

IBM Software
WebSphere

Discovering the value of WebSphere MQ v7.5

Lab Exercises



IBM

An IBM Proof of Technology

PoT.WebSphere.13.4.106.00

Contents

LAB 1	INTRODUCTION TO WEBSPHERE MQ.....	10
	1.1 CREATE A QUEUE MANAGER	10
	1.2 TESTING BASIC FUNCTIONALITY	10
	1.3 GROUPING QUEUE MANAGERS	10
	1.4 DISTRIBUTED QUEUING.....	10
	1.5 LAB CLEANUP	10
LAB 2	CONFIGURING THE WEBSPHERE MQ JMS PROVIDER	75
	2.1 CREATE ADMINISTERED OBJECTS USING MQ EXPLORER.....	75
	2.2 CREATE A CONNECTION FACTORY FOR WEBSPHERE MQ.....	75
	2.3 CREATE A DESTINATION FOR THE JMS APPLICATION TO PUT A MESSAGE TO	75
	2.4 WRITING A JMS MESSAGE USING A JAVA SAMPLE PROGRAM	75
LAB 3	PUBLISH / SUBSCRIBE ADMINISTRATION	95
	3.1 LAB OVERVIEW.....	95
	3.2 USING WEBSPHERE MQ EXPLORER TO CREATE AND DISPLAY INFORMATION	95
	3.3 A FIRST LOOK AT THE MQ EXPLORER PUB/SUB TEST TOOLS	95
	3.2 ADMINISTERED SUBSCRIPTIONS	95
	3.5 TESTING PUBLICATIONS AND SUBSCRIPTIONS FROM THE COMMAND LINE	95
LAB 4	WEBSPHERE MQ SECURITY	124
	4.1 WEBSPHERE MQ COURSE OBJECTIVES	124
	4.2 REVIEW THE SECURITY ON THE SYSTEM.....	124
	4.3 USING RUNAS TO CHANGE RUNTIME AUTHORITY	124
	4.4 POINT TO POINT (QUEUE) SECURITY	124
	4.5 PUB/SUB (TOPIC) SECURITY.....	124
	4.6 USING THE MQ EXPLORER TO MANAGE SECURITY.....	124
LAB 5	WEBSPHERE MQ – HTTP BRIDGE	144
	5.1 CREATE THE REQUIRED QUEUES	144
	5.2 START THE HTTP LISTENER	144
	5.3 TEST THE LISTENER USING CURL.....	144
	5.4 USING THE HTTP BRIDGE WITH JAVASCRIPT	144
APPENDIX A.	RESOURCES	164
APPENDIX B.	NOTICES	165
APPENDIX C.	TRADEMARKS AND COPYRIGHTS	167

THIS PAGE INTENTIONALLY LEFT BLANK

Overview

This guide and associated VMware image helps you get started with WebSphere MQ V7.5 on distributed platforms.

You can use WebSphere® MQ to enable applications to communicate at different times and in many diverse computing environments.

What is WebSphere MQ?

- WebSphere MQ is messaging for applications. It sends messages across networks of diverse components. Your application connects to WebSphere MQ to send or receive a message. WebSphere MQ handles the different processors, operating systems, subsystems, and communication protocols it encounters in transferring the message. If a connection or a processor is temporarily unavailable, WebSphere MQ queues the message and forwards it when the connection is back online.
- An application has a choice of programming interfaces, and programming languages to connect to WebSphere MQ.
- WebSphere MQ is *messaging* and *queuing* middleware, with *point-to-point*, *publish/subscribe*, and *file transfer* modes of operation. Applications can publish messages to many subscribers over *multicast*.

Messaging

Programs communicate by sending each other data in messages rather than by calling each other directly.

Queuing

Messages are placed on queues, so that programs can run independently of each other, at different speeds and times, in different locations, and without having a direct connection between them.

Point-to-point

Applications send messages to a queue, or to a list of queues. The sender must know the name of the destination, but not where it is.

