

#### **IBM Software Group**

#### WebSphere MQ 7.1 Multicast

Angel Rivera (rivera@us.ibm.com)
Miguel Rodriguez (mrod@us.ibm.com)
WebSphere MQ Unix® Level 2 Support
22 May 2012







# Agenda

- Introduction to Multicast
- Communication Information (COMMINFO) object
- Multicast attributes in a Topic object
- Multicast Security
- Configuration
- Testing of non-multicast Pub/Sub
- Testing of multicast

Thanks to Paul Clarke (IBM UK) for his assistance.





#### What is Multicast?

- "In computer networking, multicast is the delivery of a message or information to a group of destination computers simultaneously in a single transmission from the source." Wikipedia
- What messaging model can be configured in such a way whereby transmission of data from a single source can be sent to many?
- Answer: Publish and Subscribe





# IP Multicast (1)

- Multicast is a form of Publish and Subscribe handled by the Network Layer.
- In fact, Multicast is mostly implemented through the use of the Internet Protocol (IP).
- In IP Multicast the routing of messages, in the form of datagrams is handled and optimized by routers to a destination address.



# IP Multicast (2)

- Multicast can be ...
- more efficient than traditional unicast publish and subscribe,
- scaling to a high number of subscribers without affecting performance.
- However multicast is ...
- more complex and
- requires configuration of routers to pass the multicast transmissions.





#### Configuration of Multicast in MQ

- WebSphere MQ v7.1 provides the underlying code which enables MQ to participate in Multicast communications.
- MQ provides Topics whereby publishers of messages are decoupled with subscribers that consume these messages.
- Multicast is configured through:
- the MQ TOPIC object and
- a new MQ object called COMMINFO.





# Communication Information object (1)

- The attributes of COMMINFO define the behavior of the Multicast transmissions:
- Group address the group IP address or DNS name.
  - It must be set and checked to ensure that the value is a valid Multicast group address.
- Property propagation how many of the MQMD properties and user properties flow with the message.





# Communication Information object (2)

- Reliability setting amount of message history in kilobytes that is kept by the system to handle retransmissions.
- Data conversion The coded character set identifier that messages are transmitted on.



## **Notes: DEFINE COMMINFO (1)**

- http://publib.boulder.ibm.com/infocenter/wmqv7/v7r1/topic/com.ibm.mq.doc/sc14410\_.htm
- WebSphere MQ > Reference > Administration reference > MQSC reference > The MQSC commands
  - DEFINE COMMINFO
- The relevant fields are:
- TYPE(MULTICAST) => The only type as this time is MULTICAST
- GRPADDR(239.0.0.0) => The group IP address or DNS name.
- It is the administrator's responsibility to manage the group addresses.
- It is possible for all multicast clients to use the same group address for every topic; only the messages that match outstanding subscriptions on the client are delivered. Using the same group address can be inefficient because every client must examine and process every multicast packet in the network.
- It is more efficient to allocate different IP group addresses to different topics or sets of topics, but this requires careful management, especially if other non-MQ multicast applications are in use on the network.
- The default value is 239.0.0.0









# **Notes: DEFINE COMMINFO (2)**

n

0

t

e

S

- COMMEV(DISABLED) => Controls whether event messages are generated for Multicast handles that are created using this COMMINFO object.
   Events will only be generated if they are enabled using the MONINT parameter.
   If event messages are generated, then they will be placed on the SYSTEM.ADMIN.PUBSUB.EVENT.
   The default is DISABLED.
- MONINT(60) => How frequently, in seconds, that monitoring information is updated. If events messages are enabled, this parameter also controls how frequently event messages are generated about the status of the Multicast handles created using this COMMINFO object.
   The default is 60 seconds.
- MSGHIST(100) => This value is the amount of message history in kilobytes that is kept by the system to handle retransmissions in the case of NACKs (negative acknowledgments).
   The default is 100.
- PORT(1414) => The port number to transmit on. The default port number is 1414.



# Topic object

- The Topic object now contains two Multicast attributes:
- MCAST Specifies whether multicast is allowable in the topic tree
- COMMINFO The name of the Multicast communication information object associated with this topic object.



#### MCAST attribute for a Topic object

- The MCAST parameter specifies whether multicast is allowable at this position in the topic tree.
- ASPARENT: it is the default. The multicast attribute of the topic is inherited from the parent.
- ENABLED allows multicast traffic at this node (as well as non-multicast)
- ONLY allows only multicast traffic and prevents non-multicast traffic from happening.
- http://publib.boulder.ibm.com/infocenter/wmqv7/v7r1/topic/com.ibm.mq.doc/sc11305\_.htm
- The MQSC commands > DEFINE TOPIC





#### Client connection channel SHARECNV > 0

- The Multicast client, as any other MQ client, will require a Queue Manager name, hostname, port, and channel name.
- A Full Duplex client-server connection is needed.
- Make sure that the server connection channel attribute SHARECNV, sharing conversations, is set to value greater than zero (0).
- If zero is used, the Multicast function is disabled.





