through any queue manager in the queue-sharing group.

WebSphere Education IBM

Incorporating a queue-sharing group into an existing cluster



1.Define CLUSRCVR on each QSG queue manager

```
DEFINE CHANNEL(WMADMCLS.MQ##)  +
CHLTYPE(CLUSRCVR) TRPTYPE(TCP)  +
CONNAME('server#.domain.com(nnnn)') +
CLUSTER(WMADMCLS)
```

2.Define one CLUSSDR for the QSG: Using MQ41

```
DEFINE CHANNEL(WMADMCLS.MQ0A) +
CHLTYPE(CLUSSDR) TPTYPE(TCP) +
CONNAME('mvs++++.ilsvpn.ibm.com(1621)') +
CLUSTER(WMADMCLS) QSGDISP(GROUP)
```

3.Define the local queue on one of the QSG queue managers: Using MQ41

```
DEFINE QLOCAL(SG40INVT) CLUSTER(WMADMCLS) QSGDISP(SHARED) +
CFSTRUCT(APPL7)
```


Figure 5-15. Incorporating a queue-sharing group into an existing cluster WM3121.1

Notes:

This figure illustrates the process that is used to incorporate a queue-sharing group into an existing cluster and create a clustered shared queue.

- Cluster WMADMCLS has two full cluster repository queue managers: MQ0A and MQ0B.
- Queue-sharing group SG40 is composed of three queue managers: MQ41, MQ42, and MQ43.
- A highly available (shared) queue called SG40INV.

The steps to follow are:

1. Select one full repository in cluster WMADMCLS. You select MQ0A.
2. For each queue manager in the queue-sharing group, create one CLUSRCVR channel:
   a. On MQ41, define CLUSRCVR channel WMADMCLS.MQ41
   b. On MQ42, define CLUSRCVR channel WMADMCLS.MQ42
   c. On MQ43, define CLUSRCVR channel WMADMCLS.MQ43

3. Select one queue manager in the queue-sharing group to define the *shared* CLUSSDR channel for the queue-sharing group. In the figure, channel WMADMCLS.MQ0A is defined by setting the QSGDISP attribute to GROUP.
4. Issue the **`DIS CLUSQMGR(*)`** command from each queue manager in the queue-sharing group to confirm that the queue-sharing group was successfully added to the cluster.
5. Define queues as required.

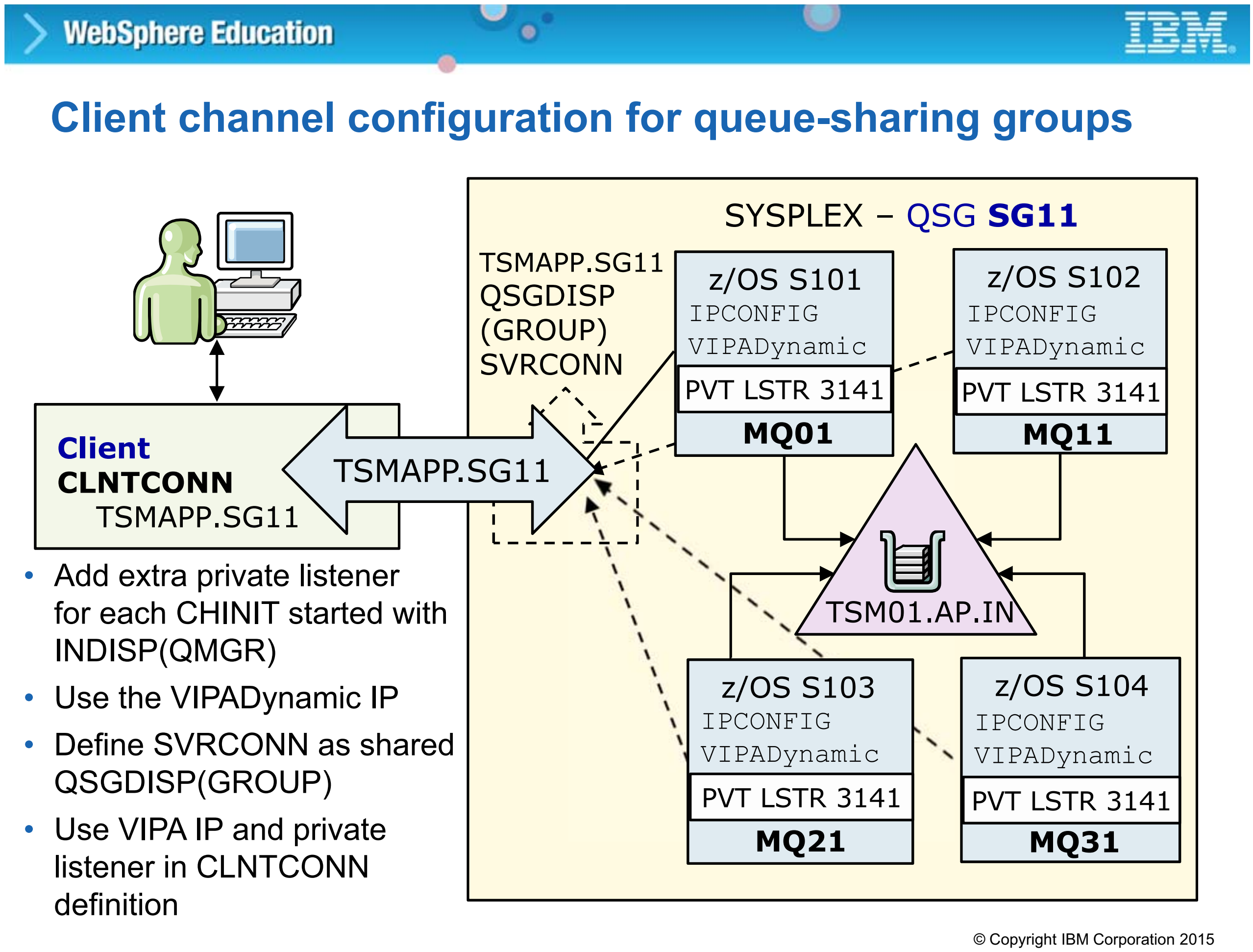


Figure 5-16. Client channel configuration for queue-sharing groups WM3121.1

Notes:

Client channels are stateless, do not require use of synchronization queues, and can use the load balancing benefits of queue-sharing groups without the need to keep status in DB2.

The efficient way to configure client channels for queue-sharing groups is:

- Create an extra private listener different that the listener used for distributed queuing, separate from the generic listener.
- Set up the listener to start with the channel initiator by using INDISP(QMGR). *This detail is important to prevent status that is recorded on DB2.*
- Define the SVRCONN channel with QSGDISP(GROUP).
- Use the DVIPA and extra private listener in the CLNTCONN definition.

Caveats

- In order for extended transactional clients or a client connection from WebSphere Application Server to use a QSG name to connect to the queue manager, the extended transactional client *must be* at IBM MQ V7.0.1 or later.

- When a connection is set up for WebSphere Application Server, always use the queue-sharing group name instead of the queue manager name.
- To have a group unit of recovery, you must ensure that you use shared resources exclusively in MQConnectionFactory if you use JMS under WebSphere Application Server.

Note

Although z/OS does not provide an external transactional client, external transactional clients can connect to a z/OS queue manager.

WebSphere Education

START CHANNEL attributes CHLDISP and CMDSCOPE

CHLDISP	CMDSCOPE blank or CMDSCOPE local queue manager	CMDSCOPE (queue manager name)	CMDSCOPE(*)
PRIVATE	• Start as a private channel on the local queue manager	Start as a private channel on the named queue manager	Start as a private channel in all queue managers
SHARED	• Shared SDR, RQSTR, and SVR channel: Start as a shared channel on most suitable queue manager • Shared RCVR and SVRCONN: Start as a shared channel on all active queue managers	Not allowed	Not allowed
FIXSHARED	• SDR, RQSTR, and SVR channel with non-blank CONNAME: Start as a shared channel on local queue manager	Start as a shared channel on the named queue manager	Not allowed

Figure 5-17. START CHANNEL attributes CHLDISP and CMDSCOPE WM3121.1

Notes:

The CHLDISP, and CMDSCOPE attributes might be used to start or obtain the status of a channel that is associated to queue-sharing groups. CHLDISP is the default channel disposition and CMDSCOPE controls which queue manager or queue managers in the queue-sharing group the command is directed to.

Channels are not considered SHARED by specifying QSGDISP(GROUP). QSGDISP(GROUP) creates the definition of the channel in DB2. To be considered PRIVATE or SHARED the channel must meet specific criteria.

Combined with the CMDSCOPE attribute, the CHLDISP attribute controls SHARED and PRIVATE channels. CHLDISP values are:

- DEFAULT: If the CHLDISP attribute is omitted, the DEFCDISP value of the channel definition is used as the default.
- FIXSHARED: Channels started with FIXSHARED are committed to a specific queue manager. Channels started with FIXSHARED do not participate in peer channel recovery.
- SHARED:

  - A receiving channel is shared if it was started in response to a sending channel directed to the queue-sharing group.
  - A sending channel is shared if its transmission queue has a QSGDISP of SHARED.
- PRIVATE:
  - A receiving channel is private if it was started in response to a channel directed at the queue manager.
  - A sending channel is private if its transmission queue is other than SHARED.

WebSphere Education IBM

DIS CHSTATUS SHORT attributes CHLDISP and CMDSCOPE

CHLDISP	CMDSCOPE blank or CMDSCOPE local queue manager	CMDSCOPE (queue manager name)	CMDSCOPE(*)
PRIVATE	STATUS and short status for current private channels on the local queue manager	STATUS and short status for current private channels on the named queue manager	STATUS and short status for current private channels on all active queue managers
SHARED	STATUS and short status for current shared channels on all active queue managers in the queue-sharing group	Not allowed	Not allowed
FIXSHARED	STATUS and short status for current private channels on the local queue manager and current shared channels in the queue-sharing group	STATUS and short status for current private channels on the named queue manager	STATUS and short status for current private, and shared, channels on all active queue managers in the queue-sharing group

Figure 5-18. DIS CHSTATUS SHORT attributes CHLDISP and CMDSCOPE WM3121.1

Notes:

CHLDISP and CMDSCOPE also apply to the display of the channel status. However, the DIS CHSTATUS command works in a similar manner as with non-shared queue managers by adding the “SHORT” attribute, the command displays the status kept in the DB2 status table.

WebSphere Education

SET CHLAUTH and the CMDSCOPE attribute

CMDSCOPE attribute implications exclusive to z/OS queue managers members of a queue-sharing group	
CMDSCOPE not specified; left blank	• Rule runs on the queue manager on which it was entered • Blank is the initial default value
CMDSCOPE is queue manager name	• Rule runs on the queue manager specified • The queue manager that is specified can be different than the queue manager where it was entered
CMDSCOPE is '*'	• Command runs in local queue manager, and in all queue managers active in the queue-sharing group • Same effect as if you enter the command in each active queue manager in the queue-sharing group

Figure 5-19. SET CHLAUTH and the CMDSCOPE attribute WM3121.1

Notes:

The CMDSCOPE attribute also applies to channel authentication rules.

- A rule with a blank CMDSCOPE value runs on the queue manager on which it was created.
- Rules can be set in one queue manager but set to be effective in another active queue manager in the queue-sharing group.
- To have the rule that is run in all queue managers in the queue-sharing group, complete one of the following options:
  - Use CMDSCOPE(‘*’) when the rule is first set
  - Define the rule in each queue manager member of the queue-sharing group

WebSphere Education IBM

Unit summary

- Explain the advantages of intra-group and distributed queuing with queue-sharing groups
- Describe how to set up peer channel recovery
- Explain intra-group queuing
- Explore various queue-sharing group configurations
- Describe how to add a queue-sharing group to an existing cluster
- Describe the optimal client connection configuration
- Explain queue-sharing group differences with START CHANNEL and DISPLAY CHSTATUS commands

Figure 5-20. Unit summary WM3121.1

Notes:

Recap of concepts in this unit:

A channel definition that uses QSGDISP(GROUP):

- Creates the channel definition on DB2
- Does not make the channel a shared channel

A shared channel:

- Inbound: Starts in response to a START CHANNEL directed at the queue-sharing group
- Inbound: Is defined as global (QSGDISP GROUP) or private (QSGDISP QMGR), if IBM MQ has access to a definition for the channel
- Outbound: Uses a shared transmission queue
- Started by using CHLDISP option

Synchronization information for a channel is:

- Kept on SYSTEM.QSG.CHANNEL.SYNCQ for shared channels

- Kept on SYSTEM.CHANNEL.SYNCQ for non-shared channels

The DB2 channel status table

- Tracks the status of channels that are started with the generic port
- Is used for channel peer recovery

WebSphere Education IBM

Checkpoint questions (1 of 3)

1. Select the correct answers: Some of the tasks that are required to implement peer shared channel recovery are:
   a. Create Dynamic VIPA IP and generic listener definitions
   b. Start the generic listener
   c. Define the shared queues
   d. Start the channel

2. True or false. You define a sender channel by using QSGDISP(GROUP), however, the QSGDISP alone does not make the channel shared.

Figure 5-21. Checkpoint questions (1 of 3) WM3121.1

Notes:

Write your answers here:

1.

2.

WebSphere Education IBM

Checkpoint questions (2 of 3)

3. Select the statements that are true for intra-group queuing:
   a. Intra-group queuing does not use the channel initiator
   b. The queue manager uses SYSTEM.QSG.TRASNMIT.QUEUE to process all messages
   c. Intra-group queuing uses the intra-group queuing agent to move messages
   d. Intra-group queuing only works with shared queues

4. True or false: DIS CHSTATUS(*) SHORT obtains the status information of the channel from the DB2 table.

Figure 5-22. Checkpoint questions (2 of 3) WM3121.1

Notes:

Write your answers here:

3.

4.

WebSphere Education

Checkpoint questions (3 of 3)

5. Which of the steps that are listed are required to incorporate a queue-sharing group into an existing cluster?
    a. Define a CLUSRCVR on each queue manager in the queue-sharing group
    b. Define one CLUSSDR for the queue-sharing group
    c. Defined a shared local queue in one of the queue managers in the queue-sharing group
    d. All of the above
    e. a and b

Figure 5-23. Checkpoint questions (3 of 3) WM3121.1

Notes:

Write your answers here:

5.

WebSphere Education

Checkpoint answers (1 of 3)

1. Select the correct answers: Some of the tasks that are required to implement peer shared channel recovery are:
   a. Create Dynamic VIPA IP and generic listener definitions
   b. Start the generic listener
   c. Define the clustered queues
   d. Start the channel

   Answer: a, b, and d. Clustered queues are not required for queue-sharing groups.

2. True or false. You define a sender channel by using QSGDISP(GROUP), however, the QSGDISP alone does not make the channel shared.

   Answer: True. QSGDISP(GROUP) adds the channel definition to DB2. A sender channel is started with CHDISP(SHARED) and uses a shared transmit queue. An inbound channel can be shared if it was started by using a CONNAME with the dynamic IP and generic listener.

Figure 5-24. Checkpoint answers (1 of 3) WM3121.1

Notes:

WebSphere Education IBM

Checkpoint answers (2 of 3)

3. Select the statements that are true for intra-group queuing:
   a. Intra-group queuing does not use the channel initiator
   b. The queue manager uses SYSTEM.QSG.TRANSMIT.QUEUE to process all messages
   c. Intra-group queuing uses the intra-group queuing agent to move messages
   d. Intra-group queuing only works with shared queues

   Answer: a and c. b is incorrect because large messages (>63 KB or larger than the SYSTEM.QSG.TRANSMIT.QUEUE) are not processed by intra-group queuing. d is incorrect because intra-group queuing also works with non-shared queues.

3. True or false: DIS CHSTATUS(*) SHORT obtains the status information of the channel from the DB2 table.

   Answer: True

Figure 5-25. Checkpoint answers (2 of 3) WM3121.1

Notes:

WebSphere Education

Checkpoint answers (3 of 3)

5. Which of the steps that are listed are required to incorporate a queue-sharing group into an existing cluster?
   a. Define a CLUSRCVR on each queue manager in the queue-sharing group
   b. Define one CLUSSDR for the queue-sharing group
   c. Define a shared local queue in one of the queue managers in the queue-sharing group
   d. All of the above
   e. a and b
   Answer: e, a, and b. The queue-sharing group might be sending messages to clustered queues outside the queue-sharing group, so definition of cluster queues is not a requirement for the queue-sharing group to join a cluster.

Figure 5-26. Checkpoint answers (3 of 3) WM3121.1

Notes:

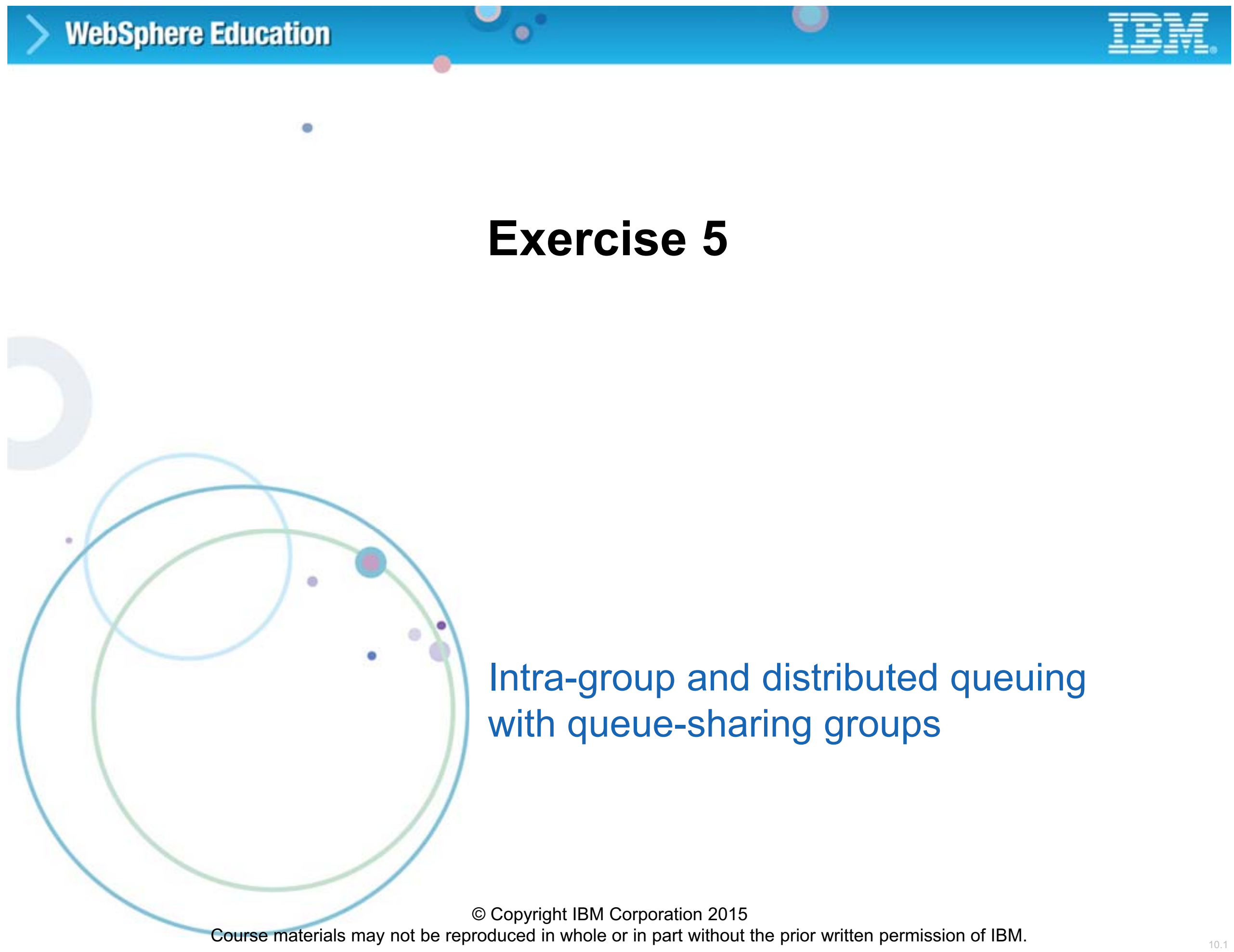


Figure 5-27. Exercise 5 WM3121.1

Notes:

WebSphere Education

Exercise objectives

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

Figure 5-28. Exercise objectives WM3121.1

Notes:

Unit 6. Eight-byte relative byte address (RBA) and buffers above the 2-GB line

What this unit is about

This unit describes the IBM MQ V8 capabilities for z/OS 8-byte relative byte address (RBA) and buffers above the 2-GB line.

