z/OS Introduction and Workshop

WebSphere MQSeries
Unit Objectives

After completing this unit, you should be able to:

• Define Message, Queue, Queue Manager
• Explain Asynchronous Communication
• Understand WMQ Channels
WebSphere MQSeries

- The WebSphere® MQ products enable programs to communicate with one another across a network of unlike components (processors, operating systems, subsystems, and communication protocols) using a consistent application programming interface. Applications designed and written using this interface are known as message queuing applications because they use the messaging and queuing style:

  - Messaging: Programs communicate by sending each other data in messages rather than calling each other directly

  - Queuing: Messages are placed on queues in storage, allowing programs to run independently of each other, at different speeds and times, in different locations, and without having a logical connection between them.
Asynchronous Communication

- Caller and Server are decoupled
  - Server does not need presence of caller
  - Caller does not need availability of server
  - Loose binding between caller and server

- Analogy: Voicemail

- Asynchronous communications has advantages for certain roles
Messages

- A message is considered to be the unit of data to be moved from one application to another
- A message is built by an application and is consumed by a different application
- Messages can contain any kind of data:
  - Binary data
  - Text data (raw text, XML)
  - Structured data (C structures, COBOL CopyBook, etc.)
  - Anything
Types of Messages

WebSphere MQ defines four types of messages:

- **Datagram**: A simple message for which no reply is expected
- **Request**: A message for which a reply is expected
- **Reply**: A reply to a request message
- **Report**: A message that describes an event such as the occurrence of an error
Queues

- Messages are delivered asynchronously to a Queue

- Queues are a named entity which hosts a collection of messages

- Messages on queue are usually in “first in-first out” (FIFO) order
  - Options exist to process by priority or directly
Types of Queues

- **Local Queue**: A physical queue on the local queue manager, used to store messages
- **Alias Queue**: An alias of a physical queue; used as a level of ‘indirection’
- **Remote Queue Definition**: Not a physical queue, but a definition of a queue on a remote queue manager; used to send messages to remote queue managers
- **Model Queue**: A queue that is used as a template for dynamically-created queues
- **Transmission Queue**: A special kind of local queue used for the delivery of messages to remote queue managers
Queue Manager

- A queue manager is a collection of queues and their messages

- Queue managers can communicate with other queue managers using “channels”
  - Channel: Reliable transport mechanism for queue managers to exchange messages with each other
  - More on channels later….
What does WMQ provide?

- Exactly once message delivery
- Loosely-coupled applications
  - Asynchronous messaging
- A single, multi-platform API
  - Easy to use ... message centric interface
  - Network independent
  - Faster application development
- Universal ...
  - runs everywhere
WMQ Asynchronous Messaging

Logical Connection

Everything Constantly Available

Everything Conditionally Available

Busy

Not Available

Message/Queuing Environment

A

B

A

B

A

B

A
Queue Manager

- Program communicates with DB2
  - DB2 accesses Data

- Program communicates with Queue Manager
  - Queue Manager manages Messages
WebSphere MQ API

Program A

MQCONN MQDISC MQOPEN MQCLOSE MQGET MQPUT MQPUT1 MQCMIT MQBACK MQBEGIN MQINQ MQSET

MQI

Queue Manager

Process Object

Queue Manager Object

Queues
Communication with WMQ

System 1

- Program A
- Program B
- MQI
- Put Q2
- Put Q1
- Get Q1
- Messaging and Queuing
- XmitQ
- Q1

System 2

- Program C
- MQI
- Get Q2
- Messaging and Queuing
- Q2

TCP/IP is primary protocol
Channels

Program 1
get

put

Local Q2
Remote Q1
XmitQ QMB
Channel QMB.QMA Receiver
Channel QMA.QMB Sender

Program 2
get

put

Local Q1
Remote Q2
XmitQ QMA
Channel QMA.QMB Receiver
Channel QMB.QMA Sender
Channel Communication
Interconnectivity
Connectivity/Adapter Diagram
Application Scenario

Application

Queue Mgr ‘QM1’

Queue Remote ‘PAYROLL.QUERY’

Queue Transmission QM2’