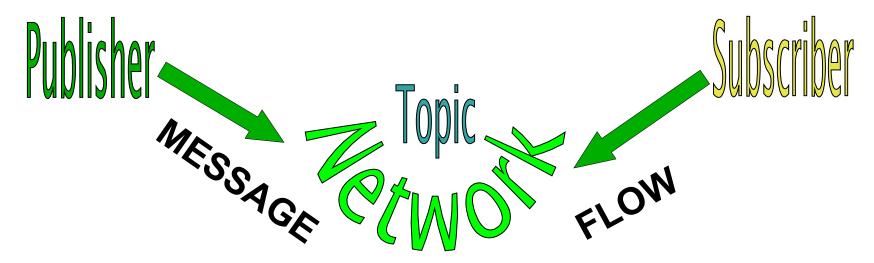
#### Multicast Features: High availability for multicast

- WebSphere® MQ Multicast provides continuous peer-to-peer operation as messages do not flow through the queue manager.
- In the intial multicast connection, the queue manager services the MQOPEN from the publisher and the MQSUB from the subscriber for a given topic object. Both the publisher and subscriber will be passed multicast topic object attributes which will be used to connect the publisher directly to the subscriber.



#### High availability for Multicast

- Multicast messages DO NOT flow through the queue manager.
- If the queue manager terminates unexpectedly, multicast message transmissions will continue to flow.



- There are two modes of operation:
- client connection and
- reconnecting client connection.

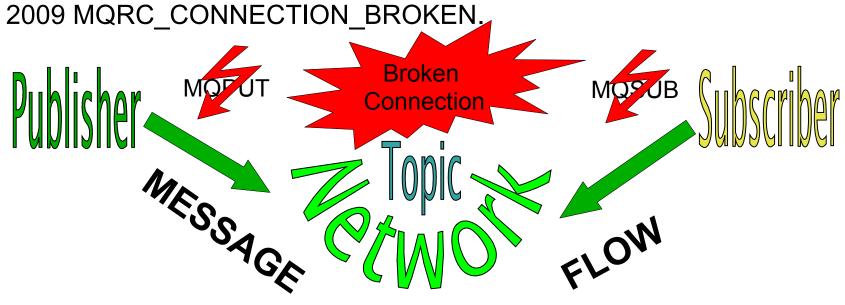




#### Client connection

Multicast communication flows while the client connection to the queue manager exists.

If the queue manager terminates unexpectedly, the clients receive:



However, a new publishing and/or subscribing client WILL NOT be able to connect to the topic until the queue manager is restarted and can service their initial MQI calls MQPUT and MQSUB

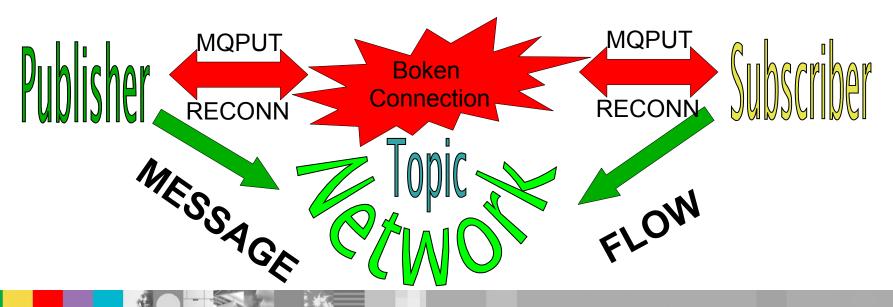




#### Reconnecting client connection

Multicast communication continues to flow even during a reconnection attempt to the queue manager.

The client attempts to reconnect to a queue manager, and if that reconnection fails, the connection handle becomes broken and all MQI calls, including initial multicast ones, fail.



#### **Automatic Client Reconnection**

- IBM WebSphere MQ recommends that multicast clients are configured for "Automatic Client Reconnection" as a component of high availability.
- http://publib.boulder.ibm.com/infocenter/wmqv7/v7r 1/index.jsp?topic=%2Fcom.ibm.mq.doc %2Ffa70190\_.htm
- NOTE: If any application explicitly issues an MQDISC, then all multicast subscriptions and object handles are closed.





#### Multicast Message Reliability

- A key element of overcoming transmission failure with multicast is WebSphere MQ's buffering of transmitted data:
  - a history of messages to be kept at the transmitting end of the link.
- This process means that no buffering of messages is required in the putting application process because WebSphere MQ provides the reliability.
- A bigger transmission buffer means that there is more transmission history to be retransmitted if needed, but due to the nature of multicast, 100% assured delivery cannot be supported.





#### Multicast Message Reliability: MSGHIST

- The Multicast message history is controlled in the communication information (COMMINFO) object by the MSGHIST attribute.
- This value is the amount of message history in kilobytes that is kept by the system to handle retransmissions in the case of NACKs (negative acknowledgments).
- A value of 0 gives the least level of reliability.
- The default value is 100 KB.





## Multicast Message Reliability: NSUBHIST

- The WebSphere MQ Multicast new subscription history is controlled in the communication information (COMMINFO) object by the NSUBHIST attribute:
- The new subscriber history controls whether a subscriber joining a publication stream
- receives as much data as is currently available, or
- receives only publications made from the time of the subscription.





# **Multicast Security**

- MQ Multicast does not have in-built security.
- Security checks are handled in the queue manager at MQOPEN time and the MQMD field setting is handled by the client.
- Some applications in the network might not be MQ applications therefore you might need to implement your own security procedures because receiving applications cannot be certain of the validity of context fields.