What you should be able to do

After completing this unit, you should be able to:

- Explain how to configure buffers above the line
- Explain how to implement 8-byte RBA in a new or existing queue manager
- Describe the considerations to implement 8-byte RBA for queue managers in a queue-sharing group

How you will check your progress

Checkpoint questions

Lab exercises

WebSphere Education IBM.

Unit objectives

- Explain how to configure buffers above the line
- Explain how to implement 8-byte RBA in a new or existing queue manager
- Describe the considerations to implement 8-byte RBA for queue managers in a queue-sharing group

Figure 6-1. Unit objectives WM3121.1

Notes:

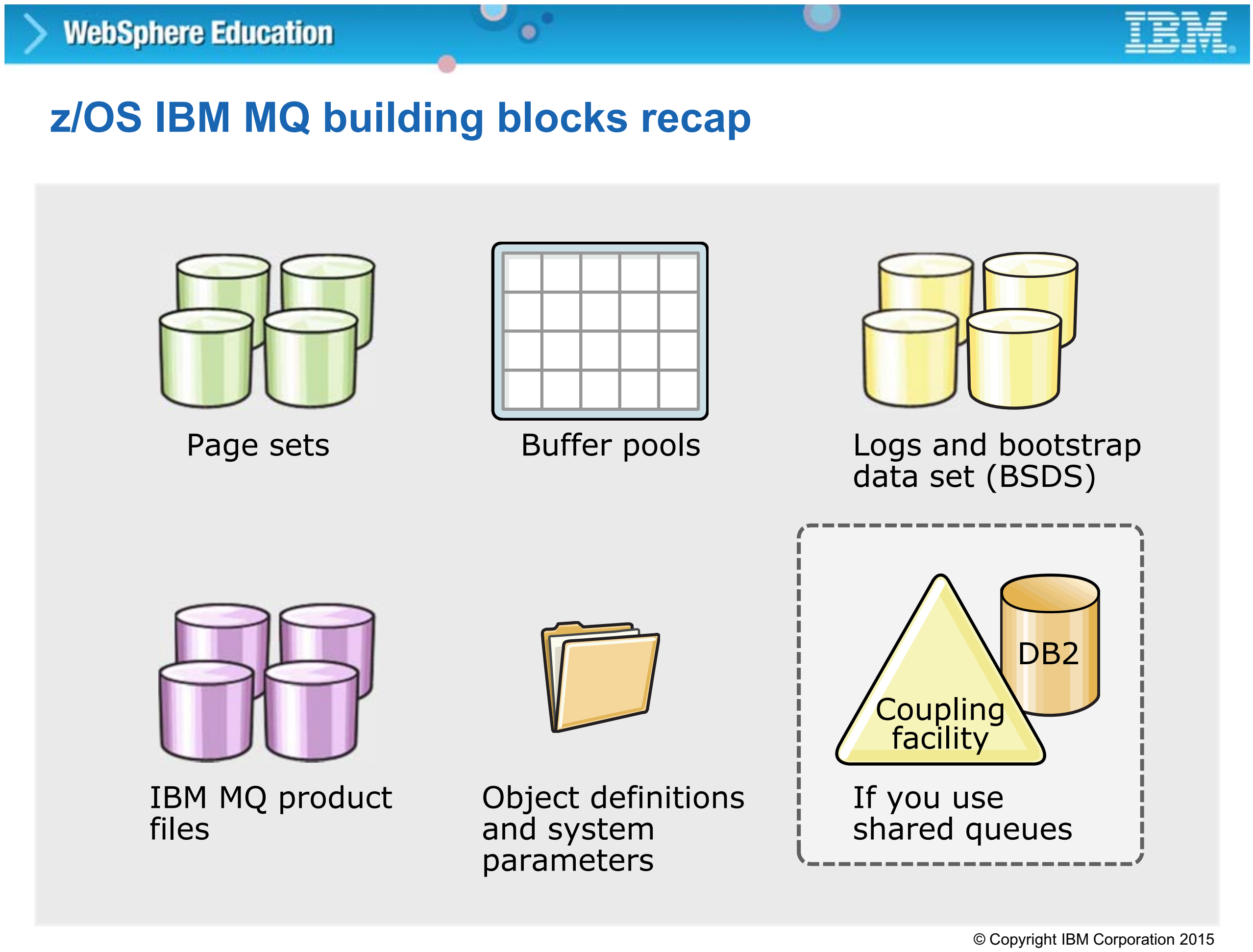


Figure 6-2. z/OS IBM MQ building blocks recap WM3121.1

Notes:

In the first unit, you reviewed the IBM MQ for z/OS building blocks. What components must be defined to configure a z/OS queue manager.

With IBM MQ V8, two new features were implemented of impact to these building blocks.

- **Buffer pools above the 2-GB line**. IBM MQ is not meant to be used as a database. In an optimal environment, messages are placed in the buffers, and are removed before they are written to a page set. However, when buffers start to fill messages are written to a page set. The implementation of buffer pools above the bar helps keep more messages in buffers, which should allow more messages to remain in the buffer pools.
- **Eight-byte record-byte address (RBA)**. Each byte in the logs is addressable by its RBA. Before IBM MQ for z/OS V8, if the RBA reached the FFF800000000 limit, you incurred a queue manager outage. Eight-byte RBA mitigates outages by providing a bigger addressable range.

You look at both of these features in this unit before the accounting and statistics topic.

Queues, buffer pools, and page sets: Message path view

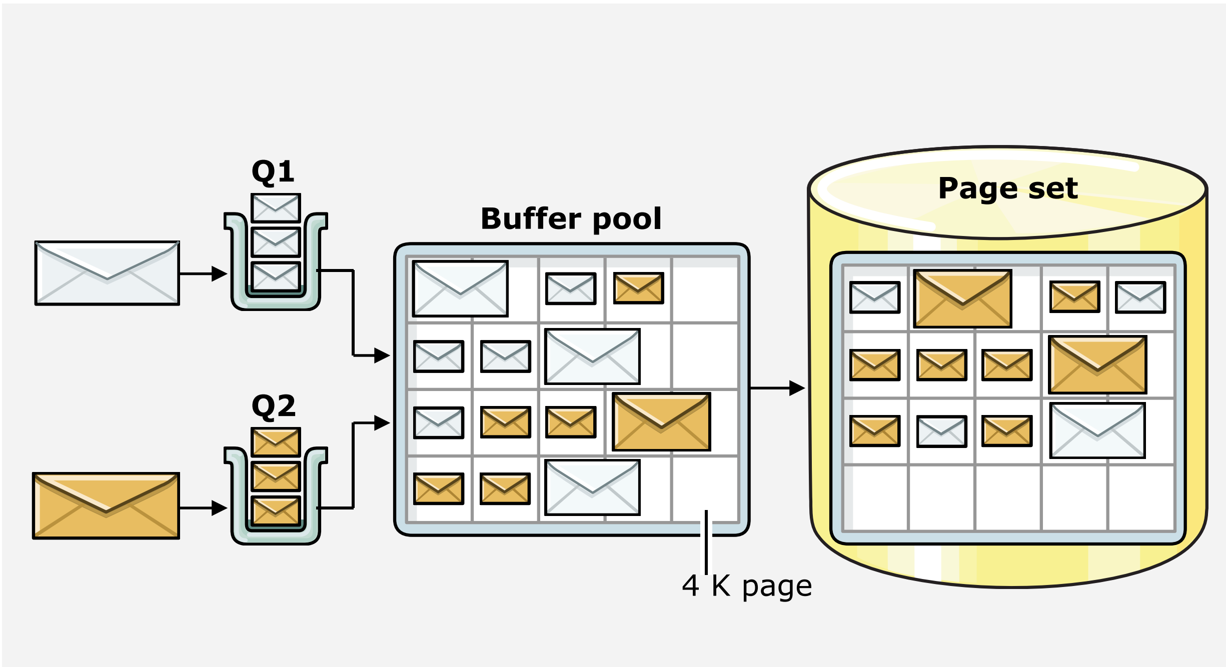


Figure 6-3. Queues, buffer pools, and page sets: Message path view WM3121.1

Notes:

Queues that hold messages are configured to use one specific page set to store its messages. One or more page sets, can be configured to use a specific buffer pool as the initial place to stage, or hold the messages.

The figure shows the path when a message is put. The definition associations, starting from buffer to queue, are: **BUFFPOOL=>PSID=>STGCLASS=>QUEUE**.

A message is initially written to the buffer pool. If the buffer pool reaches certain thresholds, or the message is in the buffer for a number of checkpoints, the message is written to the page set to free the buffers.

When a message is retrieved, if the message is not in the buffer pool, it needs to be written back from the page set for the buffer pool before the message can be retrieved.

Before IBM MQ V8 for z/OS, there were some challenges to overcome:

- After the queue manager code and other data are accounted for, there is at most, 1.6 GB left over for buffer pools.
- While a maximum of 100 page sets is supported, there are a maximum of 16 buffer pools available. This forces sharing of buffer pools for some of the page sets.

- When messages are placed and retrieved from a buffer pool, the access takes place at *memory speed*, but the same placement and retrieval from a page set occurs at *disk speed*.
- As a number of applications put and get large numbers of messages to the same buffer pool, significant time is spent shuffling messages from the buffer pool to the page set and back. This process takes a toll on the queue manager.
- Much time is spent tuning the buffer pools, moving page sets to buffer pools, and queues to page sets.

WebSphere Education

Buffer pools above the bar

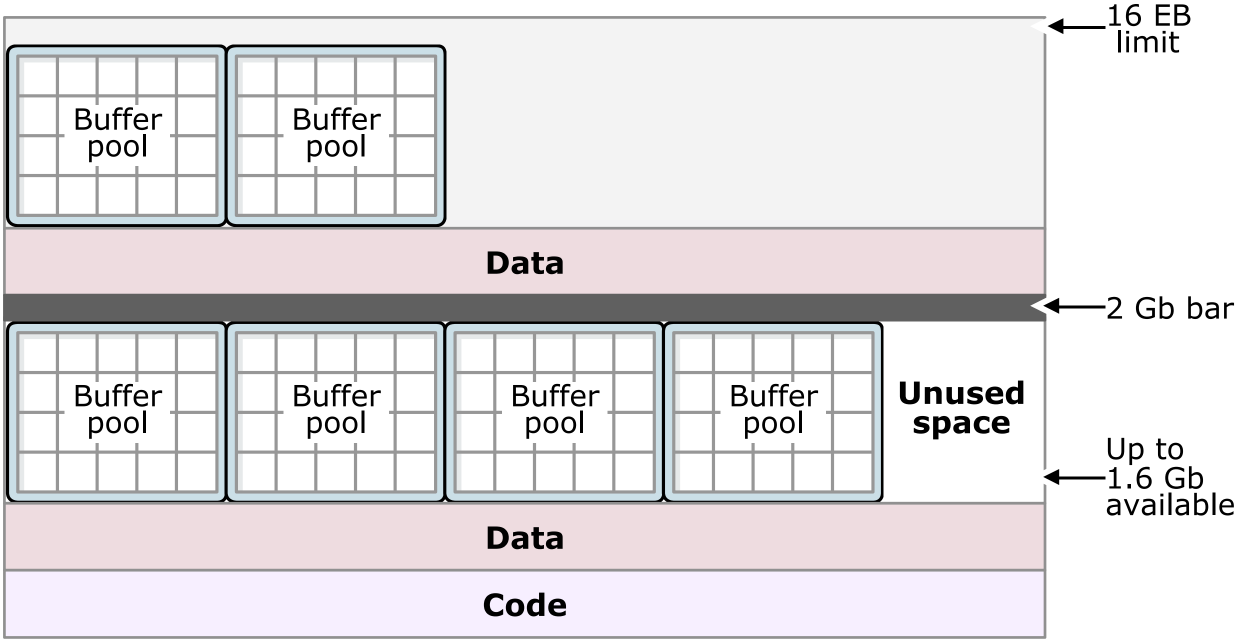


Figure 6-4. Buffer pools above the bar WM3121.1

Notes:

IBM MQ V8 introduced the ability to have buffer pools above the bar. *“The bar” is the name or accepted term that is given to the virtual line that separates storage below the 2 GB addresses from storage above the 2 GB addresses.*

Buffer pools above the bar mitigate storage constraints by the ability to move some buffer pools out of 31-bit storage:

- Buffer pools above the bar might use up to 16 EB. *16 EB s 1 billion GB.*
- The sizes of the buffer pools that are moved above the 2 GB bar can be increased.
- Number of 4 KB buffers per buffer pool.
  - BELOW the bar: 100 - 500,000
  - ABOVE the bar: 100 - 999,999,999
- Maximum number of buffer pools increased to 100 from 16, which allows a 1:1 mapping of page sets to buffer pools.

WebSphere Education

Buffer pools above the bar considerations

- To enable new IBM MQ V8 functions such as storage above the bar, the **OPMODE** attribute of the **CSQ6SYSP** macro needs to be set to **OPMODE(NEWFUNC,800)**
- New LOCATION buffer pool setting for IBM MQ V8 and later queue managers
  - BELOW: (Default) Buffer pool that is located below the 2 GB bar in 31-bit storage
  - ABOVE: Buffer pool is located above the bar in 64-bit storage
  - Can be changed dynamically
- Up to 100 buffer pools supported
- PAGECLAS(FIXED4KB) available for BUFFERS(ABOVE)

```
DEFINE BUFFPOOL( 4 ) BUFFERS( 20000 ) LOCATION( BELOW )
ALTER BUFFPOOL( 4 ) LOCATION( ABOVE )
```

- SMF 115 subtype accommodates the extended buffer statistics records

Figure 6-5. Buffer pools above the bar considerations WM3121.1

Notes:

Before you configure buffer pools above the bar:

- Ensure that the CSQ6SYSP macro has OPMODE(NEWFUNC,800).
- The record with the **PGM= EXEC CSQYASCP** in your master procedure JCL should have MEMLIMT adjusted to use buffers above the bar.
- MEMLIMIT guidance is 2 GB plus the size of the buffer pools above the bar, rounded up to the nearest GB.

DEFINE or ALTER BUFFPOOL changes

- LOCATION: BELOW, or ABOVE. This value can be altered dynamically.
- BUFFERS: Depends on LOCATION.
- The buffer pool identifier can be 0 - 99.
- PAGECLAS: Determines whether to permanently fix the page in storage.
  - 4KB is the default. Buffers with LOCATION(BELOW) must use this option.
  - FIXED4KB allowed if LOCATION(ABOVE).

You can follow the progress of these changes in the queue manager output or the system log.

Warning

Specifying PAGECLAS(FIXED4KB)for buffers above the bar might result in significant performance improvements for buffer pools that experience frequent reads and writes.

The storage is fixed for the life of the queue manager. PAGECLAS(FIXED4KB) should *not* be used without adequate analysis, justification, and planning. You should ensure that there is sufficient real storage before using PAGECLAS(FIXED4KB).

Adverse effects might be:

- Other address spaces might be impacted
- Queue manager might fail to start

A change to use PAGECLAS(FIXED4KB) requires a restart of the queue manager to take effect. If you dynamically change PAGECLAS to FIXED4KB, message CSQP062I is displayed:

```
CSQP062I MQ04 Buffer pool 3 PAGECLAS changed, restart required to take effect
```

A change to use LOCATION(ABOVE) with PAGECLAS(4KB) does *not* require a restart of the queue manager.

WebSphere Education IBM

Check buffer definitions with the DIS USAGE command

```
MQ04 DIS USAGE
CSQI010I MQ04 Page set usage ... 244
Page  Buffer    Total   Unused Persistent NonPersist Expansion
 set    pool    pages    pages data pages data pages     count
 _ 0       0      322      259         63          0 USER     0
 … … … … …
  End of page set report
CSQI065I MQ04 Buffer pool attributes ... 245
  Buffer  Available  Stealable   Stealable  Page       Location
    pool    buffers    buffers  percentage  class
_      0      50000      49965          99  4KB        BELOW
_      1      20000      20000         100  4KB        ABOVE
_      2      50000      49994          99  4KB        BELOW
_      3        100         95          95  FIXED4KB   ABOVE
 End of buffer pool attributes
```


Figure 6-6. Check buffer definitions with the DIS USAGE command WM3121.1

Notes:

Use the `DIS USAGE` command to check the buffer pools.

If you change `PAGECLAS` to `FIXED4KB`, you do not see the change until after you restart the queue manager.