‘SYSTEM.CHANNEL. INITQ’

Queue Local ‘PAYROLL.REPLY’

Queue Mgr ‘QM2’

Queue Local ‘PAYROLL’

Queue Transmission ‘QM1’

‘SYSTEM.CHANNEL. INITQ’

Application

Payroll Query

Query Message

Reply Message

Payroll processing

Query Message

Reply Message
Start MQSeries Queue Manager

%CSQ7 START QMGR

S CSQ7MSTR
$HASP373 CSQ7MSTR STARTED
CSQY000I %CSQ7 IBM WebSphere MQ for z/OS V7
CSQY001I %CSQ7 QUEUE MANAGER STARTING, USING PARAMETER MODULE CSQZPARM
CSQ3111I %CSQ7 CSQYSCMD - EARLY PROCESSING PROGRAM IS V7 LEVEL 004-005
CSQY100I %CSQ7 SYSTEM parameters ...

CSQJ127I %CSQ7 SYSTEM TIME STAMP FOR BSDS=2009-11-19 14:57:25.13
CSQJ001I %CSQ7 CURRENT COPY 1 ACTIVE LOG DATA SET IS 778

CSQP007I %CSQ7 Page set 0 uses buffer pool 0
CSQP007I %CSQ7 Page set 1 uses buffer pool 0
CSQP007I %CSQ7 Page set 2 uses buffer pool 1
CSQP007I %CSQ7 Page set 3 uses buffer pool 2
CSQP007I %CSQ7 Page set 4 uses buffer pool 3
CSQY220I %CSQ7 Queue manager is using 26 MB of local 787 storage, 1620 MB are free

CSQY022I %CSQ7 QUEUE MANAGER INITIALIZATION COMPLETE
CSQ9022I %CSQ7 CSQYASCP 'START QMGR' NORMAL COMPLETION
Start MQSeries Channel Initiator

%CSQ7 START CHINIT

S CSQ7CHIN,JOBNANE=CSQ7CHIN

CSQM138I %CSQ7 CSQMSCHI CHANNEL INITIATOR STARTING

$HASP373 CSQ7CHIN STARTED

CSQX000I %CSQ7 CSQXJST IBM WebSphere MQ for z/OS V7

CSQX001I %CSQ7 CSQXJST Channel initiator starting

CSQX011I %CSQ7 CSQXGIP Client Attachment feature available

CSQ9022I %CSQ7 CSQXCRPS 'START CHINIT' NORMAL COMPLETION

CSQX022I %CSQ7 CSQXSUPR Channel initiator initialization complete

CSQX023I %CSQ7 CSQXLSTT Listener started, port 1416 address *,TRPTYPE=TCP

CSQU012I CSQUTIL Initialization command handling completed
Each MQSeries Environment is Two Address Spaces

SDSF STATUS DISPLAY ALL CLASSES

PREFIX=CSQ*  DEST=(ALL)  OWNER=*  

<table>
<thead>
<tr>
<th>NP</th>
<th>JOBNAME</th>
<th>JobID</th>
<th>Owner</th>
<th>Prty</th>
<th>Queue</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>CSQ7CHIN</td>
<td>STC01194</td>
<td>STCOPER</td>
<td>15</td>
<td>EXECUTION</td>
</tr>
<tr>
<td></td>
<td>CSQ7MSTR</td>
<td>STC01193</td>
<td>STCOPER</td>
<td>15</td>
<td>EXECUTION</td>
</tr>
</tbody>
</table>
Stop MQSeries Channel Initiator and Queue Manager

%CSQ7 STOP CHINIT

%CSQ7 STOP QMGR
Additional Information

- WebSphere MQ v7.5 InfoCenter
- WebSphere MQ Redbooks
  - WebSphere MQ Primer (Redbook)
Professional Manuals and Documentation
Unit summary

Having completed this unit, you should be able to:

• Define Message, Queue, Queue Manager
• Explain Asynchronous Communication
• Understand WMQ Channels
Publish and Subscribe

Queue Manager

Pub/Sub Broker

Data

A

B

MQSUB - subscribe

MQSUBRQ – request retained publication