#### Multicast Security: considerations

There are three security processes to consider:

- Access Control
- Network Security
- Digital Signatures

#### Multicast Security: access control (1)

- Access control in MQ is based on user IDs.
- The user ID of the process making MQI calls is normally used.
- For MQ MQI clients, the server-connection MCA makes MQI calls on behalf of MQ MQI clients.
- You can select an alternative user ID for the server-connection MCA to use for making MQI calls.





## Multicast Security: access control (2)

- The alternative user ID can be associated either with the client workstation, or with anything you choose to organize and control the access of clients.
- The user ID needs to have the necessary authorities allocated to it on the server to issue MQI calls.
- Choosing an alternative user ID is preferable to allowing clients to make MQI calls with the authority of the server-connection MCA.
- See "Access control for clients" http://publib.boulder.ibm.com/infocenter/wmqv7/v7r1/topic/com.ibm.mq.doc/cs11910\_.htm



## Multicast Security: network security (1)

- An isolated network might be a viable security option to prevent fake messages.
- It is possible for an application on the multicast group address to publish malicious messages using native communication functions.
- It is also possible for a client on the multicast group address to receive messages that were intended for other clients on the same multicast group address.



#### Multicast Security: network security (2)

- Isolating the multicast network ensures that only valid clients and applications have access.
- For information about multicast group network addresses, see:
- "Setting the appropriate network for multicast traffic".
- http://publib.boulder.ibm.com/infocenter/wmqv7/v7r 1/topic/com.ibm.mq.doc/ps30200\_.htm



#### Multicast Security: Digital signatures (1)

- A digital signature is formed by encrypting a representation of a message.
- The encryption uses the private key of the signatory and, for efficiency, usually operates on a message digest rather than the message itself.



## Multicast Security: Digital signatures (2)

- Digitally signing a message before an MQPUT is a good security precaution,
  - but this process might have a detrimental effect on performance if there is a large volume of messages.
- For more information on this subject, see "Cryptographic concepts".
- http://publib.boulder.ibm.com/infocenter/wmqv7/v7r 1/topic/com.ibm.mq.doc/sy10490 .htm



#### **Configuration and Testing**

- For more details on the configuration and testing of the new MQ 7.1 multicast function, see the following techdoc:
- Basic configuring and usage of WebSphere MQ
   7.1 Multicast
- IBM Techdoc: 7024903
- http://www.ibm.com/support/docview.wss? rs=171&uid=swg27024903



## Objects needed

- Two objects are needed for MQ Multicast:
  - COMMINFO Communication Information object: it contains the attributes associated with multicast transmission.
  - TOPIC
    Two attributes are needed for multicast:
    - COMMINFO: A reference to a communication information object
    - MCAST: Specifies whether multicast is allowable in the topic tree.



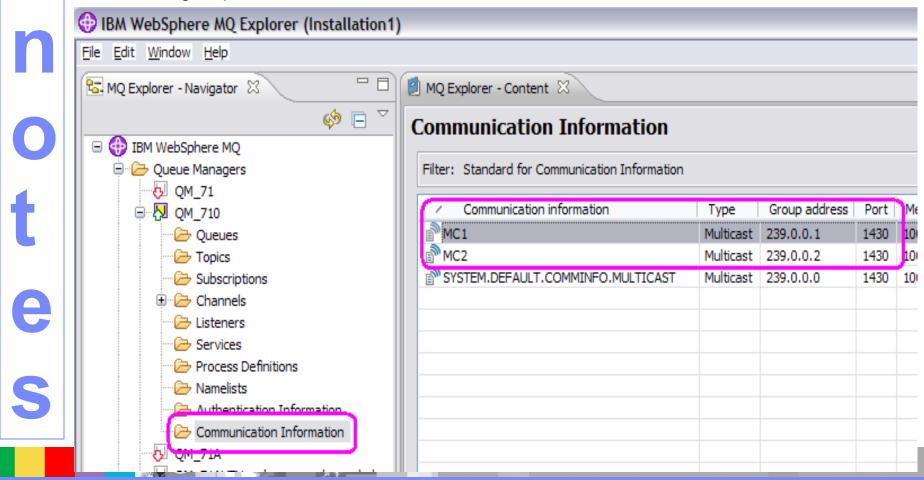
#### **DEFINE COMMINFO**

- Using runmqsc: Example for specifying the group address of 239.0.0.1 for the port 1430 and to enable the communication events:
- DEFINE COMMINFO(MC1) +
   TYPE(MULTICAST) GRPADDR(239.0.0.1) +
   PORT(1430) COMMEV(ENABLED)
- Define another object (to be used later on)
- DEFINE COMMINFO(MC2) + TYPE(MULTICAST) GRPADDR(239.0.0.2) + PORT(1430) COMMEV(ENABLED)



#### **Notes: DEFINE COMMINFO (1)**

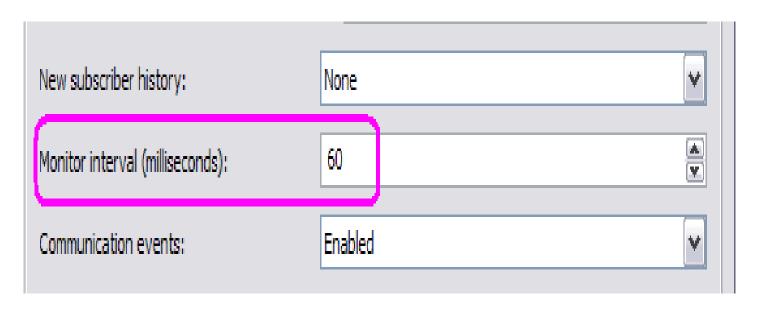
- The following screen capture from the MQ Explorer shows 2 COMMINFO objects:
- MC1 at group address 239.0.0.1
- MC2 at group address 239.0.0.2





#### **Notes: DEFINE COMMINFO (2)**

- Bug in MQ Explorer in 7.1.0.0:
- The prompt for "Monitor interval" incorrectly says that the units are "milliseconds".
- The proper units are "seconds".