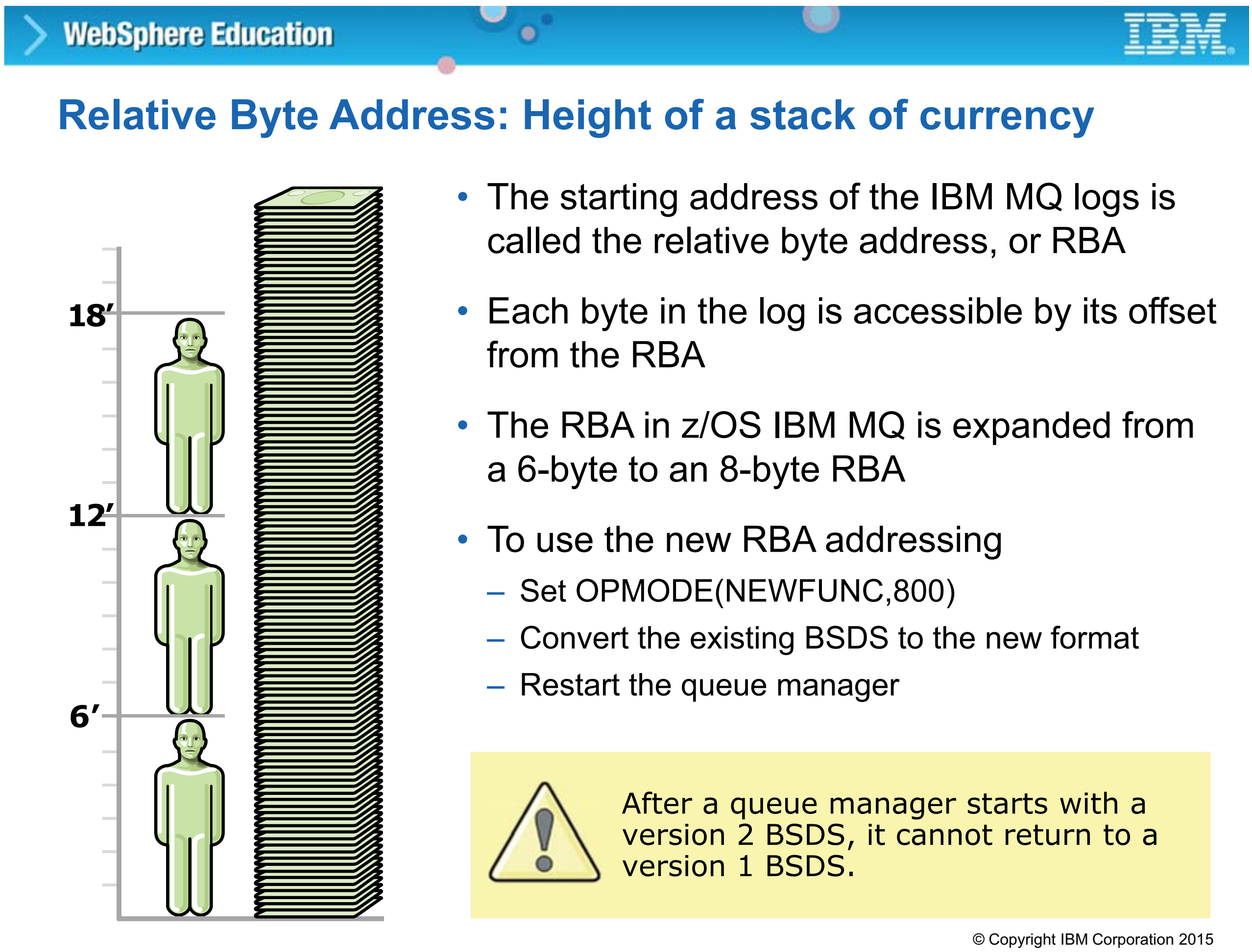


Figure 6-7. Relative Byte Address: Height of a stack of currency WM3121.1

Notes:

Each byte in the IBM MQ for z/OS logs is addressed by a relative byte address, or RBA. Queue managers before V8 used a 6-byte RBA, which allowed queue managers to log up to approximately 255 TB before they exceeded the 6-byte addressing limit. An application that logs 100 MB per second might reach this limit in 1 month.

The RBA constraint can be visualized by looking at the stack of dollar bills in the figure. The 6-byte RBA is represented by a single dollar bill out of the stack illustrated.

If a queue manager reaches the end of the log RBA, the queue manager stops and an outage is incurred. The queue manager then needs a cold start and there is the potential to lose persistent data.

To mitigate the loss of persistent data, administrators require a process to reset the log RBA at regular intervals. This process also requires an outage. Resetting the log RBA is not trivial, as it requires:

- Manually checking and resolving any unresolved units of work.
- Stopping the queue manager.

- If the queue manager is part of a queue-sharing group, a backup of all structures (BACKUP CFSTRUCT) on other queue managers for all members of a queue-sharing group is required.
- Define BSDS and logs.
- Run the `CSQUTIL RESETPAGE` utility.
- Start the queue manager.

The ability to use an 8-byte RBA significantly mitigates the need to reset the log RBA by increasing the maximum 6-byte RBA limit by 65,536 times.

For a visual comparison of this change, compare the depth of a single dollar bill to a stack of dollar bills over 18 feet high. It would take the same organization with a log rate of 100 MB per second approximately 5,578 years to reach the limit of 8-byte RBA addressability.

WebSphere Education IBM

Converting to 8-byte RBA

For each BSDS

1. Stop the queue manager
2. Create:
   – Version 1 backup (V1BK) BSDS
   – NEW BSDS
3. Back up BSDS to V1BK
4. Run CSQJUCNV
   – Active BSDS as input, SYSUT1
   – NEW BSDS as output, SYSUT3
5. Rename active to OLD
6. Rename NEW to active
7. If queue-sharing group
   a. All have OPMODE(NEWFUNC)
   b. Member queue managers can stay while others convert

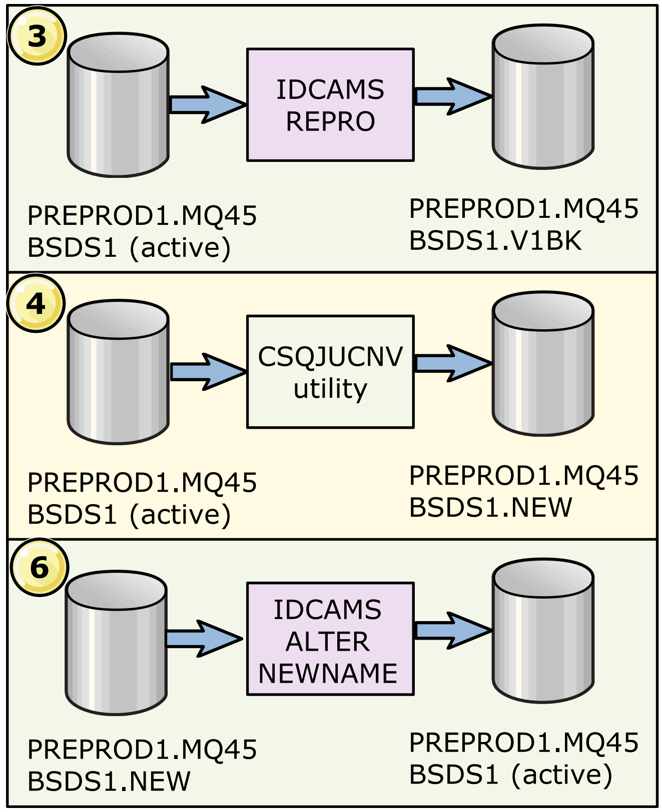


Figure 6-8. Converting to 8-byte RBA WM3121.1

Notes:

The process to convert a BSDS version 1 to version 2 BSDS should be simple.

- You must preserve the original BSDS in case there is a problem during the conversion.
- You create the BSDS, run the utility, and rename the files.

A safe process to follow is suggested in the text portion of the slide. The extent to which you alter the process in the slide is directly proportional to the importance of the queue manager, or to your ability to absorb the risk of corrupting the original BSDS.

- If you are working with a production queue manager, you might insist on a separate backup for each version 1 BSDS as shown in the text portion of the slide. Do not use this backup for any process or for renaming.
- After you stop the queue manager, there is nothing to prevent you from logically developing your own process for the conversion, such as:
  - Rename the active BSDS to OLD BSDS, such as DEV.BSDS1.OLD
  - Redefine the new *active* BSDS as named in the **MQ##MSTR** JCL, which becomes the //SYSUT3 output
  - Run the CSQJUCNV utility by using the renamed OLD BSDS as the //SYSUT1 BSDS

In a busy environment, interruptions result in losing track of the process. If you have a good backup of the BSDS, you can always start from scratch.

For a production queue manager, a separate backup of the version 1 BSDS should be a requirement.

In this slide, the name of the backup data set ends in V1BK to potentially mitigate the name of a regularly scheduled BSDS backup. The name that is chosen for the version 1 backup is up to the site standards.

When working with queue manager members of a queue-sharing group:

- All queue managers must have the parameter module set to `OPMODE(NEWFUNC)`.
- If all members have `OPMODE(NEWFUNC)`, members of the queue-sharing group can remain active while one member is undergoing the BSDS conversion.

WebSphere Education IBM

Sample CSQJUCNV JCL

- If the queue manager is not a member of a queue-sharing group, use **PARM=('NOQSG')**
- If the queue manager is a member of a queue-sharing group
  - Use **PARM=('INQSG,qsgname,dsgname,db2ssid')**
  - The utility checks that either the queue manager was added at IBM MQ V8, or that it was started one time with **OPMODE(NEWFUNC,800)**

```
//CONVERT   EXEC PGM=CSQJUCNV,REGION=32M,
//               PARM=('NOQSG')
//STEPLIB   DD DSN=SYS2.MQ8000.SCSQAUTH,DISP=SHR
//          DD DSN=SYS2.MQ8000.SCSQANLE,DISP=SHR
//SYSPRINT DD SYSOUT=*
//SYSUT1    DD DSN=PREPROD1.MQ45.BSDS1,DISP=SHR
//SYSUT3    DD DSN=PREPROD1.MQ45.BSDS1.NEW,DISP=OLD
```


Figure 6-9. Sample CSQJUCNV JCL WM3121.1

Notes:

The JCL to run the CSQJUCNV utility, found in the HLQ.SCSQPROC data set, is **CSQ4BCNV**. Other than the JOB card specifications, the changes are:

- **PARM** statement, depending on whether the queue manager is a member of a queue-sharing group:
  - Not a member of a queue-sharing group: PARM=('NOQSG')where NOQSG is a constant.
  - Member of a queue-sharing group: PARM=('INQSG,qsgname,dsgname,db2ssid') where INQSG is a constant, followed by the queue-sharing group name, data sharing group name, and DB2 sub system ID.
- The SYSPRINT statement is required.
- SYSUT1 and SYSUT2 are used for the input, version 1 BSDS data sets.
- SYSYT3 and SYSUT3 are used for the corresponding output, version 2 BSDS data sets.
- If you use one BSDS, then you use SYSUT1 for the input and SYSUT3 for the output.

Depending on your preference, you might want to have a separate job for each BSDS, versus converting two BSDS data sets on the same job.

Confirm conversion to 8-byte RBA: CSQJ034I

```
CSQJ127I MQ00 SYSTEM TIME STAMP FOR BSDS=2014-09-18 12:41:05.31
CSQJ001I MQ00 CURRENT COPY 1 ACTIVE LOG DATA SET IS  933
DSNAME=MQ00.LOGCOPY1.DS01, STARTRBA=0000000000000000
ENDRBA=000000000437FFF
CSQJ001I MQ00 CURRENT COPY 2 ACTIVE LOG DATA SET IS  934
DSNAME=MQ00.LOGCOPY2.DS01, STARTRBA=0000000000000000
ENDRBA=000000000437FFF
CSQJ099I MQ00 LOG RECORDING TO COMMENCE WITH  935
STARTRBA=00000000005E000
CSQJ034I MQ00 CSQJW007 END OF LOG RBA RANGE IS FFFFFFFFFFFFFFFF
CSQP007I MQ00 Page set 0 uses buffer pool 0
… … … … …
CSQR001I MQ00 RESTART INITIATED
CSQR003I MQ00 RESTART - PRIOR CHECKPOINT RBA=000000000005C946
CSQR004I MQ00 RESTART - UR COUNTS -  946
IN COMMIT=0, INDOUBT=0, INFLIGHT=0, IN BACKOUT=0
CSQI049I MQ00 Page set 0 has media recovery  947
RBA=0000000000052A51, checkpoint RBA=0000000000052A51
… … … … …
```


Figure 6-10. Confirm conversion to 8-byte RBA: CSQJ034I WM3121.1

Notes:

To confirm that your conversion completed uneventfully, look for message `CSQJ034I` in the queue manager start messages. The second part of the message should indicate that the end of the log RBA range is ‘**FFFFFFFFFFFFFFFF**’. Sixteen “foxes”, no zeros.

WebSphere Education

Unit summary

- Explain how to configure buffers above the line
- Explain how to implement 8-byte RBA in a new or existing queue manager
- Describe the considerations to implement 8-byte RBA for queue managers in a queue-sharing group

Figure 6-11. Unit summary WM3121.1

Notes:

WebSphere Education

Checkpoint questions (1 of 2)

1. Select the best answer. To implement buffers above the 2-GB line and 8-byte RBA you must:
   a. Migrate to IBM MQ V8 for z/OS
   b. Change (ALTER) the buffer pool to LOCATION(ABOVE)
   c. Both a and b
   d. Choices that are presented in this question are not complete

2. True or false. To use 8-byte RBA in a new, never started queue manager, you must still run the CSQJUCNV utility.

3. True or false. PAGECLAS is an attribute exclusive to buffers defined with LOCATION(ABOVE).

Figure 6-12. Checkpoint questions (1 of 2) WM3121.1

Notes:

Write your answers here:

1.

2.

3.

WebSphere Education IBM

Checkpoint questions (2 of 2)

4. Select all correct answers. Which statements regarding use of FIXED4KB value are correct about the BUFFPOOL PAGECLAS attribute?
   a. Removes the overhead of the page-fix by ensuring buffers are permanently fixed in storage.
   b. Mitigates possibilities of the RBA being reset to 0.
   c. Ensures that messages are processed at the highest priority.
   d. If there is not sufficient storage in the LPAR, the queue manager might fail to start, or it might impact other processes in the LPAR.

5. Which statements are true about converting queue managers in a queue-sharing group to use 8-byte RBA?
   a. The first value in the `PARM` statement in the `CSQJUCNV` utility is set to **`INQSG`** .
   b. All member queue managers must be converted to 8-byte RBA.
   c. RESETPAGE must be run in all member queue managers.
   d. The `CSQJUCNV` utility must be run for each BSDS in each queue manager in the queue-sharing group.

Figure 6-13. Checkpoint questions (2 of 2) WM3121.1

Notes:

Write your answers here:

4.

5.

WebSphere Education

Checkpoint answers (1 of 2)

1. Select the best answer or answers. To implement buffers above the 2-GB line and 8-byte RBA you must:
   a. Migrate to IBM MQ V8 for z/OS
   b. Change (ALTER) the buffer pool to LOCATION(ABOVE)
   c. Both a and b
   d. Choices that are presented in this question are not complete

   Answer: d. **Unless OPMODE is set to (NEWFUNC,800), you cannot implement either feature. The answer is also missing the BSDS conversion**.

2. True or false. To use 8-byte RBA in a new, never started queue manager, you must still run the CSQJUCNV utility.
   Answer: True. **You can create a second copy of the BSDS and use it as input, then use the actual BSDS as a target for the version 2 BSDS.**

3. True or false. `PAGECLAS` is an attribute exclusive to buffers defined with LOCATION(ABOVE).
   Answer: False. **`PAGECLAS` might be used for buffers above or below the line, however, for buffers above the line, 4KB is the only valid `PAGECLAS` value.**

Figure 6-14. Checkpoint answers (1 of 2) WM3121.1

Notes:

WebSphere Education

Checkpoint answers (2 of 2)

4. Select all correct answers. Which statements regarding use of FIXED4KB value are correct about the BUFFPOOL PAGECLAS attribute?
   a. Removes the overhead of the page-fix by ensuring buffers are permanently fixed in storage.
   b. Mitigates possibilities of the RBA being reset to 0.
   c. Ensures that messages are processed at the highest priority.
   d. If there is not sufficient storage in the LPAR, the queue manager might fail to start, or it might impact other processes in the LPAR.
   Answer: a and d.

5. Which statements are true about converting queue managers in a queue-sharing group to use 8-byte RBA?
   a. The first value in the `PARM` statement is set to **`INQSG`**.
   b. All member queue managers must be converted to 8-byte RBA.
   c. RESETPAGE must be run in all member queue managers.
   d. The `CSQJUCNV` utility must be run for each BSDS in each queue manager in the queue-sharing group.
   Answer: a, b, and d. c is used to reset the RBA. You do not run RESETPAGE to convert the BSDS.

Figure 6-15. Checkpoint answers (2 of 2) WM3121.1

Notes:

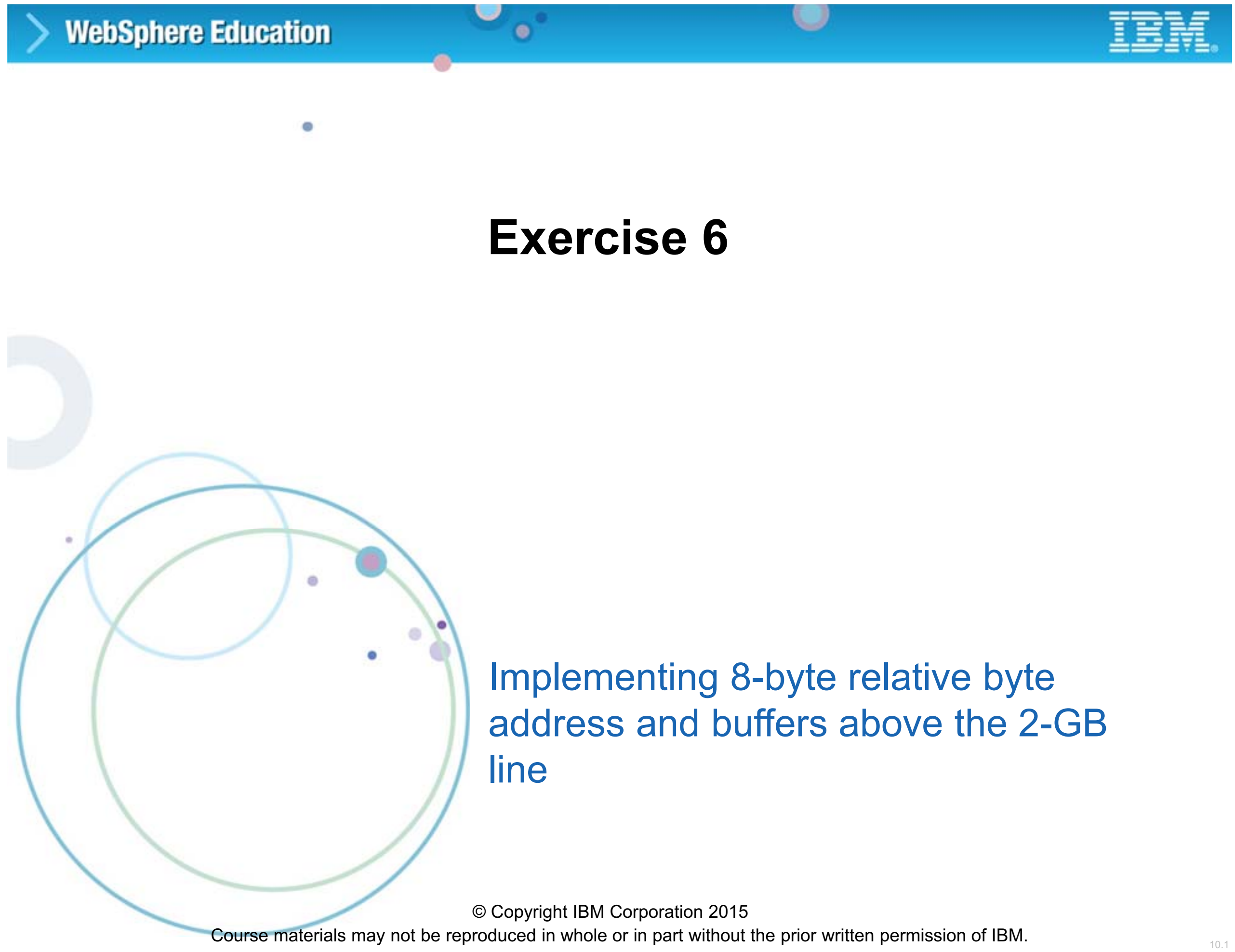


Figure 6-16. Exercise 7 WM3121.1

Notes:

WebSphere Education

Exercise objectives

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

Figure 6-17. Exercise objectives WM3121.1

Notes:

Unit 7. Introduction to IBM MQ for z/OS statistics and accounting

What this unit is about

This unit describes how to collect information that is critical for IBM MQ activities such as performance and tuning, troubleshooting, and capacity planning.