Publish/subscribe

Applications publish a message on a topic, such as the result of a game played by a team. WebSphere MQ sends copies of the message to applications that subscribe to the results topic. They receive the message with the results of games played by the team. The publisher does not know the names of subscribers, or where they are.

Multicast

Multicast is an efficient form of publish/subscribe messaging that scales to many subscribers. It transfers the effort of sending a copy of a publication to each subscriber from WebSphere MQ to the network. Once a path for the publication is established between the publisher and subscriber, WebSphere MQ is not involved in forwarding the publication.

File transfer

Files are transferred in messages. WebSphere MQ File Transfer Edition manages the transfer of files and the administration to set up automated transfers and log the results. You can integrate the file transfer with other file transfer systems, with WebSphere MQ messaging, and the web.

Telemetry

WebSphere MQ Telemetry is messaging for devices. WebSphere MQ connects device and application messaging together. It connects the internet, applications, services, and decision makers with networks of instrumented devices. WebSphere MQ Telemetry has an efficient messaging protocol that connects a large numbers of devices over a network. The messaging protocol is published, so that it can be incorporated into devices. You can also develop device programs with one of the published programming interfaces for the protocol.

What can it do for me?

WebSphere MQ sends and receives data between your applications, and over networks.

Message delivery is *assured* and *decoupled* from the application. Assured, because WebSphere MQ exchanges messages transactionally, and decoupled, because applications do not have to check that messages they sent are delivered safely.

You can secure message delivery between queue managers with SSL/TLS.

With Advanced Message Security (AMS), you can encrypt and sign messages between being put by one application and retrieved by another.

Application programmers do not need to have communications programming knowledge.

How do I use it?

- Create and manage WebSphere MQ with the WebSphere MQ Explorer GUI or by running commands from a command window or application.
- Program applications to send and receive messages by calling one of the programming interfaces. Programming interfaces are provided for different languages, and include the standard JMS programming interface, and classes for the Windows communication foundation.
- Send and receive WebSphere MQ messages from browsers with the HTTP protocol.

How does it work?

- An administrator creates and starts a queue manager with commands. Subsequently, the queue manager is usually started automatically when the operating system boots. Applications, and other queue managers can then connect to it to send and receive messages.
- An application or administrator creates a queue or a topic. Queues and topics are objects that are owned and stored by a *queue manager*.
- When your application wants to transfer data to another application, it puts the data into a message. It puts the message onto a queue, or publishes the message to a topic. There are three main ways that the message can be retrieved:
 - A point-to-point application connected to the same queue manager retrieves the message from the same queue.
 - For example, an application puts messages on a queue as way of storing temporary or persistent data. A second example: An application that shares data with another application that is running in a different process.
 - A point-to-point application connected to another queue manager retrieves the same message from a different queue.
 - Applications communicate with each other by exchanging messages on queues. The main use of WebSphere MQ is to send or exchange messages. One application puts a message on a queue on one computer, and another application gets the same message from another queue on a different computer. The queue managers on the two computers work together to transfer the message from the first queue to the second queue. The applications do not communicate with each other, the queue managers do.
 - A subscriber application connected to any queue manager retrieves messages on common topics.
 - A publisher application creates a message and publishes it to a topic on one computer. Any number of subscriber applications subscribe to the same topic on different computers. WebSphere MQ delivers the publication to queues that belong to the queue managers the subscribers are connected to. The subscribers retrieve the message from the queues.
- *MQ channels* connect one queue manager to another over a network. You can create MQ channels yourself, or a queue manager in a cluster of queue managers creates MQ channels when they are needed.
- You can have many queues and topics on one queue manager.
- You can have more than one queue manager on one computer.

- An application can run on the same computer as the queue manager, or on a different one. If it runs on the same computer, it is a WebSphere MQ server application. If it runs on a different computer, it is a WebSphere MQ client application. Whether it is WebSphere MQclient or server makes almost no difference to the application. You can build a client/server application with WebSphere MQ clients or servers.