#### DEFINE TOPIC – 2 parameters

- New or existing topics in MQ 7.1 can be used with multicast by changing the values of the following TOPIC object parameters:
  - COMMINFO
  - MCAST.

The COMMINFO parameter specifies the name of the multicast communication information object.



#### **DEFINE TOPIC - example**

- Using runmqsc: Example of defining a Topic object that will have only multicast traffic, using the communication object MC1
- DEFINE TOPIC(FRUIT) TOPICSTR('price/fruit') +
   MCAST(ONLY) COMMINFO(MC1)
- For the test of non-multicast, let's define:
- DEFINE TOPIC(FISH) TOPICSTR('price/fish') + <u>MCAST(DISABLED)</u> COMMINFO(MC2)



# **Notes: DEFINE TOPIC (1)**

Tab: Distributed Publish/Subscribe Specify: Multicast: ONLY Communication information: MC2 Hew Topic Change properties Change the properties of the new Topic General Distributed Publish/Subscribe Distributed Publish/Subscribe Cluster Proxy subscription behavior: ٧ First use Publication scope: As parent ٧ Subscription scope: As parent Multicast: Only Communication information: Select... MC2



# Testing using C-source samples (1)

The zip file from the techdoc 7024903 includes the following samples written in C-code:

### amqshacpub.c

- It provides an endless loop in which a publisher publishes 1 message every 2 seconds.
- It includes logic to provide details if the client got disconnected (queue manager ended, for example) and the attempts to try to automatically reconnect.
- It is based on the following sample shipped with MQ: amqsphac.c (high availability client for doing MQPUT, point to point).





# Testing using C-source samples (2)

- amqshacsub.c
- It provides an endless loop in which a subscriber gets published messages.
- It includes logic to provide details if the client got disconnected.
- Based on sample: amqsghac.c (high availability client for doing MQGET, point to point).
- Sample files shipped with MQ are located at:
- Windows:
  - C:\Program Files\IBM\WebSphere MQ\tools\c\Samples
- Linux: /opt/mqm/samp





## Executables / Compiling code

WindowsLinux

amqshacpub.exe
amqshacpub

amqshacsub.exe
amqshacsub

 The source code files have comments in the header that explain how to compile them (Windows and Linux x86 32-bit)

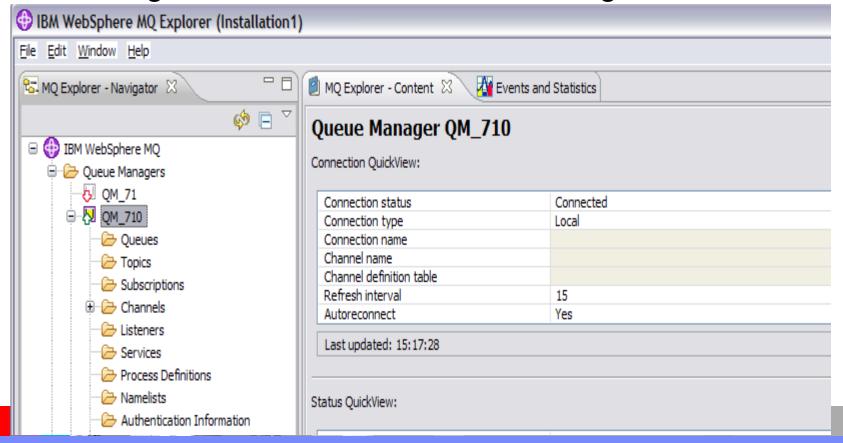
- For Windows the following batch files are provided:
- cl\_hacpub.bat
- cl\_hacsub.bat





# Setup for testing – non multicast (1)

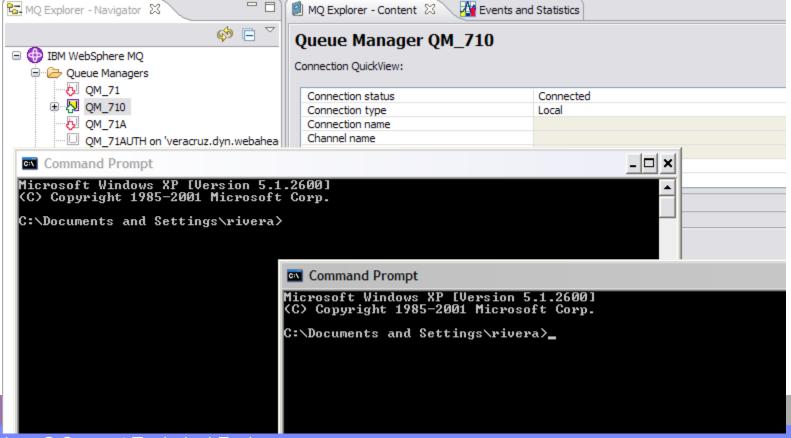
 Use MQ Explorer to show status of queue manager, which in this case, is running.