What you should be able to do

After completing this unit, you should be able to:

- Identify the types of information that can be captured from statistics and accounting records
- List the SMF record types that are used for IBM MQ accounting and statistics
- Describe how to generate accounting and statistics traces
- Describe how to use SupportPac MP1B to baseline application activity, identify high use queues, and analyze potential resource constraints
- Describe how to use SupportPac MP1B to analyze channel initiator information for dispatchers, adapters, SSL, and channels
- Describe the information that SupportPac MP1B generates to support analysis of queue-sharing group resources

How you will check your progress

- Checkpoint questions
- Lab exercises

WebSphere Education

Unit objectives

- Identify the types of information that can be captured from statistics and accounting records
- List the SMF record types that are used for IBM MQ accounting and statistics
- Describe how to generate accounting and statistics traces
- Describe how to use SupportPac MP1B to baseline application activity, identify high use queues, and analyze potential resource constraints
- Describe how to use SupportPac MP1B to analyze channel initiator information for dispatchers, adapters, SSL, and channels
- Describe the information that SupportPac MP1B generates to support analysis of queue-sharing group resources

Figure 7-1. Unit objectives WM3121.1

Notes:

Warning

This unit relies on formatting and analysis that is provided by the MP1B utility, available in the form of a SupportPac. MP1B is the widely used analytical and formatting tool for accounting and statistics records. The objective of this unit is to help you *get started* with statistics and accounting work. Proficiency in this work might require regular activities with accounting and statistics, and extra research of the IBM MQ processes.

MP1B is a SupportPac. Review the note that is included at the start of the MP1B documentation:

"Information contained in this report has not been submitted to any formal IBM test and is distributed "as is". The use of this information, and the implementation of any of the techniques, is the responsibility of the customer, and depends on the customer's ability to evaluate and integrate them into their operational environment."

WebSphere Education

IBM MQ for z/OS statistics and accounting

- What do you need to accomplish?
  - Determine how the queue manager is doing
  - Establish a normal baseline
  - Investigate reasons for poor response
  - Mitigate outages that are caused by resource constraints
  - Identify problem applications
- How is the channel initiator doing?
  - Adapters
  - DNS
  - SSL
  - Resource use
- Problems that are experienced might be outside IBM MQ

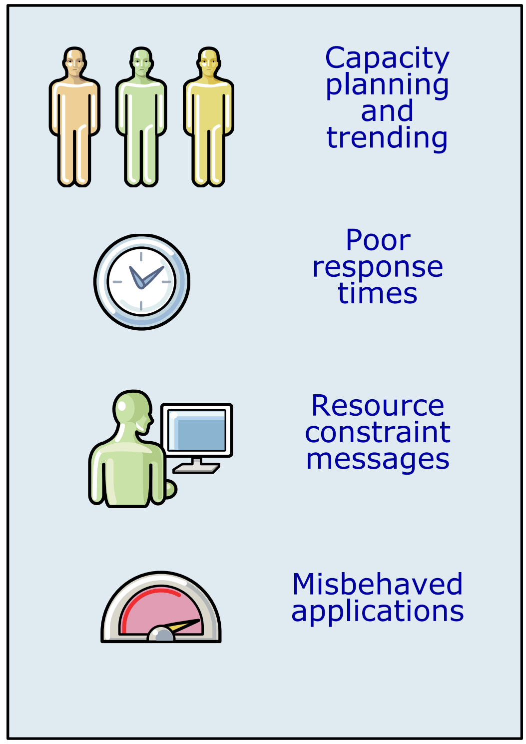


Figure 7-2. IBM MQ for z/OS statistics and accounting WM3121.1

Notes:

Accounting and statistics records help IBM MQ administrators by providing the information required to respond to common questions, such as:

- How do you know whether things are going well? If not what do you need to do?
- How do you find out what's going on? Which are the top queues to watch?
- If a problem is being blamed on IBM MQ, how do you prove that IBM MQ is not the culprit?

After the IBM MQ infrastructure is configured, organizations implement a monitoring solution, which is a *reactive* method to be alerted of a problem. In addition to troubleshooting, statistics and accounting traces also avail the IBM MQ administrator with a *proactive* means to observe resource usage and prevent potential problems. Additionally, they help establish application profiles that can be used for capacity planning.

Use of statistics and accounting records is an administrative task that must be scheduled regularly. In this unit, you find information to help you get started with statistics and accounting work.

IBM MQ internal processes generate the statistics and accounting information written to SMF.

WebSphere Education

Information captured by statistics and accounting traces

Class	Trace Description
	SMF 115 statistics trace
01	IBM MQ subsystem statistics
	Queue manager statistics
02	Queue manager storage summary
03	Queue manager storage detail summary
04	Channel initiator statistics
	SMF 116 accounting trace
01	• Processor time spent processing MQI calls • Count of MQPUT, MQPUT1, and MQGET calls
03	Enhanced accounting and statistics data
04	Channel accounting data

Figure 7-3. Information captured by statistics and accounting traces WM3121.1

Notes:

The various IBM MQ process managers produce the accounting and statistics records according to the trace requested. The trace descriptions provide information on what type of record the different classes of traces capture, such as:

- Statistics class(01) contains information including basic queue statistics, count of messages, and log statistics.
- Statistics class(04) is channel initiator statistics, such as TCB information.
- Accounting class(01) shows information such as number of puts and gets, and total CPU time.
- Accounting class(03) has details such as task and queue CPU used, elapsed time, delays.
- Accounting class(04) shows details such as what are channels doing and high use channels.

You need to be aware of these considerations, which are covered in this unit:

- Impact on the environment. Traces, such as the statistics trace class 03, can have significant impact on the CPU. Although it is necessary to run these traces periodically, their use must be carefully monitored. A thorough explanation on the impact of each trace can be found in: *SupportPac MP16: Capacity Planning and Tuning for IBM MQ for z/OS*.

- What is the trace interval? When does data collection take place? This information is presented later in this unit.

It is helpful to recognize the internal IBM MQ process managers and what identifiers represent data from which process manager. IBM MQ uses the following processes and corresponding four-character abbreviations to represent each process manager:

- QSST: Storage manager. Manages storage for IBM MQ for z/OS, for example, storage pool allocation and expansion.
- QJST: Log manager.
- QMST: Message manager. Processes all IBM MQ API requests.
- QIST: Data manager. Manages the links between messages and queues. Calls the buffer manager to process the pages with messages.
- QPST: Buffer manager. Manages the buffer pools in virtual storage, the writing of pages to page sets, and the reading of pages from page sets.
- QLST: Lock manager.
- Q5ST: DB2 manager.
- QEST: Coupling facility manager. Manages the interface with the coupling facility.
- QTST: Topic manager. Manages publish/subscribe topic and subscription information.
- QESD: SMDS usage.
- QWSX: Channel initiator.

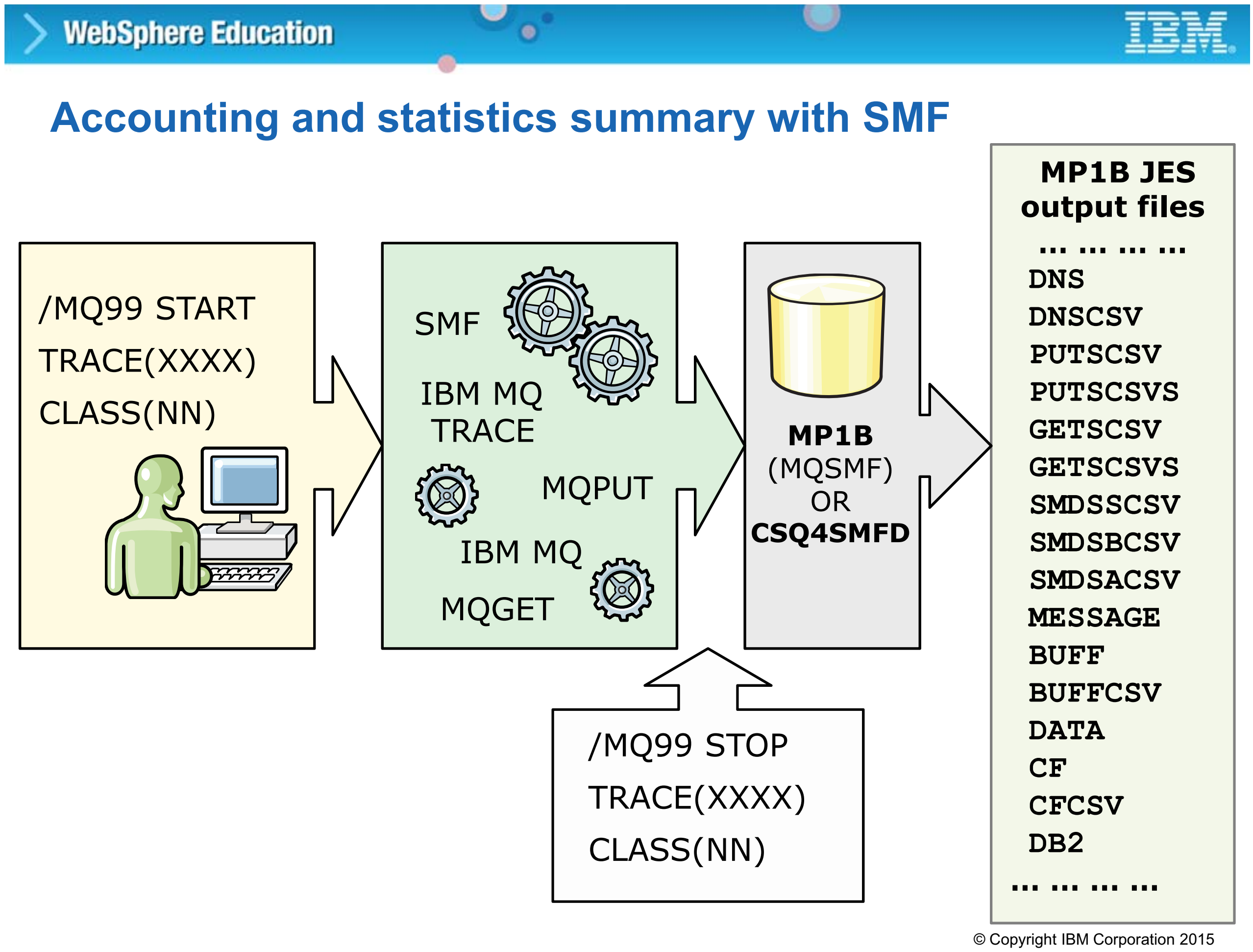


Figure 7-4. Accounting and statistics summary with SMF WM3121.1

Notes:

The most common destination for statistics and accounting IBM MQ traces is the System Management Facility, or SMF.

Before you start one or more IBM MQ traces, SMF must be running. SMF must be configured to record IBM MQ statistics, SMF type 115, and IBM MQ accounting records, SMF type 116.

Use the SMF IFASMFDP utility to copy and preserve your trace to a separate file. Although the graphic in this slide is not explicit regarding the copy step, IFASMFDP should be run before formatting.

The output from the IFASMFDP utility reports on the statistics and accounting records captured in your trace. Optionally, you can include selection criteria in the IFASMFDSP SYSIN file to filter what data is copied to your data set. Consult the z/OS SMF documentation for a list of selection parameters.

After you save your SMF data set, you are ready to format it.

Sample JCL and output from the IFASMFDP utility is shown in the notes display.

Example

JCL used to copy the SMF data to a separate data set.

```
//SMFDMP   EXEC PGM=IFASMFDP
//DUMP01   DD DSN=SYS2.S101.MAN1,DISP=SHR
//OUTDD1   DD DSN=TSM0099.SMF0007,DISP=(NEW,CATLG,DELETE),
 ... ... ...
//SYSIN    DD *
   INDD(DUMP01,OPTIONS(DUMP))
    OUTDD(OUTDD1,TYPE(115,116))
```

The IFASMFDP program provides a summary of the IBM MQ SMF records found and copied to the data set.

Example

Example IFASMFDP output:

```
OUTDD(OUTDD1,TYPE(115,116)) -- SYSIN
INDD(DUMP01,OPTIONS(DUMP)) -- SYSIN
OUTDD1   -- TSM0004.SMF0007
DUMP01   -- SYS2.S101.MAN1
                                     SUMMARY ACTIVITY REPORT
START DATE-TIME  09/08/2015-13:54:41                         END DATE-TIME
RECORD       RECORDS          PERCENT      AVG. RECORD   MIN. RECORD   MAX.
  TYPE          READ         OF TOTAL           LENGTH        LENGTH
     2             0
     3             0
   115            93          42.47 %         2,267.91           280
   116           126          57.53 %         2,784.47           404
TOTAL            219            100 %         2,565.11           280
NUMBER OF RECORDS IN ERROR                 0
```

Although there are separately licensed products, such as Tivoli Decision Support, to format the IBM MQ traces, there are two readily available options at no additional cost:

- SupportPac MP1B formats data on key IBM MQ components. It has a built-in ‘expert’ to analyze the data and writes informational, warning, or error messages on its findings.
- CSQ4SMFD utility prints the data in hex format by including component-specific eye catchers to identify the different IBM MQ processes. The data must be analyzed by reading the hex data and following the record layout definitions.

MP1B is the commonly used utility to process the SMF records. The load module for MP1B is called MQSMF. MQ1B and MQSMF are used interchangeably in this unit. MQSMF generates a number of different JES output files extracted, or derived, from accounting and statistics records.

WebSphere Education

Setting up accounting and statistics traces (1 of 2)

- Before you start to set the traces, understand *when* the trace records are collected
  - Check the **`STATIME`** value in the **`CSQ6SYSP`** parameter module
  - If STATTIME is 0, the SMF interval (INTVAL) is used
- SMF must be set up to generate the statistics and accounting records:
  - Configure by requesting: **`SETSMF SYS(TYPE(115,116))`**
  - Confirm by typing **`/D SMF,O`** The result should display:
    `SYS(TYPE(115,116)) -- PARMLIB`
- Optional: Switch or clear the existing SMF trace to discard any unneeded data

Figure 7-5. Setting up accounting and statistics traces (1 of 2) WM3121.1

Notes:

In this slide, you look at the areas that need to be addressed to set up the traces.

When the data is collected:

Before the trace is set up, ensure that you understand when the trace information is collected. For statistics traces, check the z/OS SMF **INTVAL** setting and the **STATIME** value in the **CSQ6SYSP** macro:

- If the STATTIME value is 0, the information is collected when the SMF interval (INTVAL) expires or when **INTVAL** is changed. If the SMF interval is 15, then data is collected at intervals such as hh: 00, hh: 15, hh: 30, hh; 45.
- If STATTIME is not 0, the trace information is collected in interval of minutes according to the value of STATTIME. For example, if STATTIME is 15 minutes, the data might be collected at 15:07, 15:22, 15:37, 15:52. STATIME value can be 0 - 1440.

For accounting traces:

- Accounting class(01): When task ends; this period might be days in duration
- Accounting class(03): When statistics are collected, or when task ends
- Accounting class(04): When statistics are collected, or when the channel ends

Check SMF information:

- You can check the SMF settings by typing **/D SMF,O** in the SDSF log pane. The items you need to check are:

  ```
  INTVAL(30) -- DEFAULT
    ... ... ... ... ...
  SUBSYS(STC,TYPE(115,116)) -- SYS
  ```

  The output confirms that the collection interval, or INTVAL value is 30 minutes, and that the IBM MQ statistics and accounting records are being collected.

- The name of the active SMF data set also needs to be checked. This data set is copied to your data set by using the IFASMFDP utility. You can optionally switch SMF logs to start your collection with an empty log. You can add time and date selection statement to your IFASMFDP utility to isolate a specific interval in the data set.

  To check the name of the active data set, type **/D SMF** in the SDSF log panel. The response should indicate the active data set:

  ```
  D SMF
   IEE974I 10.03.25 SMF DATA SETS 637
             NAME                VOLSER SIZE(BLKS) %FULL  STATUS
           P-SYS2.S101.MAN1      MVM005      3600     5  ACTIVE
           S-SYS2.S101.MAN2      MVM005      3600     0  ALTERNATE
  ```

WebSphere Education

Setting up the accounting statistics traces (2 of 2)

- Check the **ACCTQ,STATCHL**, and **STATACHL** queue manager and object attributes
- Start IBM MQ trace by typing the **START TRACE** command with correct specifications, such as:
  - **START TRACE(ACCTG) CLASS(04)**
  - **START TRACE(STAT) CLASS(01)**
  - **START TRACE(STAT) CLASS(04)**
- Stop the trace by typing the IBM MQ command **STOP TRACE** for the corresponding trace and class
- Confirm active traces by using the **DISPLAY TRACE** command

Figure 7-6. Setting up the accounting statistics traces (2 of 2) WM3121.1

Notes:

Checking IBM MQ definitions

- Queue manager ACCTQ controls collection on queues:
  - If queue manager is set to ON, records are collected for all queues with same attribute set to QMGR.
  - If queue manager is set to NONE, no records are collected regardless of the setting in the queue.
  - If queue manager is set to OFF, records are collected for queues with the same attributes set to ON.
- Queue ACCTQ:
  - If same attribute on queue manager is set to NONE, no collection takes place regardless of the value of this attribute for the queue.
  - If the queue value for these attributes is set to QMGR, records are collected if the same attribute for the queue manager is set to ON. QMGR is the initial value.
  - If the queue value for these attributes is set to OFF, collection is disabled for this queue.

- Queue manager STATCHL controls collection on channels:
  - NONE and OFF work the same as in ACCTQ, but for channels.
  - LOW, MED, HIGH are the equivalent of ON in ACCTQ.
- Queue manager STATACLS controls collection on auto-defined CLUSSDR channels:
  - QMGR setting inherits the queue manager STATCHL setting.
  - OFF, LOW, MED, HIGH work the same as in STATCLS.
- Channel STATCHL:
  - QMGR setting inherits queue manager behavior for the same attribute.
  - LOW, MED, HIGH means collection is ON unless the queue manager has NONE in the same attribute.
  - OFF disables statistics collection for this channel.

Setting and controlling the IBM MQ trace

Before you start an IBM MQ trace, confirm that there are no active traces. You can see how to display the traces in the next page.