What tools and resources come with WebSphere MQ?

- Control commands, which are run from the command line. You create, start, and stop queue managers with the control commands. You also run WebSphere MQ administrative and problem determination programs with the control commands.
- WebSphere MQ script commands (MQSC), which are run by an interpreter. Create queues and topics, configure, and administer WebSphere MQ with the commands. Edit the commands in a file, and pass the file to the **runmqsc** program to interpret them. You can also run the interpreter on one queue manager, which sends the commands to a different computer to administer a different queue manager.
- The Programmable Command Format (PCF) commands, which you call in your own applications to administer WebSphere MQ. The PCF commands have the same capability as the script commands, but they are easier to program.
- Sample programs.
- On Windows and Linux x86 and x86-64 platforms, where you can run the following utilities:
 - The WebSphere MQ Explorer. The explorer does the same administrative tasks as the script commands, but is much easier to use interactively.
 - The *Postcard* application to demonstrate messaging and verify your installation.
 - Tutorials.

Icons

The following symbols appear in this document at places where additional guidance is available.

Icon	Purpose	Explanation
	Important!	This symbol calls attention to a particular step or command. For example, it might alert you to type a command carefully because it is case sensitive.
	Information	This symbol indicates information that might not be necessary to complete a step, but is helpful or good to know.
	Trouble-shooting	This symbol indicates that you can fix a specific problem by completing the associated troubleshooting information.

Lab 1 Introduction to WebSphere MQ

In this lab you will be introduced to the WebSphere® MQ Explorer, the primary interface for administering a WebSphere MQ environment. You will have an opportunity to see how easy it is to create various MQ objects, view their status and manipulate them.

You will also exercise some basic command line facilities that will allow you to place messages onto queues and remove messages from queues.

1.1 Create a Queue Manager

Before you can do any useful work in a WebSphere MQ environment you must have a queue manager present.

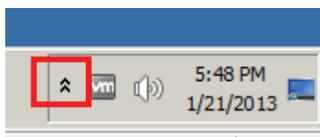
1. Once the Windows VMware image starts up, you should have a desktop that looks similar to the screen capture below.



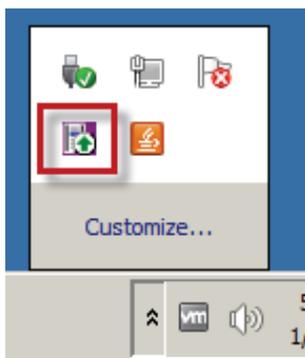
- __2. Sign in as **Administrator** by double-clicking on the icon.
- __3. Enter passw0rd for the password and click the right arrow.



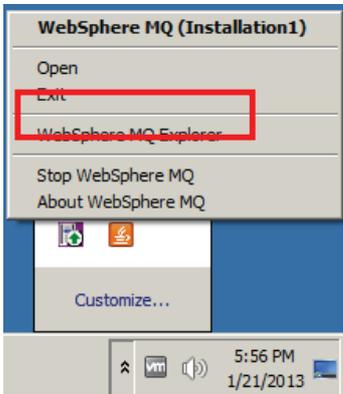
- __4. Click the up arrows to open the hidden taskbar..