# Setup for testing – non multicast (2)

 Open 2 windows, one for pub (on the left) and another for sub (on the right)



# Setup for testing – non multicast (3)

- Ensure that the queue manager is running.
- In this case it is called QM\_710 using port 1430.
- In each window:
- 1) Change to the directory where the samples were downloaded and extracted, such as:
- cd C:\MQ\71-multicast
- 2) Setup the environment to use MQ 7.1. In this case, it is in Installation1:
- "C:\Program Files\IBM\WebSphere MQ\_1\bin\setmqenv" -n Installation1
- 3) Setup the MQSERVER variable:
- set MQSERVER=SYSTEM.DEF.SVRCONN/TCP/x.com(1430)



# Running the publisher – non multicast

- Run the publisher for topic 'price/fish' for queue manager 'QM\_710'.
- This topic has MCAST(DISABLED)
- Notice that the publisher will get into an endless loop, publishing 1 message every 2 seconds.
- .\amqshacpub price/fish QM\_710
- Sample amqshacpub start
- target topic is price/fruit
- message <Message 1>
- message <Message 2>



## Running the subscriber – non multicast

- Run the subscriber for topic 'price/fish' for queue manager 'QM\_710'.
- This topic has MCAST(DISABLED)
- Notice that the subscriber will get into an endless loop, displaying each received message.
- .\amqshacsub price/fish QM\_710
- Sample amqshacsub start
- message <Message 1>
- message <Message 2>



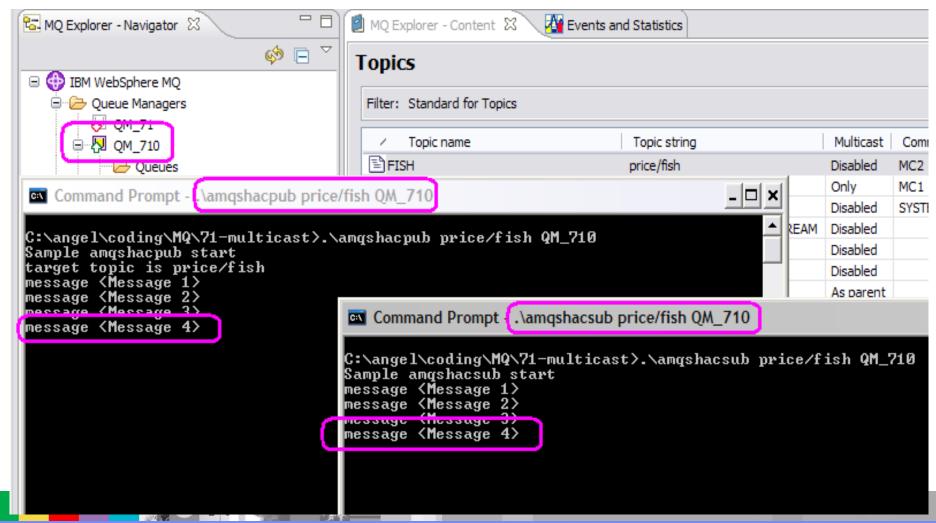
# MQ Explorer + pub + sub, non multicast

- The picture in the next slide shows 3 windows:
- On the upper portion: The MQ Explorer showing that QM\_710 is <u>running</u>.
- On the left: Window-1 with the publisher.
   So far, 4 messages where published.
- On the right: Window-2 with the subscriber, showing the 4 received messages.





# MQ Explorer + pub + sub (2), non multi





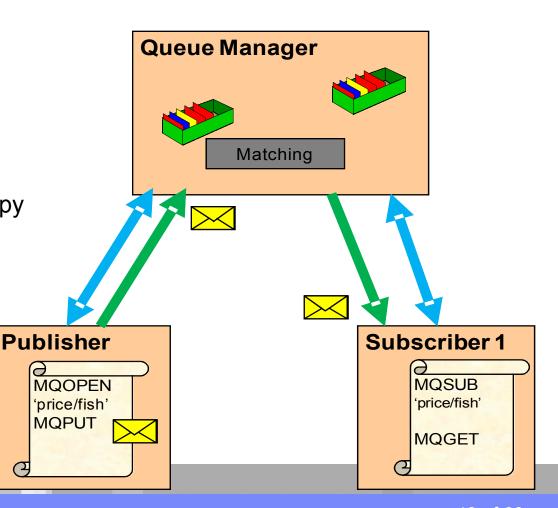
# Queue manager handles messages

#### Blue Arrows:

- Control and feedback

#### **Green Arrows:**

- Messages flow through Queue
   Manager
- Subscriber matching is done by Queue Manager on publish
- Each subscriber sent its own copy of message





# Stop the queue manager (1)

- From MQ Explorer, stop the queue manager by specifying:
- (x) Immediate
- (x) Instruct reconnectable clients to reconnect
- From a line command, issue the command to end the queue manager requesting:
- an immediate termination and
- alert the reconnectable clients to try to reconnect:
- endmqm -ir QMgrName