While the `START TRACE` attribute has other parameters, you can usually start most statistics and accounting traces by its name and class. The name and class can be abbreviated. For example, to start a class 4 channel accounting trace, you might type:

```
/STA TRACE(A) CLASS(4)
```

To stop a trace, you select the same parameters you used when you started the trace:

```
/STOP TRACE(A) CLASS(4)
```

You might also change selected settings of a trace by typing an /ALTER TRACE command. This command stops and starts the trace.

WebSphere Education IBM

DISPLAY TRACE

- IBM MQ trace information that is displayed includes trace number (TNO)
- Traces such as the CHINIT trace require specification of the TNO in the STOP TRACE attribute.

```
MQ04 DIS TRACE
  CSQW127I MQ04 CURRENT TRACE ACTIVITY IS - 946
  TNO TYPE   CLASS        DEST     USERID   RMID
  01  STAT   01           SMF      *        *
  02  STAT   04           SMF      *        *
  03  ACCTG  04           SMF      *        *
  00  CHINIT *            RES      *        *
  END OF TRACE REPORT
```


Figure 7-7. DISPLAY TRACE WM3121.1

Notes:

The figure shows the output of a DISPLAY trace command with some active traces. It is not necessary to use the trace number, or TNO attribute, to stop the statistics or accounting traces. You can stop these traces by typing the STOP TRACE command with the type and class attributes.

CHINIT type traces start automatically after SMF data collection is enabled. The contents of this trace are for use by IBM. There are two options to stop it:

- Type the STOP trace command by including the TNO number along with the trace type, For example:

  `STOP TRACE(CHINIT) TNO(0)`

- Include the **`STOP TRACE(CHINIT) TNO(0)`** command in the //CSQINPX data set so it is stopped upon start of the channel initiator.

IBM support might request other types of traces. This unit focuses on the accounting and statistics traces.

WebSphere Education IBM

Using the MP1B SupportPac

- Download instructions, JCL, and load module from SupportPac site
  - From the JCL library select member MQSMFP
  - Customize JCL with location of your MQSMF load module
  - Best to omit SYSIN statements for first run
- SYSIN can be used to narrow record selection criteria such as date and time range, queue manager name, or available thresholds
- Depending on SMF records generated, output is in the form of a series of JES output files of different formats:
  - Summary files on different IBM MQ components
  - CSV formatted files with information that can be imported to a spreadsheet for further review
  - Analytical files with messages which flag potential problems

MP1B JES output files

```
… … … …
DNS
DNSCSV
PUTSCSV
PUTSCSVS
GETSCSV
GETSCSVS
SMDSSCSV
SMDSBCSV
SMDSACSV
MESSAGE
BUFF
BUFFCSV
DATA
CF
CFCSV
DB2
… … … …
```


Figure 7-8. Using the MP1B SupportPac WM3121.1

Notes:

When you download the MP1B SupportPac, there are other utilities and documentation included. To work with the statistics and accounting formatting and analysis component, use the MQSMFP JCL member, The MQSMF load module is the MP1B utility.

Before you use the MP1B utility, you should have:

- Checked that collection of SMF accounting and statistics records for IBM MQ is enabled
- Reviewed the data collection cycles, started the traces, and captured the information
- Run the IFASMFDP utility to create your copy of the records
- Checked the report generated by IFASMFDP to ensure that you captured the expected records

When you are ready to submit MP1B, customize the MQSMFP JCL to reflect:

- The location of the MQSMF load module
- The name of *your copy* of the SMF records produced by the IFASMFDP utility

You can disregard the //IGNORE data set; it is a scratch pad area where you can save old, frequently used parameters.

The first time you run the job:

- Do not add any //SYSIN statements to mitigate unexpected output.
- Select the output with a **?** so you see the list of all the files generated.

When MP1B completes, before reviewing the trace data, take time to become familiar with the information provided about the specific run of the utility:

- Select the SDSF output with a question mark (**?)** to list the output JES output files.
- You should see a long list of JES output files that requires scrolling down to see the all the available output files.
- Scroll down and determine whether you have a SYSOUT file. If you find a SYSOUT file, check that there were no missing DD statements, or that no other error was reported. If you get an error such as **"error opening //STGCSV..."**, add the missing file to the JCL. For other errors, if not obvious issues, consult the MP1B documentation.
- Open the SYSPRINT data set. In the first part of this data set, you find a list of initial or default values for the record selection parameters available to run MP1B.
- Have your MP1B documentation open. The parameters listed on the SYSPRINT data set are documented in the section called *Record Selection Parameters.* If you do not provide any SYSIN parameters, the values reported in the SYSPRINT data set are used.
- The parameters are specified by typing one name and value pair per SYSIN data set record.

Important

When you specify a high `Detail` parameter value, *some* of the information formatted *might* be at a level of detail suitable for IBM support.

- At the end of the SYSPRINT data set, you find a summary of the records processed by MP1B. This information is helpful to confirm whether you captured a specific SMF record type.

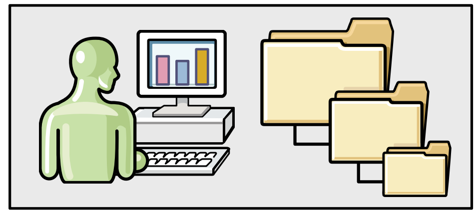
Example

```
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
```

WebSphere Education

Information that can be extracted with MQSMF (MP1B)



- What happens in the queue manager
  - How many API calls are processed
  - How much logging is taking place
  - How are the buffer pools behaving
- What is the application activity
  - What is happening with the queues?
  - Is anything wrong?
  - Which queues use the most CPU?
- Channel initiator information
  - What are the high use channels
  - Is any application approaching the maximum channels that are allowed or available channel initiator storage
  - Are there any issues with dispatchers, adapters, SSL, or DNS?
- Messages with analysis of potential problem areas

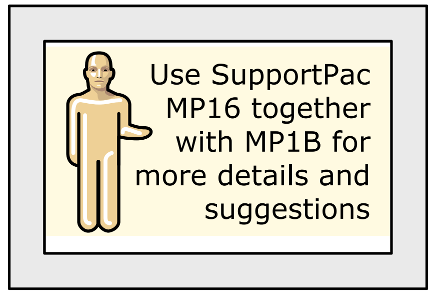


Figure 7-9. Information that can be extracted with MQSMF (MP1B) WM3121.1

What traces do you use to collect the different types of data.

- Information about *what happens in the queue manager*: Start the statistics trace. The cost of statistics trace is not significant.
- Information about *what is the application activity*: Start an accounting trace, which is a higher impact trace. A class 3 trace is of higher impact than a class 1 trace.
- Information on the channel initiator: Start class(04) accounting and statistics traces. You look at more details later in this unit.

There is a substantial amount of information you can obtain from the trace records as you proceed to work with statistics and accounting. While initial efforts might prove difficult, perseverance and continued research provide the required expertise.

To help get started, have two documents handy:

- The current copy of SupportPac *MP16: Capacity Planning and Tuning for IBM MQ for z/OS*. This document contains extensive information on IBM MQ internals and supplements the documentation in MP1B. As of the date this course was written, MP16 is updated through IBM MQ for z/OS V8.

- The current copy of the MP1B documentation. Although this course assumes IBM MQ V8, you should be able to use MP1B for IBM MQ V7.1, or as updated in the SupportPac site. You can supplement the information found in MP1B with the topics in IBM Knowledge Center.

Another valuable resource, along with the IBM Knowledge Center, is the collection of IBM MQ for z/OS performance SupportPacs. Although SupportPacs are created for specific IBM MQ versions, they might contain extra information for a feature released at the same version as the SupportPac. For example, SupportPac MP1H, for IBM MQ V7.1, contains a wealth of extra information on SMDS and the publish/subscribe topic scavenger. Always compare information in older SupportPacs with the IBM Knowledge Center to ensure that no new changes are added.

What do you need to look for?

In the rest of this unit, you to look at a selection of different trace data formatted by MP1B.

WebSphere Education IBM

How many API calls are processed

- Data from message manager provides counts of APIs calls
  - Failed calls are included in count
  - MQGETs are counted whether they return a message
- MP1B formats and writes data to MSGM and MSGMCSV files

```
S101,MQ04,2015/09/08,17:34:45,VRM:800,
  From 2015/09/08,16:43:39.318184 to 2015/09/08,17:34:45.753322,
  duration  3066 seconds
   MQOPENs     112,  MQCLOSEs      90,  MQGETs    4975,  MQPUTs    2689
   MQPUT1s       8,  MQINQs        36,  MQSETs       7,  C ALL H      0
   MQSUBs        1,  MQSUBRQs       0,  MQCBs        0
   MQCTLs        0,  MQSTATs        0,  Publish      0
   MQGet rate 1.000000/sec  MQPut rate 0.000000/sec
S101,MQ04,2015/09/08,17:35:21,VRM:800, .......
```

```
MVS,QM,Date,Time,Puts,Put1s,Gets,Open,Close,Inquire,Set,...........
S101,MQ04,2015/09/08,17:34:45,  2689, 8,  4975,   112, 90,  36,  7,
S101,MQ04,2015/09/08,17:35:21,     0, 0,     8,     0,   0,  0,  0,
```


Figure 7-10. How many API calls are processed WM3121.1

Notes:

There are some details in common with most of the formatted trace files generated by MP1B:

- The time and date range for the grouping of trace records collected. This information repeats throughout this file for all the records used to run MP1B.
- If the file name ends in 'SUM', there might be one group of trace records rolled up for the stated date/time threshold.
- Files ending in 'CSV' contain the information in the detailed file in comma-separated value formats.
- You might have an older performance and tuning presentation for MP1B, but find that the output is different. Differences in output might fluctuate because:
  - MP1B was run with the `Detail` parameter requesting a higher level of information. The default detail level is 5; maximum is 20.
  - You might be looking at details produced with older versions of IBM MQ and MP1B.

The MSGM and MSGMCSV files provide information helpful when you need to track the number of API calls made in the manager. Later in the presentation you see how to find information to identify what activity takes place on a queue basis.

While most of the identifiers for the details shown, such as MQOPENs, are intuitive, some data items are not so clear. One such example is “**`C ALL H`**”. To check what these fields represent, start with the MP1B documentation:

- Do a find on MSGM.
- The search should lead you to the **Message Manager** section.
- Look under “**The field names are**”. In that section, you find “**`C ALL H`**” stands for “`Close all handles - issued at the end of the task`”.

The name of the SCSQMACS PDS member that contains the layout of the raw SMF record for the Message Manager is provided in the MP1B documentation. This member is CSQDQMST. Earlier in this unit, a list of IBM MQ internal managers was presented. You can confirm that QMST is the identifier for the Message Manager component. In these SCSQMACS members, you find information similar to the field descriptions in MP1B.

The CSV files can be used to report queue manager trends and other details.

WebSphere Education

What are the API use patterns?

- Import CSV data to a spreadsheet and create profile charts

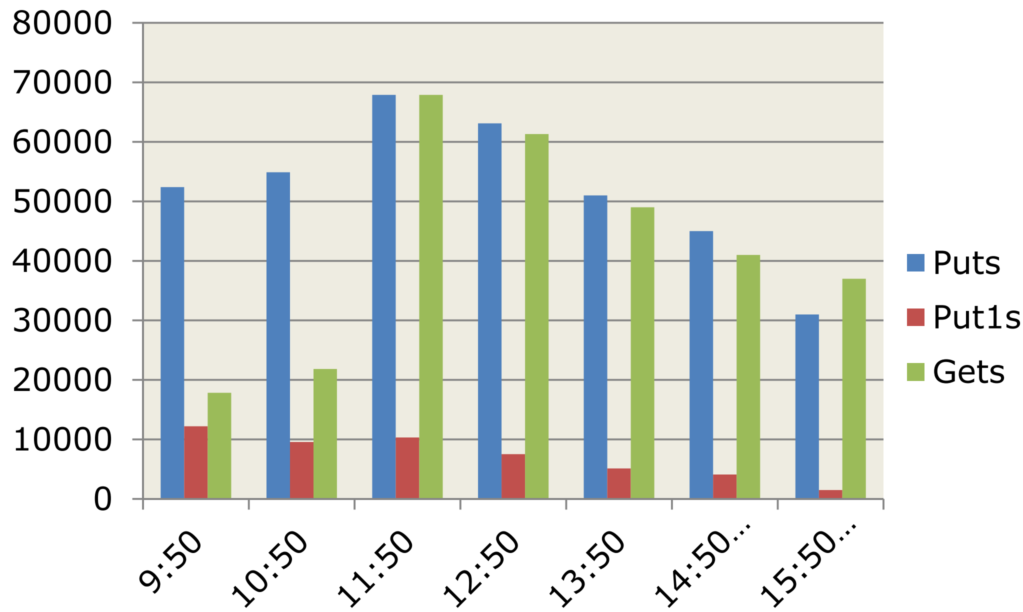


Figure 7-11. What are the API use patterns? WM3121.1

Notes:

The files containing comma-separated values are helpful when you need to create a chart. You might want to create a chart of API calls, channel initiator usage, or any other area where trends and profiles need to be documented.

To create a chart:

- Copy the information from the CSV file; in case of the chart in this page, MSGMCSV was used.
- Paste the information and save it in a text file.
- Import the data to a spreadsheet and generate the chart by selecting the pertinent fields.

If you remember the excerpt of the MSGMCSV shown earlier, you notice that Puts, Put1s, and Gets are field headers. These fields were self-documented in the chart legend.

Administrators might build a realistic application profile by taking the peak hour activity and create a cumulative chart with the data.

WebSphere Education IBM

How much logging is taking place: Key metrics

- Running out of log buffers
- How busy was the logging task?
  - Total time doing I/O in statistics collection period
  - Busy is > 95%
- Pages per I/O
  - The busier the system, more pages that are written per I/O
  - Do logs need to be striped?
- I/O response time
  - 500 uSeconds OK for DASD
  - 1000 uSeconds OK for mirrored DASD

- Log statistics are written to files LOG, LOGCSV, and LOGBUSY
- Analytical, log related messages that include the **`QJST`** identifier (such as **`MQQJST01W`**), are written to the MESSAGE file

Figure 7-12. How much logging is taking place: Key metrics WM3121.1

Notes:

One of the key areas that impacts performance is the logging task.

Excessive logging, or inadequate configuration of log files can result in delays. It is not uncommon to have unnecessary logging in an application. The delays might be due to the application design under the premise that non-persistent messages are lost during restarts of the queue manager. However, is the application justified in making all messages persistent?

- An application that updates critical data is justified in making the messages persistent. In that case, messages should be made persistent in the application code, not in the queue definition.
- But what about applications that query volatile data, such as a stock price or a basketball score? How long is the information valid for, and is it worth it to incur the logging?

A trace cannot provide information on *why* the data is persistent. Some of this information you obtain by attending application design meetings, or questioning your application contacts.

In the last unit, you reviewed how information from sources such as persistent messages, checkpoints, and a BACKUP CFSTRUCT command, are written to log records.

MP1B generates three files with logging information:

- LOG
- LOGCSV
- LOGBUSY

This slide summarizes some of the information to check when you analyze your logging environment. To find additional information to analyze your log environment, check the MP1B documentation.

You can also:

- Enter the /DIS LOG command for the queue manager
- Review the logging section on SupportPac MP16 for configuration suggestions and answers to questions such as:
  - How much log space does your message use?
  - How much log space do you need to back up a CF structure?

WebSphere Education IBM.

Logging statistics from LOG file (1 of 2)

- Did it run out of buffers
  - The *Wait for buffers* label displays how many times out of a number of requests logging had to wait for a buffer, followed by a percentage of wait per requests
  - Although a small percentage is acceptable, 0 is best

```
S101,MQ04,2015/09/09,10:47:46,VRM:800,
  From 2015/09/09,10:46:27.016847 to 2015/09/09,10:47:46.838918, duration
    80 seconds
    Wait for buffers(should be 0):             0 out of        16, 0%
    Total Number of pages written:           118
    Number of pages written/sec:               1
    Amount of data written/sec:                0 MB/Sec
    Total Number of write requests:          112
    Number of write requests/sec:              1
    Pages written per I/O:                     1
    Total number of read requests:           600
    _____,__ write requests,     CIs, Average I/O , After I/O  ,  pages/IO
  ... ... ... ... ...
```


Figure 7-13. Logging statistics from LOG file (1 of 2) WM3121.1

Notes:

STOP **Stop**

Screen displays in this course are generated to acquaint you with the *appearance* of the information contained in the MP1B output files as of the date this course was created.

Some of the key data to analyze the output is identified in the notes for the display. The data itself is *not* set up to contain a problem or specific situation. You use this material as a starting point, and supplement it with the suggested documentation.

The setup of the course environment *should not be construed* as an optimal configuration. The objective of the course environment is to accommodate the different resources required for each student for the lab exercises. The volume of activity generated to capture the trace data is small.

All displays were produced in the WM312 course environment by using IBM MQ V8 for z/OS and the August 2015 version of MP1B.

The information in the LOG file is split into two slides. A key piece of information for the LOG view is Wait for buffers; so critical that even the label states “should always be 0”. For more information about what to do for this situation, see the MP1B documentation. Guidance for the **`“Wait for buffers”`** field is copied as an example of the information you find in the MP1B documentation:

Example

Wait for buffers should always be zero. If this number is greater than zero, the internal buffer filled up and there was no space to store any more data. Applications processing persistent messages are delayed until buffers are available.

Space in this buffer is freed when the I/O that uses the space has completed. This problem can be caused by:

1. Active logs filling up. Perhaps due to a problem with archiving.
2. More data is being logged than the I/O system can handle. Improve the I/O rate, perhaps by striping the logs, or moving the log data sets to low use volumes, or reduce the work on the queue manager.
3. It can be caused by putting large messages, or many smaller messages, and then doing a commit. If the application has put more data than available buffers, you get a wait for buffers condition.

WebSphere Education IBM

Logging statistics from LOG file (2 of 2)

```
... ... ...
_____,__ write requests,    CIs, Average I/O , After I/O   , pages/IO
                                 time in uSec, time in uSec,
 Log 1, 1 page        55,        55,        300,           8,        1
 Log 1,>1 page         1,         4,        355,           2,        4
 Log 2, 1 page        55,        55,        297,           7,        1
 Log 2,>1 page         1,         4,        342,           1,        4
 Standard deviation of first log, 1 page per I/O, response time +-  43
 Log 1, 1 page Longest I/O       1148 at 2015/09/09,15:46:30.249880 UTC
 Log 1, 1 page Longest Request  1249 at 2015/09/09,15:46:30.249880 UTC
 Log 1,>1 page Longest I/O        355 at 2015/09/09,15:46:30.228817 UTC
 Log 1,>1 page Longest Request    357 at 2015/09/09,15:46:30.228817 UTC
 Log 2, 1 page Longest I/O       1139 at 2015/09/09,15:46:30.249887 UTC
 Log 2, 1 page Longest Request  1242 at 2015/09/09,15:46:30.249887 UTC
 Log 2,>1 page Longest I/O        342 at 2015/09/09,15:46:30.228832 UTC
 Log 2,>1 page Longest Request    343 at 2015/09/09,15:46:30.228832 UTC
Log write rate          0MB/s per copy
Logger I/O busy : 0.02%
Logger task busy: 0.50%
```


Figure 7-14. Logging statistics from LOG file (2 of 2) WM3121.1

Notes:

This section of the LOG output shows:

1. Log reporting is done per control interval, or CI as 4-KB pages. For dual logging, 2 CIs are written per 4-KB page.
2. The existences of Log 2 shows use of dual logging.
3. `Log write rate:` (the number of pages written per log per second * 4096 bytes per page). Monitoring the response time that it takes to write one page gives you a good indication of the behavior of the DASD over time.
4. `Average I/O time in` uSec: OK values are 500 uSeconds for DASD, 1000 uSeconds for mirrored DASD.
5. `Standard deviation of first log, 1 page per I/O, response time +-` : The value should be close to 0 and represents how consistent the response time was for writing one page. A value larger than the average response might be an indicator that your I/O subsystem might not be responding consistently. *This value required investigation in the course environment; and was the result other virtual systems using the same device*.