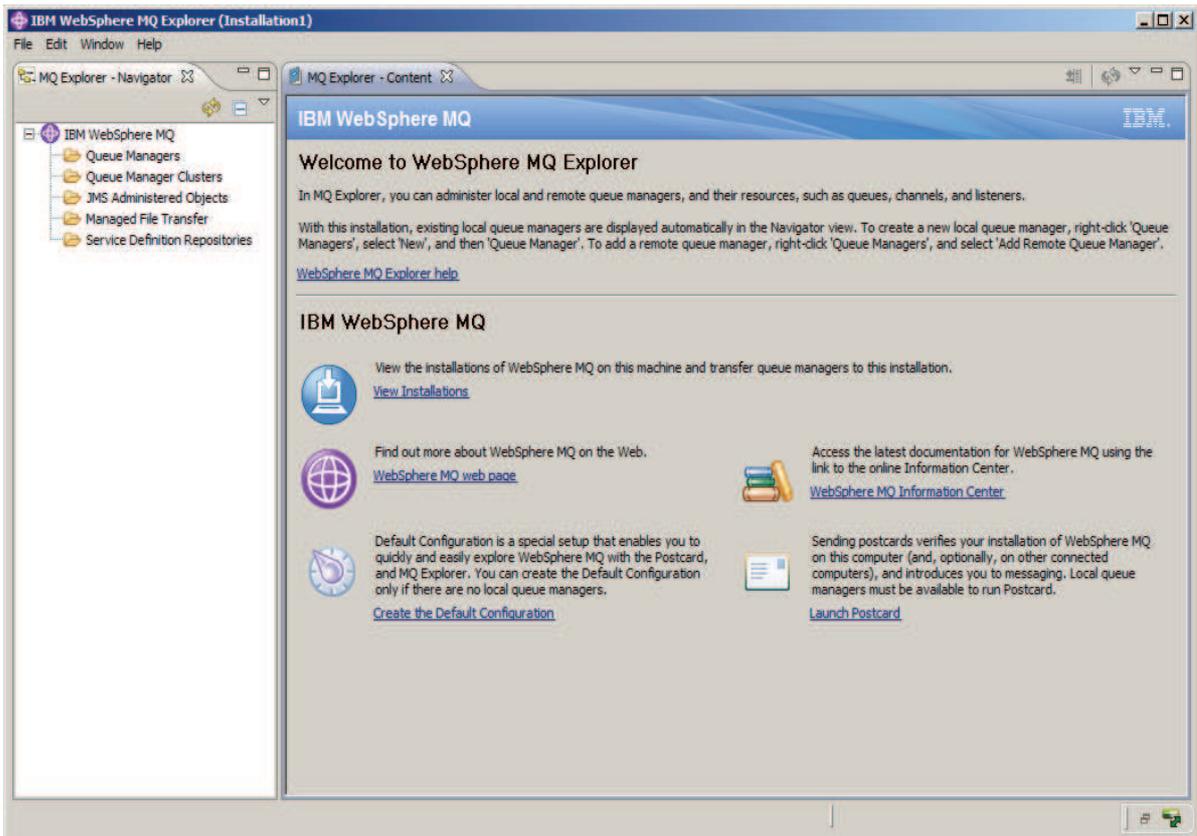
- __5. The indicated icon represents WebSphere MQ on this system.