# Stop the queue manager (2)

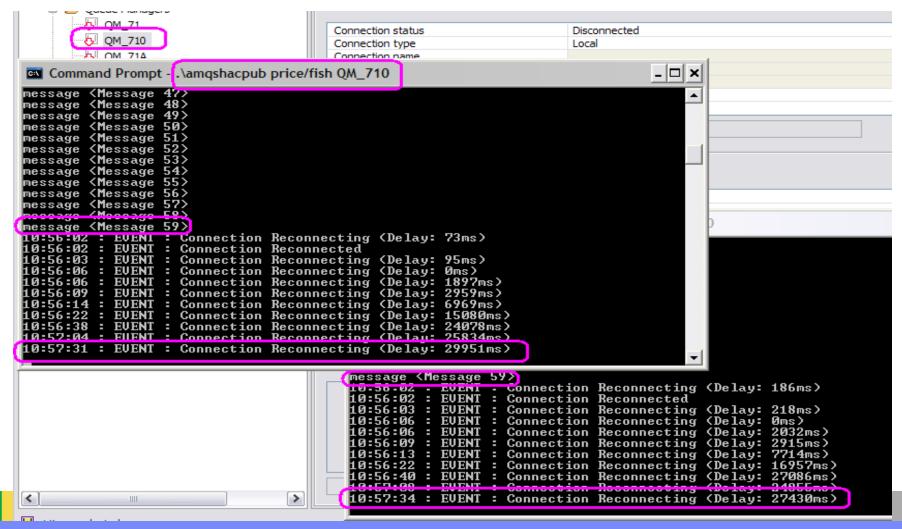
Notice the following for the next slide:

- The queue manager is no longer running.
- The publisher's last message is 59.
- The subscriber's last message is also 59 and it is trying to reconnect.
- There are NO new messages published since the publisher detected the disconnection.
- The applications are displaying they attempts to reconnect:
  - EVENT : Connection Reconnecting (Delay: XXXms)





# No more messages are handled





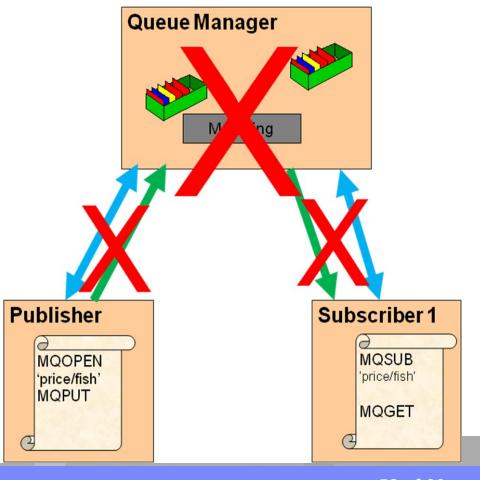
# Queue manager is down

#### Blue Arrows:

Control and feedback – not exchanged

#### Green Arrows:

- Publisher cannot send messages to queue manager
- Subscriber cannot receive messages.





# Setup for testing – multicast (1)

- Similar setup as the non-multicast setup.
- Topic to be used is FRUIT ('price/fruit') which has MCAST(ONLY)
- On the publisher window run: .\amqshacpub price/fruit QM\_710
- On the subscriber window run: .\amqshacsub price/fruit QM 710





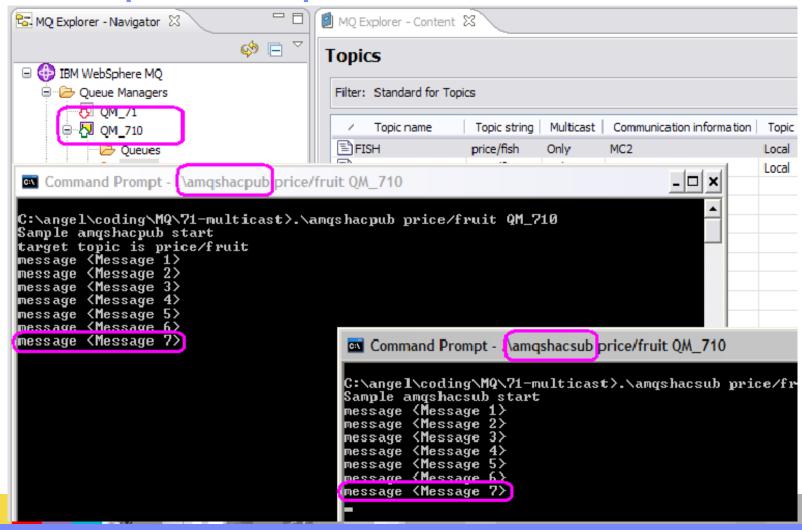
# MQ Explorer + pub + sub, multicast

- The picture in the next slide shows 3 windows:
- On the upper portion: The MQ Explorer showing that QM\_710 is <u>running</u>.
- On the left: Window-1 with the publisher.
   So far, 7 messages where published.
- On the right: Window-2 with the subscriber, showing the 7 received messages.