6. `Number of pages per I/O`: If you have small messages under 4 KB, you might see one or two pages per I/O. Many pages per I/O are written with large messages, large units of work, or channels with a large batch size. Many pages per I/O are more efficient.

7. `Logger task busy`: is a calculated field that, as its description implies, indicates the percentage of time the log task was busy for the reported interval.

Note

What about information on checkpoints?

As noted earlier, some details reported by MP1B depend on the level of the `Detail` parameter you specify in the MQSMF `//SYSIN` statement. The checkpoint field, renamed to LLCheckpoints, is generated if the `//SYSIN Detail` field is >= 10.

LLCheckpoints shows the number of checkpoints due to the value of the LOGLOAD parameter. It does not include checkpoints taken due to logs filling up.

Here is an example of the first LOG file page for the same time interval, formatted by using `Detail 20.` LCheckpoints is in bold. Notice the extra information provided at the higher `Detail` level.

Note

```
From 2015/09/09,10:46:27.016847 to 2015/09/09,10:47:46.838918, duration    80
seconds

      Wait for buffers(should be 0):           0 out of       16, 0%
      Total Number of pages written:         118
      Number of pages written/sec:             1
      Amount of data written/sec:              0 MB/Sec
      Total Number of write requests:        112
      Number of write requests/sec:            1
      Pages written per I/O:                   1
      Total number of read requests:         600
      Write_Wait         0, Write_Nowait      418, Write_Force        15, WTB
      Read_Stor         12, Read_Active       588, Read_Archive        0, TVC
      BSDS_Reqs         20, CIs_Created        16, BFWR               45, ALR
      ALW                0, CIs_Offload         0, LLCheckpoints       0
      Read delayed       0, Tape Lookahead      0, Lookahead Mount     0
      Write_Susp        45, Write_Reqs        112, CI_Writes         118
      Write_Serl         0, Write_Thrsh         0, Buff_Pagein         0
      _____,__ write requests,      CIs, Average I/O , After I/O   , pages/IO
                                              time in uSec, time in uSec,
```

WebSphere Education

How are the buffer pools behaving

- All private (non-shared) puts or gets, persistent, or non-persistent use buffers in the buffer pool
- The buffer manager (QPST) writes buffer pool pages to page sets
- Two buffer pool patterns
  - Short-lived messages might be used before they are written to a page
  - Long-lived messages usually exceed the buffer memory limit and get written to a page set
- If buffer pools reach > 85% busy, asynchronous task writes to page set
- If buffer pools reach > 95% busy, synchronous task writes changed pages to page set
- Checkpoints also generate write to page set
- Performance critical buffer pools should be kept < 85% busy

Figure 7-15. How are the buffer pools behaving WM3121.1

Notes:

A buffer pool is an area of virtual storage in the queue manager address space used to minimize I/O to and from page sets on disk. You can think of the buffer pool as the first landing area for a message. For optimal scenarios, the buffer pool would be the only place a message exists until it is retrieved. This behavior is not always the case.

Buffer pools are written to page sets by the Deferred Write Process, or DWP (also known as the DWT) at the following times:

- An asynchronous write takes place when the threshold of free buffer pages goes below 15%. The DWT stops where there are 25% stealable buffers available. The term "***stealable buffers***" is defined later in this page.
- If the free buffer pages fall below 5%, a synchronous write occurs.
- For persistent messages, when the second checkpoint occurs for a page containing a change.
- For persistent messages, at shutdown.

To understand what is a “*stealable buffer*”, first you look at the five possible states of a page in a buffer pool.

- **Unused**: All pages in the buffer pool start with this state.
- **Changed and in use**: Means that the content of the pool no longer matches the content of its corresponding page on the page set, and the buffer pool is in use by an application.
- **Changed and not in use**: Same as *Changed and in use*, but not currently in use by an application.
- **Unchanged and in use:** Means that the page in the buffer pool matches its corresponding page on the page set, but an application is using it. This state might be a browse application.
- **Unchanged and not in use**: The page in the buffer pool matches its corresponding page in the page set, and is not in use by an application.

`Stealable buffers:` Buffers that are *unused* or *unchanged and not in use* are considered stealable buffers.

It is the number of *stealable buffers* that is represented as the free buffer pages that affect the behavior of the buffer pool.

A buffer page reverts to *unchanged and not in use* status after it is written to disk.

WebSphere Education IBM

Buffer pool statistics

- BUFF file

```
Buffer statistics
S101,MQ04,2015/09/10,09:49:21,VRM:800,
  From 2015/09/09,16:31:16.405524 to 2015/09/10,09:49:21.485137, duration
  62285 seconds
= BPool    0, Size    50000,%full now  0, Highest %full  0, Disk reads   120
... ... ...
= BPool    2, Size    50000,%full now  0, Highest %full  0, Disk reads     4
 > 02 Buffs   50000  Low    49994  Now   49994  Getp     129  Getn        28
   02 Rio         4  STW      125  TPW       2  WIO        2  IMW          2
   02 DWT         0  DMC        0  STL       8  STLA       0  SOS          0
   02 Below the bar   PAGECLAS 4KB
= BPool    3, Size    20000,%full now  5, Highest %full  8, Disk reads     0
 > 03 Buffs   20000  Low    18382  Now   18881  Getp    3256  Getn      1616
   03 Rio         0  STW     4850  TPW       4  WIO        4  IMW          4
   03 DWT         0  DMC        0  STL    1592  STLA       0  SOS          0
   03 Above the bar   PAGECLAS 4KB
```


Figure 7-16. Buffer pool statistics WM3121.1

Notes:

The MP1B BUFF file contains information for buffer pools with pertinent information displayed per group of buffer pools per collection interval. A corresponding BUFFCSV file is also produced which includes key values per buffer pool, per collection interval.

There is a significant amount of information about buffer pool statistics in the MP1B and MP16 documentation. You look at some key data that impact buffers.

- **Rio/RIO**: Number of pages read from page set.
- **TPW**: Total number of pages written to page sets.
- **DWT**: Number of times the deferred write process was started.
- **DMC**: Number of times pages were synchronously written because the stealable page threshold was less than 5%.
- **STL**: Number of times a page was not found in the buffer pool and a stealable page was used.
- **STLA**: Number of times there was contention when getting a stealable page.
- **SOS:** Short on storage condition incurred when there were no stealable pages available when a page was needed.

- **WIO**: The number or write requests.
- **Above the bar**: This area of the display indicates whether a specific buffer resides below or above the 2-GB bar.
- **PAGECLAS 4 KB**: This area of the display denotes the buffer pool page class (PAGECLAS). This value is either 4 KB, or FIXED4KB.

What are some of the first observations you can make?

- If SOS, STLA, or DMC are greater than zero complete one of the following options:
  - Increase the buffer pool
  - Move the page to a different buffer pool
- If DWT is 0, it means that there were always 15% free buffers.
- Check the `Highest % full`, and `Disk reads` in the upper right of the display.
- An example of the information you find on MP1B shows this explanation on Disk reads: "The `number of pages read from disk. You get best performance if all of your data is in the buffer pool. Reading from the page set usually indicates the buffer pool had filled, and so buffers had to be written out to the page set`"

MP16 and MP1B contain information on more displayed fields.

WebSphere Education IBM

Buffer pool statistics by message lifetime

- Short-lived messages
  - Aim to keep the set of working messages in memory
  - Configure sufficient buffers to accommodate the peak message rate
  - DWT should be 0 with a minimum of 15% free buffers; higher % preferable
  - TPW (pages written) might be >0 due to checkpoint activity but not because buffer pools are getting full. If TPW >0, check applications
  - RIO (read operations) should be 0
- Long-lived messages
  - Expect messages to be written to and read from page set
  - For the period for which messages were processed, such as written by day and read by night, RIO should be approximately the TPW
  - If RIO is larger than TPW pages are read multiple times, check for application issue

Figure 7-17. Buffer pool statistics by message lifetime WM3121.1

Notes:

When looking at buffer pools for *short lived messages* and buffer pool 0:

- As usual, DWT should be 0
- Checkpoint activity might make TPW greater than 0
- Unless messages are read from a page set because the queue manager restarted, RIO should be 0
- A greater than 0 value in STL might indicate pages not previously used, are read. In this case, you need to add enough buffers to handle the peak message rate. The causes might be:
  - Increased message rate
  - Messages not processed as fast as before (or buildup of messages)
  - Larger messages

For more observations and suggestions on how to analyze data for long-lived messages, see the information on SupportPac MP16.

MP1B also writes analytical messages to the MESSAGES file.

WebSphere Education

Activity by queue and application task

- Detailed application activity in SMF 216 records
  - START TRACE(ACCTG) CLASS(03)
  - Heavyweight, known to generate multiple records for each transaction, and SMF intervals for long UoWs
  - 5-10% CPU overhead
  - Might swamp SMF
- But details learned justify the overhead
  - Produces a significant amount of useful information on what is taking place
  - Turn on a few minutes each month
  - Become familiar with activities and patterns
- Reports on details such as
  - How much CPU was used by each queue or IBM MQ verb
  - Why were messages delayed
  - How many messages were processed

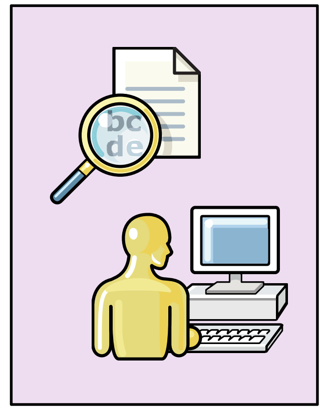


Figure 7-18. Activity by queue and application task WM3121.1

Notes:

The class 3 accounting trace is SMF intensive. However, the information captured by this trace is valuable. This information might help resolve the most challenging application situations, which, at first, might surface as “an IBM MQ problem”.

The accounting class 3 trace should be scheduled at regular intervals as one of the proactive tasks needed to adequately support applications that use the infrastructure.

There are documented guidelines for the resource-based areas you looked at earlier, such as logging and buffers. How to obtain a good picture of the application activity from the accounting traces might require more research and expertise. In the next group of slides, you look at the accounting data.

WebSphere Education

Is anything wrong?

- The TASKSUM file contains a summary of problems identified from the task and queue records
- Depending on the message reported, you might be able to obtain more details

Rec#	Count	Value	Message
96	1	1	MQQIST01W S101,MQ04 QIST read ahead messages > 0
53	2	23	MQQPST07I S101,MQ04 BP 3 Write rate 23 pages per second
86	1	1	MQQPST08I S101,MQ04 BP 0 Read rate 1 pages per second
13	3	4	MQQ5ST04E Q5ST SCS Maximum rows returned on query > 0
85	**3**	**450**	**MQQ5ST12W List Max DB2 time > 100**
194	27	584	MQTASK01E Queue not indexed PRICE.CL

- The letter at the end of each message denotes severity
  - **W** messages: Warnings
  - **E** messages: Errors, need to investigate
  - **S** messages: Serious, go look immediately

Figure 7-19. Is anything wrong? WM3121.1

Notes:

The TASKSUM file contains a summary of the analysis of the trace data done by MP1B.

To understand how to read these entries, select the next to last line in the display that starts with 85, which might or might not be a good candidate for further research.

- Message `MQQ5ST12W List            Max DB2 time > 100` was found *three times* (the number under the `Count` column).
- The highest value reported was 450, and this value was found on record #85 of the SMF input file.
- The meaning of 450 depends on the message itself. Go to the MP1B documentation, and search for the message ID, that is `MQQ5ST12W`. The search results in an entry in a table called *Thresholds for reporting out-of-line conditions in accounting data*. The description is:

  ```
  If the average time for DB2 request in micro seconds is greater than this
  value, produce message MQQ5ST11W, MQQ5ST12W,MQQ5ST13W or MQQ5ST14W.
  ```

  Based on this information, you determine that 450 the average DB2 time in micro seconds.

It might be that you do not always find a meaningful message in the TASKSUM file. You would normally look for records that might apply to a problem with an application, for example:

Example

```
Record# Count Value Message

2202 25 98908 MQTASK13E long commit time C,'CP15','IYFFC000',
```

The records captured in the lab environment for this slide have a more direct outcome and do not warrant further research. The record in the example shows a situation that needs further research, that is, how to determine why a commit time is long. With continued work, you develop a good idea of what type of records to look for in the TASKSUM file.

To obtain more details on the information reported by `MQTASK13E`, you run MP1B but request the highest level of detail for the specific record. For example, the //SYSIN statement would be:

```
Detail 20
FirstRecord 2202
LastRecord 2202
```

Messages resulting from this request are written to the MESSAGE file, or to the CMESSAGE file if the messages are for channel initiator statistics.

The letter at the end of the message indicates the urgency level. Any messages that end in S should be researched *immediately*. Messages that end in E should be investigated. Messages that end in W are warnings.

WebSphere Education

Is the queue manager set up OK? Applications behaved?

- MESSAGE file

```
MQSMDS12E S101,MQ04,2015/09/10,09:49:21,VRM:800, SMDS APPL1 write I/O:
 wait time (0.000294) > I/O time(0.000282). Buffer shortage.
MQQPST04E S101,MQ04,2015/09/10,09:49:21,VRM:800, BP 0 Many (120) pages
 read from disk.  This is typical of long lived messages. Buffer pool
 may be too small
MQQPST05I S101,MQ04,2015/09/10,09:49:21,VRM:800, BP 1 Some (8) pages
 read from disk.  Buffer pool may be too small
MQQPST05I S101,MQ04,2015/09/10,09:49:21,VRM:800, BP 2 Some (4) pages
 read from disk.  Buffer pool may be too small
MQQEST03W S101,MQ04,2015/09/10,09:51:19,VRM:800, QEST structure CSQ_ADMIN
 long average response time 120 uS
MQQEST03W S101,MQ04,2015/09/10,09:57:20,VRM:800, QEST structure APPL1
 long average response time 101 uS
MQQIST01W S101,MQ04,2015/09/18,15:57:32,VRM:800, QIST read ahead message
 count 1 > 0
```


Figure 7-20. Is the queue manager set up OK? Applications behaved? WM3121.1

Notes:

The messages file holds detail messages for other than channel initiator-related records.

You might find information in this file that helps you answer questions such as:

- Is the queue manager set up adequately? Analytical messages about the buffer pools and other critical resources are written to this file. For those messages that are not of an obvious nature, you can refer to the documentation for MP1B for an explanation.
- Are applications well behaved? In this file, you might find messages alerting you of certain application behaviors that might need follow up, such as:
  - An application that repeatedly fails to get a message from a queue, which might indicate a possible correlation or message ID issue.
  - Numerous gets for a message without the queue being indexed.

Some messages are only shown if a higher level of detail is used in the //SYSIN data set.

WebSphere Education

What information is reported on a queue

- TASK file

```
 S101,MQ04,2015/09/08,17:37:44,VRM:800,
MQ04 Batch Jobname:MQ04JGET Userid:TSM0004
Start time Sep  8 17:37:20 2015 Started this interval
Interval   Sep  8 17:37:20 2015 - Sep  8 17:37:20 2015 : 0.002527
Other reqs : Total ET              0.000007 Seconds
Other reqs : Total CPU             0.000007 Seconds
Commit count                       0
Commit avg elapsed time            0 uS
Open name                                  PRICE.CL
Queue type:QLocal                          PRICE.CL
Page set ID                        4       PRICE.CL
Buffer pool                        3       PRICE.CL
Get count                         95       PRICE.CL
Get avg elapsed time               6 uS    PRICE.CL
Get avg CPU time                   6 uS    PRICE.CL
Total Queue elapsed time         799 uS    PRICE.CL
Total Queue CPU used             674 uS    PRICE.CL
Grand total CPU time             681 uS
Grand Elapsed time               806 uS
```


Figure 7-21. What information is reported on a queue WM3121.1

Notes:

The TASK file might be used to obtain details to resolve problems that are reported against IBM MQ, but are caused by an application or an interfacing resource. Adequate use of the information in the TASK file usually requires added research, such as looking at several tasks to reconstruct the story of how a queue is used.

Information in the TASK file varies by the type of task. For example, the entry might be for a batch job, or for the MOVER task, which is another name used for the channel initiator. The detail level of information can also vary by the value provided in the MP1B `Detail` parameter.

Two factors to consider when you need to determine what takes place in an application:

- What are the areas where delays can occur
- Where is the time spent in an IBM MQ application

Areas where the delays can occur might be:

- The duration of puts and gets
- Full buffer pools
- I/O to page sets, coupling facility, or SMDS

- Duration of the commit
- Network issues
- Speed of disk when a high amount of data is written

Where is the time spent in an IBM MQ application:

- It is important to look at the end-to-end time that a message is in a queue, and what takes place in between.
- There might be messages in the queue where the get cannot complete. Might be a wait on a commit; until the commit occurs the get waits. This type of behavior surfaces as a “long time in IBM MQ”, but is an application coding issue.
- Generally, long MQPUTs or MQCommits usually add milliseconds to a request, such as 1 to 5 milliseconds. Delays such as 10 seconds are likely to be application problems.
- Other delays in the application might surface by a wait for a long DB2 request, or issues with CICS or IMS. Check that the application is not waiting for an outside resource.
- Generally a delay of more than 1 second is considered a significant problem; an increase in response time by 10 milliseconds is considered a small problem.

WebSphere Education IBM

Which queues use the most CPU?

- QALL file

```
Queue data summarised by queue
          0 Open name                               FROM.MQ24
          0 Queue type:QLocal                       FROM.MQ24
          0 Page set ID                    4        FROM.MQ24
          0 Buffer pool                   12        FROM.MQ24
          0 Get count                   6368        FROM.MQ24
          0 Get avg elapsed time           5 uS     FROM.MQ24
          0 Get avg CPU time               5 uS     FROM.MQ24
          0 Total Queue elapsed time   38186 uS     FROM.MQ24
          0 Total Queue CPU used       35094 uS     FROM.MQ24
          0 Open name
          0 Base name                               MQ24
          0 Queue type:QLocal                       MQ24
```

- Capture all "Total Queue CPU" used records and sort

```
Total Queue CPU used          301956 uS    PRICE.CL
Total Queue CPU used           35094 uS    FROM.MQ24
Total Queue CPU used           32983 uS    TSM0024.SHAREDQ
... ... ...
```


Figure 7-22. Which queues use the most CPU? WM3121.1

Notes:

The QALL file summarizes information for queues that hold messages during the collection interval. One useful piece of information provided is the total queue CPU used.