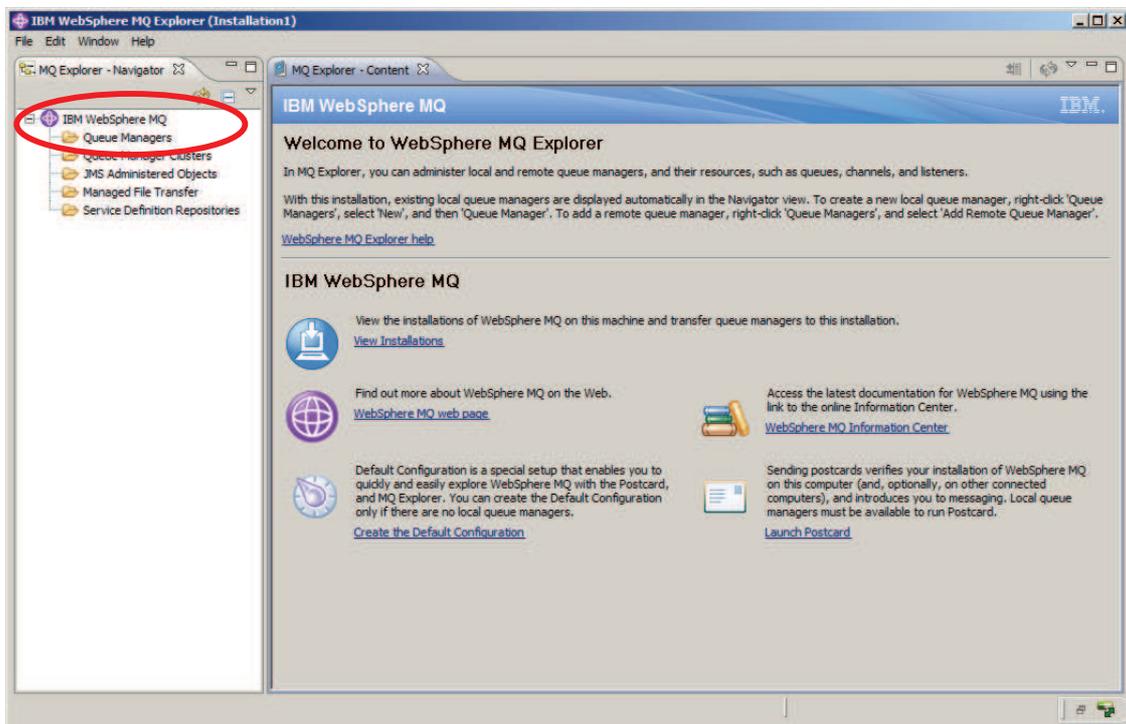
- 6. Start the WebSphere MQ Explorer by **right-clicking** on the icon and selecting **WebSphere MQ Explorer**.



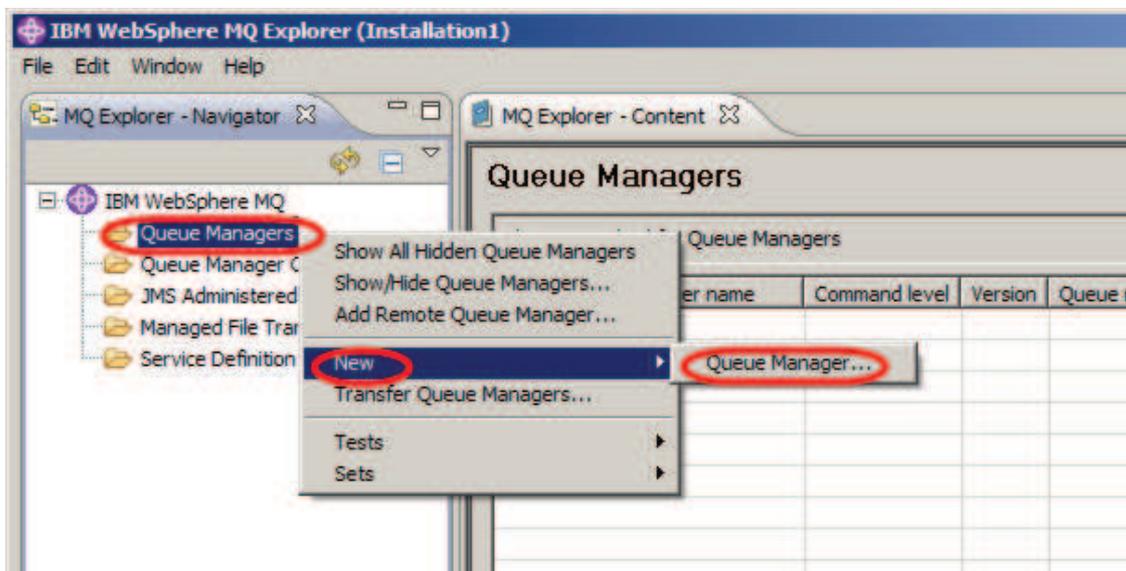
- 7. The welcome screen provides a nice selection of resources for the product. Note the various options on the Welcome screen and explore them if you would like. The first time you launch WebSphere MQ Explorer after an install of WebSphere MQ this Welcome screen will be displayed automatically.



- __8. The left-hand pane in WebSphere MQ Explorer is called the **Navigation pane**. In the navigation pane, locate the **Queue Managers** folder.