# MQ Explorer + pub + sub - multicast





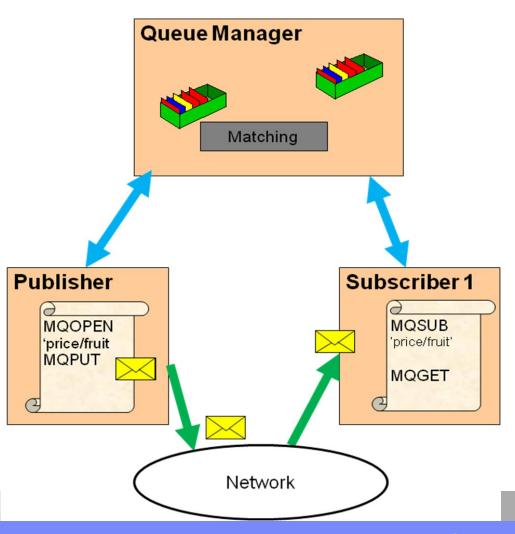
# Qmgr does NOT handle messages

#### Blue Arrows:

- Control and feedback (events)

#### **Green Arrows:**

- Messages flow through Network
- Subscriber matching is done by MQ
   Client
- The queue manager is NOT involved with the handling of messages







# Is multicast really being used?

- Question:
- Is there an easy way to prove that the network multicast is being used and that the messages are NOT being handled by the queue manager?
- Answer:
- Yes, let's terminate the queue manager ...
- ... and the messages keep flowing.
- This proves that the queue manager is not involved with the distribution of multicast messages.
- Stop the queue manager: endmqm -ir QmgrName





## Clients detected the disconnect (1)

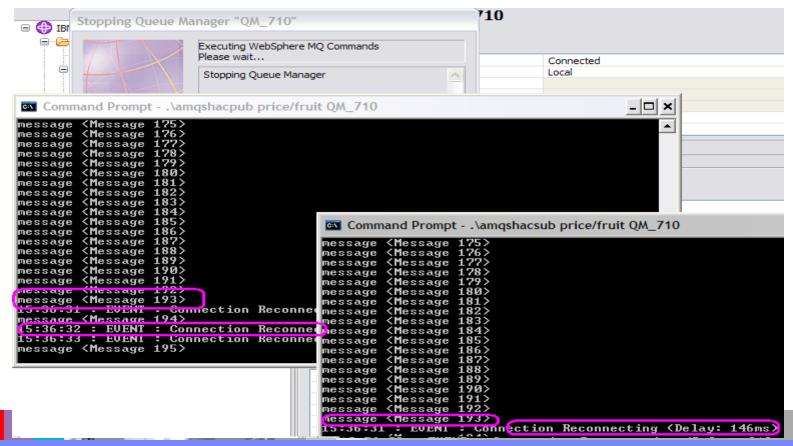
- Notice in the next slide that the publisher and subscriber will detect that they cannot communicate with the queue manager anymore and they will attempt to reconnect.
- Both will display a notice indicating:
- EVENT : Connection Reconnection (Delay: x ms)
- This means that the MQ client code being used by the application is going to wait x milliseconds before trying to reconnect again with the queue manager.





## Clients detected the disconnect (2)

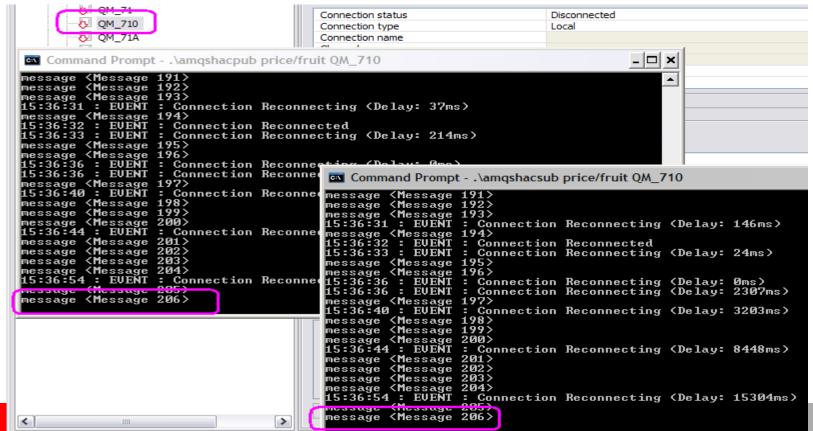
 The message counter was 193 when the MQ client code detected the disconnection





# Client continued to handle messages (1)

Even though the queue manager is NOT running, the applications are STILL handling messages !!!





# Client continued to handle messages (2)

- The queue manager is NOT running anymore, but the publisher and subscriber continued handling messages!!
- In the previous slide the message counter is now 206 (the counter was at 193 when the queue manager terminated).
- This test proves that these samples are using the network multicast to send messages that do NOT require the intervention of the queue manager.





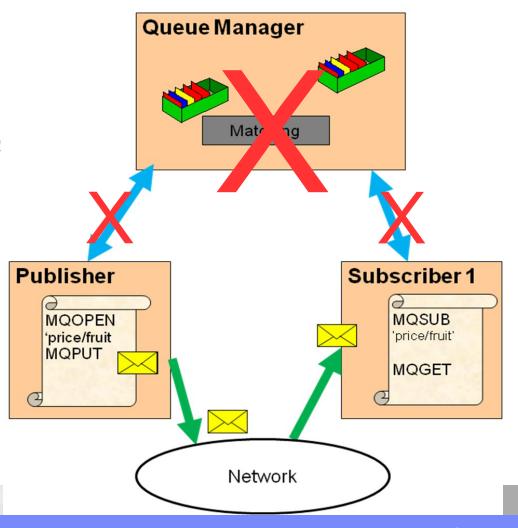
# Qmgr does NOT handle messages

#### Blue Arrows:

- Control and feedback

#### Green Arrows:

- Messages flow through Network
- Subscriber matching is done by MQ Client
- The queue manager is NOT involved with the handling of messages