Note

Fields that contain large numbers might be reported in standard exponential notation. For example:

- 1.5e03 is 1500, 7.3e6 is 7,300,000

This file can be saved and manipulated to create a good profile of the high use queues.
You can enter a sequence of ISPF commands to get all records with "Total queue CPU", such as:

```
X ALL
F "Total Queue CPU" ALL
delete all X
prefix command COLS <=== usually just type COLS
sort 20 30 d <=== confirm correct columns to sort from the COLS display
```

WebSphere Education

Accounting and statistics for the channel initiator

- Statistics provides information about
  - How many channels
  - How many dispatchers, adapters, SSL, DNS
  - Are extra dispatchers, adapters, or SSL tasks needed?
- Accounting provides details about
  - What activity took place with the channels
  - Which are the high use channels

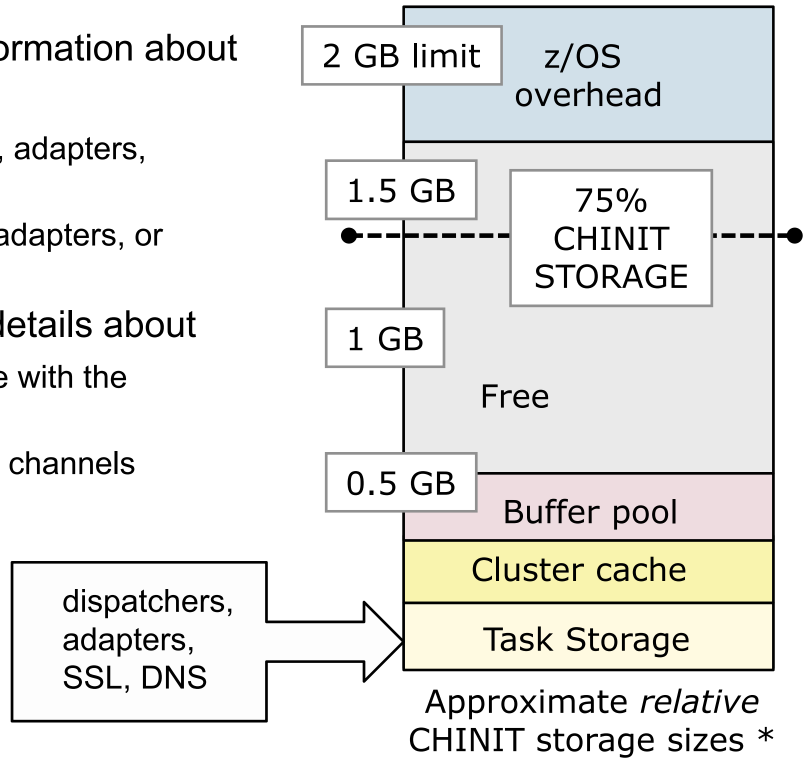


Figure 7-23. Accounting and statistics for the channel initiator WM3121.1

Notes:

The CLASS(04) traces enable collection of channel initiator-related accounting and statistics information.

The CLASS(04) statistics traces provide information important for capacity planning and tuning activities. The accounting traces provide data necessary to establish profiles on what is taking place with your channel initiator.

Much can be learned about the channel initiator by looking at the data in the class 04 traces and the documentation in the MP16 and MP1B SupportPacs. For example, MP16 documents the storage relative CHINIT storage sizes as pictured in the graphic.

MP1B also documents analytical channel initiator messages that might be found in the formatted file CMESSAGE. These messages might alert you to usage situations for channel-initiator related resources.

WebSphere Education IBM

Channel initiator summary

- CHINIT file

```
S101,MQ04,2015/09/10,17:38:52,VRM:800,
From 2015/09/08,17:36:38.799559 to 2015/09108,17:38:52.277925 duration
  133.47836
QSG............SG04
Peak number used of current channels...........   3
Peak number used of active channels ...........   3
MAXCHL. Max allowed current channels........... 200
ACTCHL. Max allowed active channels............ 200
TCPCHL. Max allowed TCP/IP channels............ 200
LU62CHL. Max allowed LU62 channels............. 200
Storage used by Chinit.........................  31MB
```

- Have you adjusted the projected number of channels?

Figure 7-24. Channel initiator summary WM3121.1

Notes:

The channel initiator summary is useful to determine trends in the number of channels and storage usage. The number of current and active channels reflect the values when the statistics record was created, so there might be changes across different collection intervals.

WebSphere Education

Channel initiator CSV

- CHINITCSV file

```
mvs,qm,date,time,QSG,HWMCCHL,HWMACHL,MaxCur,MaxAct,MaxTCP,MaxLU,StgMB
S101,MQ04,2015/09/08,13:54:41,SG04,1,1,200,200,200,200,21
S101,MQ04,2015/09/08,14:24:35,SG04,3,3,200,200,200,200,21
S101,MQ04,2015/09/08,14:31:22,SG04,3,3,200,200,200,200,23
... ... ...
```

- Chart your CSV data to create a profile regularly
- Number of current and actual channels
  – How close are you to the maximums
  – Do you have a *current* profile
- How much storage is used by the channel initiator?
  – Are you taking advantage of 64-bit storage
  – Is the storage consumption increasing

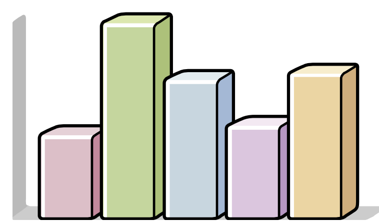


Figure 7-25. Channel initiator CSV WM3121.1

Notes:

The CHINITCSV is the comma-separated value file for the channel initiator information.

Establish a meaningful baseline for the applications by regularly collecting and analyzing this information and checking for any trends that might require adjustments.

WebSphere Education IBM

Dispatchers

- DISP file

```
S101,MQ04,2015/09/10,09:54:15,VRM:800,
From 2015/09/10,09:51:19.516234 to 2015/09/10,09:54:15.461299 duration
 175.945064 seconds
Task,Type,Requests,Busy %,     CPU used, CPU %,"avg CPU","avg ET"
    ,    ,        ,      ,      Seconds,      , uSeconds,uSeconds
   0,DISP,    3260,   0.0,     0.023592,  0.0,         7,       7
   1,DISP,    2222,   0.0,     0.020963,  0.0,         9,      15
   2,DISP,       0,   0.0,     0.000000,  0.0,         0,       0
   3,DISP,       0,   0.0,     0.000000,  0.0,         0,       0
   4,DISP,       0,   0.0,     0.000000,  0.0,         0,       0
Summ,DISP,    5482,   0.0,     0.044555,  0.0,         8,      10
   0,DISP, number of channels on this TCB,    1
   1,DISP, number of channels on this TCB,    1
   2,DISP, number of channels on this TCB,    0
   3,DISP, number of channels on this TCB,    0
   4,DISP, number of channels on this TCB,    0
Summ,DISP, number of channels on all TCBs,    2
```


Figure 7-26. Dispatchers WM3121.1

Notes:

Dispatchers are used to make calls to the communications network. A channel is associated with a dispatcher. If a channel has high use, you might notice a dispatcher associating a higher number of requests.

The number of dispatchers should not exceed the number of processors in the LPAR. When there are more dispatchers than processors, they might compete for CPU resources.

WebSphere Education

Adapters

- ADAP file

```
 S101,MQ04,2015/09/08,14:24:35,VRM:800,
From 2015/09/08,13:54:41.445708 to 2015/09/08,14:24:35.559284 duration
  1794.113575 seconds
Task,Type,Requests,Busy %,     CPU used, CPU %,"avg CPU","avg ET"
    ,    ,        ,      ,      Seconds,      , uSeconds,uSeconds
   0,ADAP,    2859,   0.0,     0.041845,  0.0,        15,      16
   1,ADAP,       0,   0.0,     0.000085,  0.0,         0,       0
   2,ADAP,       0,   0.0,     0.000000,  0.0,         0,       0
   3,ADAP,       0,   0.0,     0.000000,  0.0,         0,       0
   4,ADAP,       0,   0.0,     0.000000,  0.0,         0,       0
   5,ADAP,       0,   0.0,     0.000000,  0.0,         0,       0
   6,ADAP,       0,   0.0,     0.000000,  0.0,         0,       0
   7,ADAP,       0,   0.0,     0.000000,  0.0,         0,       0
Summ,ADAP,    2859,   0.0,     0.041930,  0.0,        15,      16
```


Figure 7-27. Adapters WM3121.1

Notes:

An adapter processes IBM MQ requests. The considerations to determine when more adapters are required are similar to the considerations to determine whether more dispatchers are required, that is:

When IBM MQ makes a request, the first free adapter task is used.

- You have enough adapters if there is at least one adapter either less than 1% busy, or not used.
- If all adapters are used, you *might* need to allocate extra dispatchers.
- If all adapters are used and they are busy for most of the collection interval, you *need* to allocate more adapters.

The adapter information is also available in CSV format in the ADAPCSV file. For extra guidance on adapter settings, see the MP1B documentation.

WebSphere Education

DNS

- DNS file

```
S101,MQ04,2015/09/09,10:52:27,VRM:800,
From 2015/09/09,10:49:27.580968 to 2015/09/09,10:52:27.935950
  duration 180.354982 seconds
Task,Type,Requests,Busy %,     CPU used, CPU %,"
    ,    ,        ,      ,      Seconds,      ,
   0,DNS ,       7,   0.0,     0.000124,  0.0,
Summ,DNS ,       7,   0.0,     0.000124,  0.0,

"avg CPU","avg ET",longest ,date      ,time
 uSeconds,uSeconds,uSeconds,          ,
       18,       15,      74,2015/09/09,10:51:02.635605
       18,       15,      74,2015/09/09,10:51:02.635605
```


Figure 7-28. DNS WM3121.1

Notes:

Domain name service (DNS) is used when a channel starts from an input IP address to a connection name, or for an outbound connection name to an IP address.

Since DNS might go outside your enterprise to look up a value, elapsed CPU time might be higher than expected.

A DNS task that is busy for 25% of the duration of the task is a candidate for investigation.

You might need to work with your network team if request time values are unacceptable.

WebSphere Education IBM

SSL

```
S101,MQ04,2015/09/08,17:16:05,VRM:800,
From 2015/09/08,17:15:04.864874 to 2015/09/08,17:16:05.682322 duration
  60.817448 seconds
Task,Type,Requests,Busy %,     CPU used, CPU %,"avg CPU","avg ET",...
    ,    ,        ,      ,      Seconds,      , uSeconds,uSeconds,
   0,SSL ,       0,   0.0,     0.000000,  0.0,        0,       0,
   1,SSL ,       3,   0.0,     0.000087,  0.0,       29,      28,
   2,SSL ,       0,   0.0,     0.000000,  0.0,        0,       0,
   3,SSL ,       0,   0.0,     0.000020,  0.0,        0,       0,
   4,SSL ,      38,   0.0,     0.004175,  0.0,      110,     115,
Summ,SSL ,      41,   0.0,     0.004283,  0.0,      104,     100,

 ... longest ,date      ,time
     uSeconds,          ,
            0,2015/09/08,17:15:55.295125
           59,2015/09/08,17:15:43.491042
            0,2015/09/08,17:15:55.295128
            0,2015/09/08,17:15:55.295129
         3319,2015/09/08,17:15:55.301686
         3319,2015/09/08,17:15:55.301686
```


Figure 7-29. SSL WM3121.1

Notes:

A running channel is associated with an SSL task.

According to behaviors observed, the cipher specification used does not appear to have an impact on the channel.

The SSL channels use the cryptographic co-processors available for the LPAR, so elapsed time might include time spent in the co-processor.

If SSL tasks are busy for a long time during the interval, increasing the number of SSL tasks might help. However, if the SSL tasks are waiting for internal processes such as the cryptographic co-processor, increasing the number of SSL tasks might not make a difference.

WebSphere Education IBM

Channel information (1 of 2)

- DCHS file

```
WMCLS.MQ04 10.31.18.11  Connection name                 10.31.187.119
WMCLS.MQ04 10.31.18.11  Channel disp                    PRIVATE
WMCLS.MQ04 10.31.18.11  Channel type                    CLUSRCVR
WMCLS.MQ04 10.31.18.11  Channel status                  RUNNING
WMCLS.MQ04 10.31.18.11  Channel STATCHL                 HIGH
WMCLS.MQ04 10.31.18.11  Remote qmgr/app                 MQ00
WMCLS.MQ04 10.31.18.11  Channel started date & time    2015/09/10,14:51:20
WMCLS.MQ04 10.31.18.11  Channel status collect time    2015/09/10,14:54:15
WMCLS.MQ04 10.31.18.11  Active for                        174 seconds
WMCLS.MQ04 10.31.18.11  Last msg time                  2015/09/10,14:51:20
WMCLS.MQ04 10.31.18.11  Last msg time delta             174.849459 seconds
WMCLS.MQ04 10.31.18.11  Batch size                                  50
WMCLS.MQ04 10.31.18.11  Messages/batch                              49.1
WMCLS.MQ04 10.31.18.11  Number of messages                       1,523
WMCLS.MQ04 10.31.18.11  Number of persistent messages            1,523
WMCLS.MQ04 10.31.18.11  Number of batches                           31
WMCLS.MQ04 10.31.18.11  Number of full batches                      30
WMCLS.MQ04 10.31.18.11  Number of partial batches                    1
```


Figure 7-30. Channel information (1 of 2) WM3121.1

Notes:

In the DCHS file, you can distinctly identify a channel by connection name, channel name, name of the remote queue manager, and channel type. This file contains substantial information on the channel, such as:

- What was the batch size and how many batches of messages were full or partial
- Which messages were persistent
- What was the average message size during the collection interval
- How much data was sent or received during the interval

The information in this file can be extracted to create a profile for the application or applications that use the channels.

WebSphere Education

Channel information (2 of 2)

```
WMCLS.MQ04 10.31.18.11  Buffers sent                       32
WMCLS.MQ04 10.31.18.11  Buffers received                1,524
WMCLS.MQ04 10.31.18.11  Message data                  877,248   856 KB
WMCLS.MQ04 10.31.18.11  Persistent message data       877,248   856 KB
WMCLS.MQ04 10.31.18.11  Non persistent message data         0     0  B
WMCLS.MQ04 10.31.18.11  Total bytes sent                1,136  1136  B
WMCLS.MQ04 10.31.18.11  Total bytes received          877,516   856 KB
WMCLS.MQ04 10.31.18.11  Bytes received/Batch           28,306    27 KB
WMCLS.MQ04 10.31.18.11  Bytes sent/Batch                   36    36  B
WMCLS.MQ04 10.31.18.11  Batches/Second                      0
WMCLS.MQ04 10.31.18.11  Bytes received/message            576   576  B
WMCLS.MQ04 10.31.18.11  Bytes sent/message                  0     0  B
WMCLS.MQ04 10.31.18.11  Bytes received/second     5,043  5043  B/sec
WMCLS.MQ04 10.31.18.11  Bytes sent/second                   6     6  B/sec
WMCLS.MQ04 10.31.18.11  Compression rate                    0
WMCLS.MQ04 10.31.18.11  Exit time average                   0 uSec
WMCLS.MQ04 10.31.18.11  DNS resolution time                 0 uSec
... ... ... ... ...
```


Figure 7-31. Channel information (2 of 2) WM3121.1

Notes:

This slide is a continuation of the channel information contained in the DCHS file.

Note

The SSL information is truncated. If you view this file, you might find incorrect information in the CipherSpec and distinguished name fields of the output. This error in the display is due to an APAR, which is not applied in the lab environment, it is *not* a problem with the MP1B utility.

In your site, you check to determine whether APAR UI20637 is applied, which was available October 2014. To check, open the <MQM>.V800.SCSQAUTH(CSQXSMFT) file and look for UI20637 and UI26133.

If they are not present, then the incorrect display is due to the missing maintenance. If you have them applied, then a problem must be reported to IBM.

WebSphere Education IBM

Information related to queue-sharing groups

- SMDSSCSV – SMDS space
- SMDSBCSV – SMDS buffers
- SMDSACSV – SMDS activity
- SMDS – SMDS statistics
- CF, CFCSV – Coupling facility statistics
- DB2

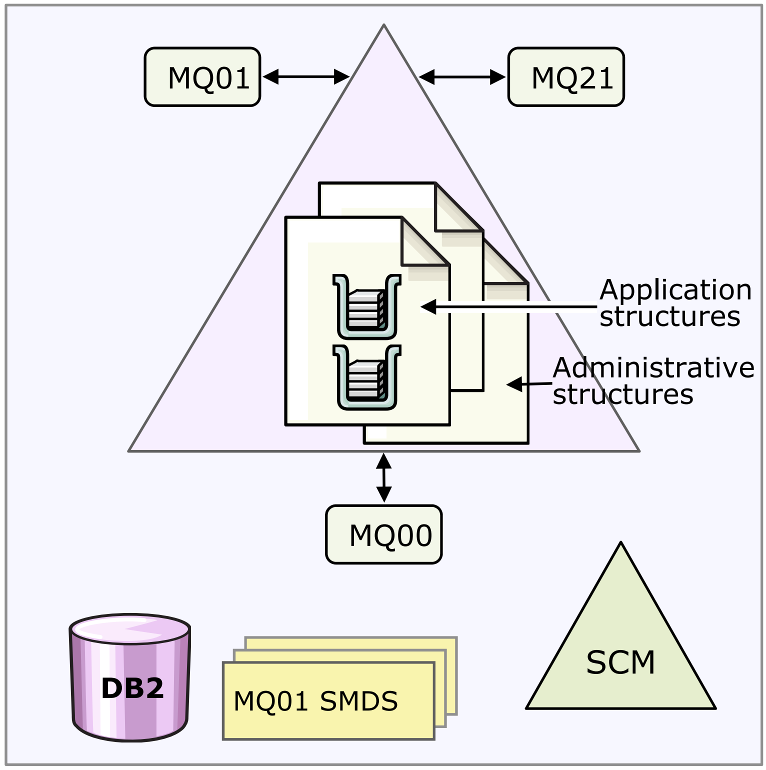


Figure 7-32. Information related to queue-sharing groups WM3121.1

Notes:

There is a number of files generated by MP1B that provide information about the resources used by queue-sharing groups.

The MP1B documentation contains information on how to interpret the details in each queue-sharing group-related file. The one exception to the available information is storage class memory (SCM); accounting and statistics collection does not have access to SCM information.

MP16 contains substantial information on the interfaces between queue-sharing groups, SMDS, SCM, and the coupling facility. This information includes recovery scenarios, large message scenarios, and storage class memory.