- __9. Right-click on **Queue Managers** and select **New** then **Queue Manager**.



- __10. Create a new queue manager and name it **WMQ75**. Check the box to make this the **default queue manager**. Note that the use of a default queue manager is not recommended in a production environment as it allows requests from a program or command to complete successfully without the inclusion of a queue manager name. We are using this option here in the labs as a convenience to reduce typing for you.

Important! Default Queue Manager



Important!!!! Be sure to check the box that indicates this is the default queue manager!!!

If you do not do this you will have issues in future labs!!!!!!

Specify **SYSTEM.DEAD.LETTER.QUEUE** as the dead letter queue, and then click the **Finish** button.

Create Queue Manager
Queue Manager
Enter basic values

Queue manager name: WMQ75

Make this the default queue manager

Default transmission queue:

Dead-letter queue: SYSTEM.DEAD.LETTER.QUEUE

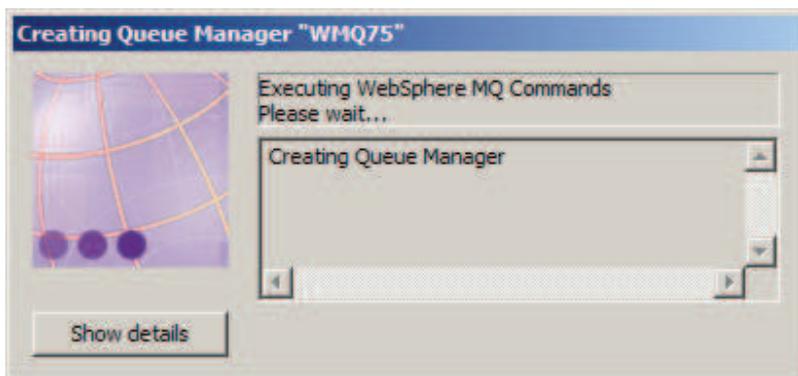
Max handle limit: 256

Trigger interval: 999999999

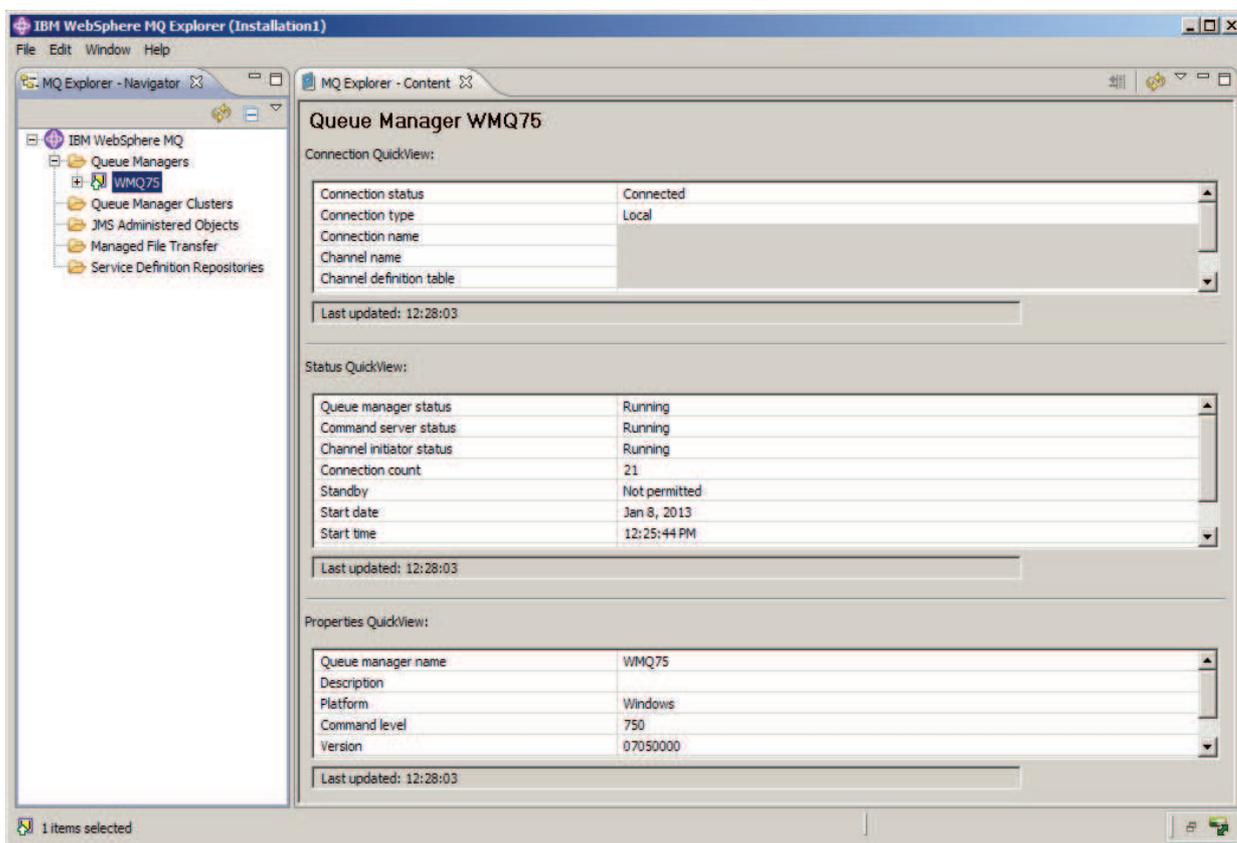
Max uncommitted messages: 10000

? < Back Next > Finish Cancel

- __11. The create queue manager process takes a few seconds during which time the following panel is displayed...