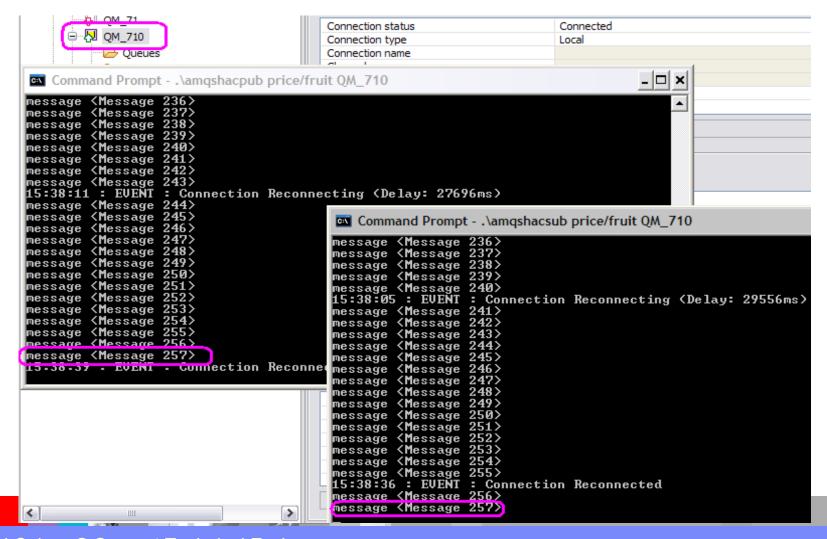
## Restarting the queue manager

- Restart the queue manager.
- From MQ Explorer, select the queue manager, right click and select: Start
- From the command line: strmqm QmgrName
- Notice in the next slide that the queue manager is running and the publisher and subscriber were able to reconnect.
- Still both are handling messages!
- The message counter is 257 (the counter was at 193 when the queue manager terminated).





### Client continue to handle messages



# Testing with SupportPac MA01 – Q program

- A useful SupportPac for testing the multicast capability is MA01 "Q program".
- It was refreshed recently to include Pub/Sub and reconnection.
- Chapter 6 of the mentioned techdoc 7024903 describes the details on how to use the q.exe in Windows to perform publish and subscribe with multicast.
- http://www.ibm.com/support/docview.wss? uid=swg24000647
- MA01: WebSphere MQ Q program



## Monitoring Pub/Sub Events

- Pub/Sub Events can be generated into: SYSTEM.ADMIN.PUBSUB.EVENT
- See "Chapter 7: Configuration to receive pub/sub events" from the techdoc 7024903
- Basic configuring and usage of WebSphere MQ 7.1 Multicast
- http://www.ibm.com/support/docview.wss?rs=171&uid=swg27024903
- Add the following plugin to the MQ Explorer
- http://www.ibm.com/support/docview.wss? rs=171&uid=swg24011617
- MS0P: WebSphere MQ Explorer Configuration and Display Extension Plug-ins



### References

- Initial multicast concepts
  - http://publib.boulder.ibm.com/infocenter/wmqv7/v7r1/topic/com.ibm.mq.doc/ps30004\_.htm
- Getting started with multicast
  - http://publib.boulder.ibm.com/infocenter/wmqv7/v7r1/topic/com.ibm.mq.doc/ps30170\_.htm
- Multicast topic topology issues
  - http://publib.boulder.ibm.com/infocenter/wmqv7/v7r1/topic/com.ibm.mq.doc/ps30230\_.htm
- DEFINE COMMINFO
  - http://publib.boulder.ibm.com/infocenter/wmqv7/v7r1/topic/com.ibm.mq.doc/sc14410 .htm
- Testing multicast applications on a non-multicast network
  - http://publib.boulder.ibm.com/infocenter/wmqv7/v7r1/topic/com.ibm.mq.doc/ps30190\_.htm



### Additional WebSphere Product Resources

- Learn about upcoming WebSphere Support Technical Exchange webcasts, and access previously recorded presentations at: http://www.ibm.com/software/websphere/support/supp\_tech.html
- Discover the latest trends in WebSphere Technology and implementation, participate in technically-focused briefings, webcasts and podcasts at: http://www.ibm.com/developerworks/websphere/community/
- Join the Global WebSphere Community: http://www.websphereusergroup.org
- Access key product show-me demos and tutorials by visiting IBM Education Assistant: http://www.ibm.com/software/info/education/assistant
- View a webcast replay with step-by-step instructions for using the Service Request (SR) tool for submitting problems electronically: http://www.ibm.com/software/websphere/support/d2w.html
- Sign up to receive weekly technical My Notifications emails: http://www.ibm.com/software/support/einfo.html



# Connect with us!

### 1.Get notified on upcoming webcasts

Send an e-mail to wsehelp@us.ibm.com with subject line "wste subscribe" to get a list of mailing lists and to subscribe

### 2.Tell us what you want to learn

Send us suggestions for future topics or improvements about our webcasts to wsehelp@us.ibm.com

### 3.Be connected!

Connect with us on Facebook
Connect with us on Twitter





### **Questions and Answers**