WebSphere Education

Unit summary

- Identify the types of information that can be captured from statistics and accounting records
- List the SMF record types that are used for IBM MQ accounting and statistics
- Describe how to generate accounting and statistics traces
- Describe how to use SupportPac MP1B to baseline application activity, identify high use queues, and analyze potential resource constraints
- Describe how to use SupportPac MP1B to analyze channel initiator information for dispatchers, adapters, SSL, and channels
- Describe the information that SupportPac MP1B generates to support analysis of queue-sharing group resources

Figure 7-33. Unit summary WM3121.1

Notes:

WebSphere Education

Checkpoint questions (1 of 2)

1. Select correct answers. The buffer manager:
   a. Starts a synchronous write to a page when buffers are 85% full
   b. Starts an asynchronous write to a page when buffers are 85% full
   c. Starts a synchronous write to a page when buffers are 95% full
   d. Starts an asynchronous write to a page when buffers are 95% full

2. Which MP1B formatted file do you use to create a profile of puts and gets for an application?
   a. PUTGTCSV
   b. MESSAGE
   c. CMESSAGE
   d. MSGMCSV

3. True or false. The class 3 accounting trace has a negligible system impact and should be left on to capture application profile activity.

Figure 7-34. Checkpoint questions (1 of 2) WM3121.1

Notes:

Write your answers here:

1.

2.

3.

WebSphere Education IBM

Checkpoint questions (2 of 2)

4. Select the best answers. What is a good log response time when writing one page:
   a. 500 uSeconds
   b. 1000 uSeconds
   c. 5000 uSeconds
   d. 10000 Microsecond

5. ACCTG trace CLASS(04) generates information on:
   a. Individual channels
   b. Coupling facility element and entry counts
   c. Individual SMDS buffer activity
   d. Log buffer accounting data to create an application profile

Figure 7-35. Checkpoint questions (2 of 2) WM3121.1

Notes:

Write your answers here:

4.

5.

WebSphere Education

Checkpoint answers (1 of 2)

1. Select correct answers. The buffer manager:
   a. Starts a synchronous write to a page when buffers are 85% full
   b. Starts an asynchronous write to a page when buffers are 85% full
   c. Starts a synchronous write to a page when buffers are 95% full
   d. Starts an asynchronous write to a page when buffers are 95% full

   Answer: b and c. Pages are also written if data in the buffer has changed and a second checkpoint occurs after the change.
2. Which MP1B formatted file do you use to create a profile of puts and gets for an application?
   a. PUTGTCSV
   b. MESSAGE
   c. CMESSAGE
   d. MSGMCSV

   Answer: d, MSGMCSV. PUTGTCSV does not exist. MESSAGE and CMESSAGE contain analytical messages from MP1B.
3. True or false. The class 3 accounting trace has a negligible system impact and should be left on to capture application profile activity.
   Answer: False. A class 3 accounting trace has significant system impact. It should run for limited short periods as needed to collect critical information.

Figure 7-36. Checkpoint answers (1 of 2) WM3121.1

Notes:

WebSphere Education

Checkpoint answers (2 of 2)

4. Select the best answers. What is a good log response time when writing one page:
   a. 500 uSeconds
   b. 1000 uSeconds
   c. 5000 uSeconds
   d. 10000 Microsecond

   Answer: a and b. 500 uSeconds is good for DASD. 1000 uSeconds is good for mirrored DASD. A uSecond is a Microsecond.

5. ACCTG trace CLASS(04) generates information on:
   a. Individual channels
   b. Coupling facility element and entry counts
   c. Individual SMDS buffer activity
   d. Log buffer accounting data to create an application profile

   Answer: a. The class 4 accounting trace generates detailed information on channel activity.

Figure 7-37. Checkpoint answers (2 of 2) WM3121.1

Notes:

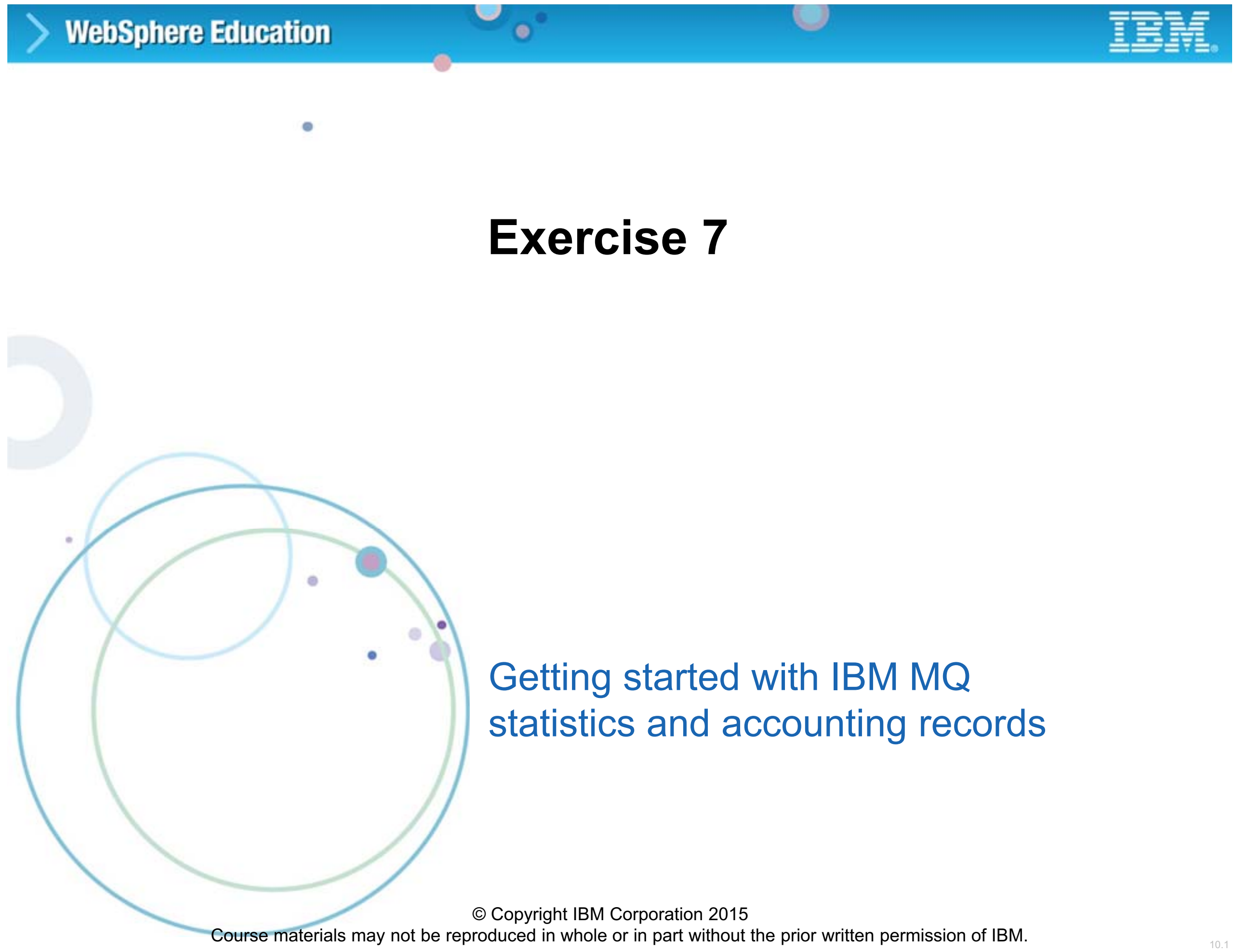


Figure 7-38. Exercise 7

WM3121.1

Notes:

WebSphere Education

Exercise objectives

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

Figure 7-39. Exercise objectives WM3121.1

Notes:

Unit 8. Course summary

What this unit is about

This unit summarizes the course and provides information for future study.

What you should be able to do

After completing this unit, you should be able to:

- Explain how the course met its learning objectives
- Access the IBM Training website
- Identify other IBM Training courses that are related to this topic
- Locate appropriate resources for further study

WebSphere Education IBM

Unit objectives

- Explain how the course met its learning objectives
- Access the IBM Training website
- Identify other IBM Training courses that are related to this topic
- Locate appropriate resources for further study

Figure 8-1. Unit objectives WM3121.1

Notes:

WebSphere Education

Course learning objectives

After completing this course, you should be able to:

- Explain how SSL/TLS contributes to authentication and confidentiality
- Describe how to configure SSL/TLS in IBM MQ z/OS queue managers and IBM MQ clients
- Implement SSL/TLS with multiple certificate authority (CA) certificates
- Implement various types of channel authentication rules
- Describe how to configure and manage queue-sharing groups
- Explain how to incorporate shared message data sets and storage class memory to maximize coupling facility storage
- Describe how to add a queue-sharing group to an existing IBM MQ cluster

Figure 8-2. Course learning objectives WM3121.1

Notes:

WebSphere Education

Course learning objectives

After completing this course, you should be able to:

- Implement 8-byte relative byte address and buffers above the 2-GB line
- Describe how to use statistics and accounting capabilities for problem determination, tuning, and capacity planning activities

Figure 8-3. Course learning objectives WM3121.1

Notes:

WebSphere Education IBM

To learn more on the subject (1 of 2)

- IBM Training website:
  `www.ibm.com/training`
- IBM support pack *MP16: Capacity Planning and Tuning for WebSphere MQ z/OS:*
  `www.ibm.com/support/docview.wss?uid=swg24007421`
- IBM support pack *MP1J: WebSphere MQ for z/OS V8.0 Performance report:*
  `www.ibm.com/support/docview.wss?rs=171&uid=swg24038347`
- IBM support pack *MP1H: WebSphere MQ for z/OS V7.1.0 Performance report:*
  `www.ibm.com/support/docview.wss?uid=swg24031663`
- IBM support pack *MP1B: WebSphere MQ Interpreting accounting and statistics data, and other utilities:*
  `www.ibm.com/support/docview.wss?uid=swg24005907`

Figure 8-4. To learn more on the subject (1 of 2) WM3121.1

Notes:

WebSphere Education

To learn more on the subject (2 of 2)

- Redbooks publication IBM MQ V8 Features and Enhancements:
  `www.redbooks.ibm.com/abstracts/sg248218.html?Open`

- Additional information by an IBM MQ product developer on connection authentication:
  `https://www.ibm.com/developerworks/community/blogs/messaging/tags/connauth?sortby=0&maxresults=30&lang=en`

Figure 8-5. To learn more on the subject (2 of 2) WM3121.1

Notes:

WebSphere Education IBM

Unit summary

- Explain how the course met its learning objectives
- Access the IBM Training website
- Identify other IBM Training courses that are related to this topic
- Locate appropriate resources for further study

Figure 8-6. Unit summary WM3121.1

Notes:

Appendix A. List of abbreviations

AES	Advanced Encryption Standard
AMS	(IBM) MQ Advanced Message Security
APF	authorized program facility
API	application programming interface
ASCII	American Standard Code for Information Interchange
ATM	automated teller machine
BMP	batch message processing program
BSDS	bootstrap data set
CA	certificate authority
CCDT	client channel definition table
CF	coupling facility
CFRM	coupling facility resource management
CICS	Customer Information Control System
CMS	Certificate Management System
CPF	Command prefix string
CPU	central processing unit
CRL	certificate revocation list
CSF	cryptographic service facility
CSV	comma-separated values
DASD	direct access storage device
DD	data definition (statement)
DER	Distinguished Encoding Rules
DES	Data Encryption Standard
DN	distinguished name
DNS	domain name server
DVIPA	dynamic virtual IP address
DWP	Deferred Write Process
DWT	Deferred Write Process
EB	exabyte
EBCDIC	Extended Binary Coded Decimal Interchange Code
ECC	elliptic curve cryptography

ECDH	Elliptic Curve Diffie-Hellman
ECDSA	Elliptic Curve Digital Signature Algorithm
ESM	external security manager
FIPS	Federal Information Processing Standards
FMID	function modification identifier
FTP	File Transfer Protocol
FTPS	File Transfer Protocol Secure
GB	gigabyte
GSKit	Global Security Kit
HTTP	Hypertext Transfer Protocol
HTTPS	Hypertext Transfer Protocol Secure
I/O	input/output
IBM	International Business Machines Corporation
ICSF	Integrated Cryptographic Service Facility
ID	identification data
IGQ	intra-group queuing
IMS	Information Management System
IP	Internet Protocol
IPCS	Interactive Problem Control System
ISPF	Interactive System Productivity Facility
JCEKS	Java Cryptography Extension Key Store
JCL	job control language
JES	Job entry subsystem
JKS	Java key store
LDAP	Lightweight Directory Access Protocol
LPA	link pack area
LPAR	logical partition
LUWID	logical unit of work identifier
MAC	media access control
MCA	Message Channel Agent
MQADV	MQ Advanced
MQAMS	MQ Advanced Message Security
MQCD	MQ channel definition structure

MQCNO	MQ client connect options structure
MQI	Message Queue Interface
MQMD	message queuing message descriptor
MQMFT	(IBM) MQ Managed File Transfer
MQSC	MQ Scripts
MSN	message sequence number
NSA	National Security Agency
OS	operating system
PCOMM	IBM Personal Communications
PDF	programmable command format
PDS	partitioned data set
PEM	Privacy Enhanced Mail
PIN	personal identification number
PKCS	Public-Key Cryptographic Standards
PSID	page set ID
PSW	program status word
QM	queue manager
QSG	queue-sharing group
RACF	Resource Access Control Facility
RBA	relative byte address
RFH	rules and formatting header
RIO	number of pages read from page set
RRS	Resource Recovery Services
SAF	System Authorization Facility
SCM	storage class memory
SDN	Subjects Distinguished Name
SDSF	System Display and Search Facility
SHA	Secure Hash Algorithm
SMDS	shared message data set
SMF	System Management Facilities
SMP/E	System Modification Program/Extended
SNI	Server Name Indication
SSID	subsystem identifier

SSL	Secure Sockets Layer
TCB	task control block
TCP	transmission control protocol
TLS	Transport Layer Security
TM	transaction manager
TNO	trace number
TPW	total pages written
TSO	Time Sharing Option
UOW	unit of work
VIPA	virtual IP address
VRM	Version, Release, and Maintenance (level code)
VUE	Value Unit Edition
VSAM	Virtual Storage Access Method
WLM	workload manager
WMB	WebSphere Message Broker
WTO	write to operator
XA	extended architecture
XCF	cross-system coupling facility
z/OS	zSeries operating system

Appendix B. Resource guide

Completing this IBM Training course is a great first step in building your IBM Middleware skills. Beyond this course, IBM offers several resources to keep your Middleware skills on the cutting edge. Resources available to you range from product documentation to support websites and social media websites.

Training

- **IBM Training website**
  - Bookmark the IBM Training website for easy access to the full listing of IBM training curricula. The website also features training paths to help you select your next course and available certifications.
  - For more information, see: `http://www.ibm.com/training`
- **IBM Training News**
  - Review or subscribe to updates from IBM and its training partners.
  - For more information, see: `http://bit.ly/IBMTrainEN`
- **IBM Certification**
  - Demonstrate your mastery of IBM Middleware to your employer or clients through IBM Professional Certification. Middleware certifications are available for developers, administrators, and business analysts.
  - For more information, see: `http://www.ibm.com/certify`
- **Training paths**
  - Find your next course easily with IBM training paths. Training paths provide a visual flow-chart style representation of training for many IBM products and roles, including developers and administrators.
  - For more information, see: `http://www-304.ibm.com/jct03001c/services/learning/ites.wss/us/e` n?pageType=page&c=a0003096

Social media links

Connect with IBM Middleware Education and IBM Training, and learn about the latest courses, certifications, and special offers by seeing any of the following social media websites.

- **Twitter**
  - Receive concise updates from Middleware Education a few times each week.

- Follow Middleware Education at: `twitter.com/websphere_edu`

- **Facebook**:
  - Follow IBM Training on Facebook to keep in sync with the latest news and career trends, and to post questions or comments.
  - Find IBM Training at: `facebook.com/ibmtraining`
- **YouTube**:
  - See the IBM Training YouTube channel to learn about IBM training programs and courses.
  - Find IBM Training at: `youtube.com/IBMTraining`

Support

- **Middleware Support portal**
  - The Middleware Support website provides access to a portfolio of downloadable support tools, including troubleshooting utilities, product updates, drivers, and Authorized Program Analysis Reports (APARS). The Middleware Support website also provides links to online Middleware communities and forums for collaboratively solving issues. You can now customize the IBM Support website by adding or deleting portlets to show the most important information for the IBM products that you work with.
  - For more information, see: `http://www.ibm.com/software/websphere/support`
- **IBM Support Assistant**
  - The IBM Support Assistant is a local serviceability workbench that makes it easier and faster for you to resolve software product issues. It includes a desktop search component that searches multiple IBM and non-IBM locations concurrently and returns the results in a single window, all within IBM Support Assistant.
  - IBM Support Assistant includes a built-in capability to submit service requests; it automatically collects key problem information and transmits it directly to your IBM support representative.
  - For more information, see: `http://www.ibm.com/software/support/isa`
- **IBM Education Assistant**
  - IBM Education Assistant is a collection of multimedia modules that are designed to help you gain a basic understanding of IBM software products and use them more effectively. The presentations, demonstrations, and tutorials that are part of the IBM Education Assistant are an ideal refresher for what you learned in your IBM Training course.

- For more information, see:
  `http://www.ibm.com/software/info/education/assistant/`

Middleware documentation and tips

- **IBM Redbooks**
  - The IBM International Technical Support Organization develops and publishes IBM Redbooks publications. IBM Redbooks are downloadable PDF files that describe installation and implementation experiences, typical solution scenarios, and step-by-step “how-to” guidelines for many Middleware products. Often, Redbooks include sample code and other support materials available as downloads from the site.
  - For more information, see: `http://www.ibm.com/redbooks`
- **IBM documentation and libraries**
  - IBM Knowledge Centers and product libraries provide an online interface for finding technical information on a particular product, offering, or product solution. The IBM Knowledge Centers and libraries include various types of documentation, including white papers, podcasts, webcasts, release notes, evaluation guides, and other resources to help you plan, install, configure, use, tune, monitor, troubleshoot, and maintain Middleware products. The Knowledge Center and library are located conveniently in the left navigation on product web pages.
- **developerWorks**
  - IBM developerWorks is the web-based professional network and technical resource for millions of developers, IT professionals, and students worldwide. IBM developerWorks provides an extensive, easy-to-search technical library to help you get up to speed on the most critical technologies that affect your profession. Among its many resources, developerWorks includes how-to articles, tutorials, skill kits, trial code, demonstrations, and podcasts. In addition to the Middleware zone, developerWorks also includes content areas for Java, SOA, web services, and XML.
  - For more information, see: `http://www.ibm.com/developerworks`

Services

- IBM Software Services for Middleware are a team of highly skilled consultants with broad architectural knowledge, deep technical skills, expertise on suggested practices, and close ties with IBM research and development labs. The Middleware Services team offers skills transfer, implementation, migration, architecture, and design services, plus customized workshops. Through a worldwide network of services specialists, IBM Software Service for Middleware makes it easy for you to design, build, test, and deploy solutions, helping you to become an on-demand business.

- For more information, see: `http://www-935.ibm.com/services/us/en/it-services/systems/middleware-services/`