- __12. The newly created queue manager is displayed in the WebSphere MQ Explorer.
- __13. Click on queue manager WMQ75. In the Content pane on the right-hand side you can see details about the queue manager, such as its status, properties, etc.



This concludes this portion of Lab 1.

1.2 Testing Basic Functionality

In this section of the lab you will create a local queue, place a test message in the queue, browse the message, clear it from the queue and finally delete the queue.

- __14. Expand the new **WMQ75** queue manager by clicking on the “+” symbol in front of it in the navigation tree.



- __15. Observe the tree structure that is displayed in the Navigator pane. There are separate folders for *Queues*, *Topics* and *Subscriptions*, *Channels*, just to name a few.

Queue Manager WMQ75

Connection QuickView:

Connection status	Connected
Connection type	Local
Connection name	
Channel name	
Channel definition table	

Last updated: 12:29:48

Status QuickView:

Queue manager status	Running
Command server status	Running
Channel initiator status	Running
Connection count	21
Standby	Not permitted
Start date	Jan 8, 2013
Start time	12:25:44 PM

Last updated: 12:29:48

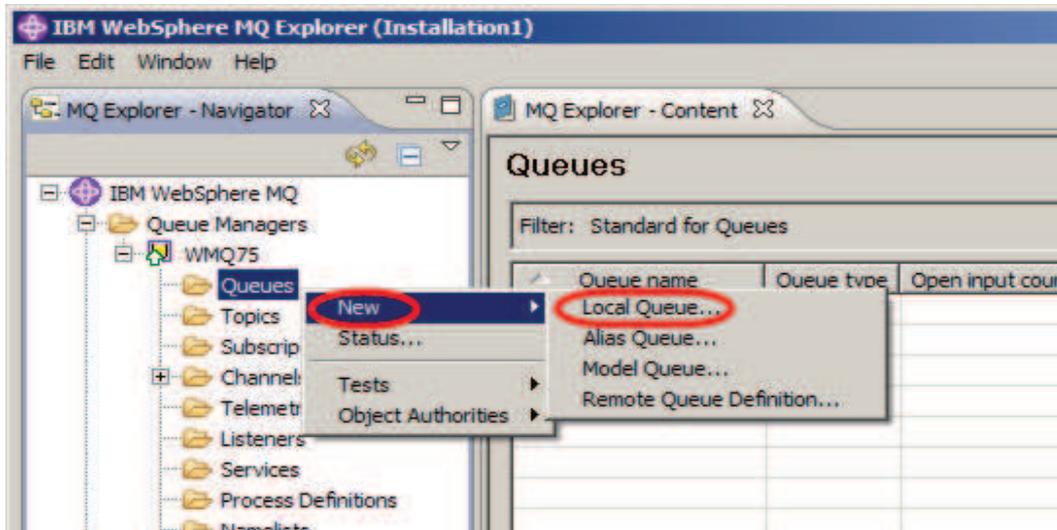
Properties QuickView:

Queue manager name	WMQ75
Description	
Platform	Windows
Command level	750
Version	07050000

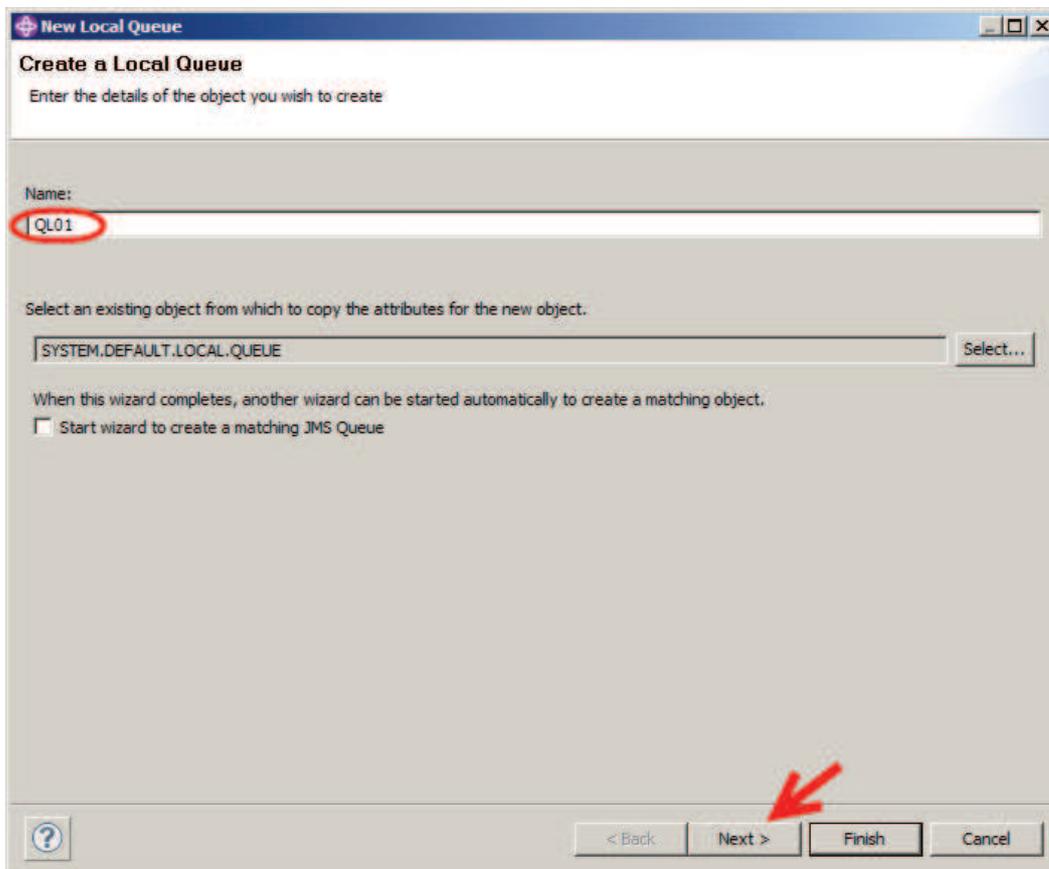
Last updated: 12:29:48

1 items selected

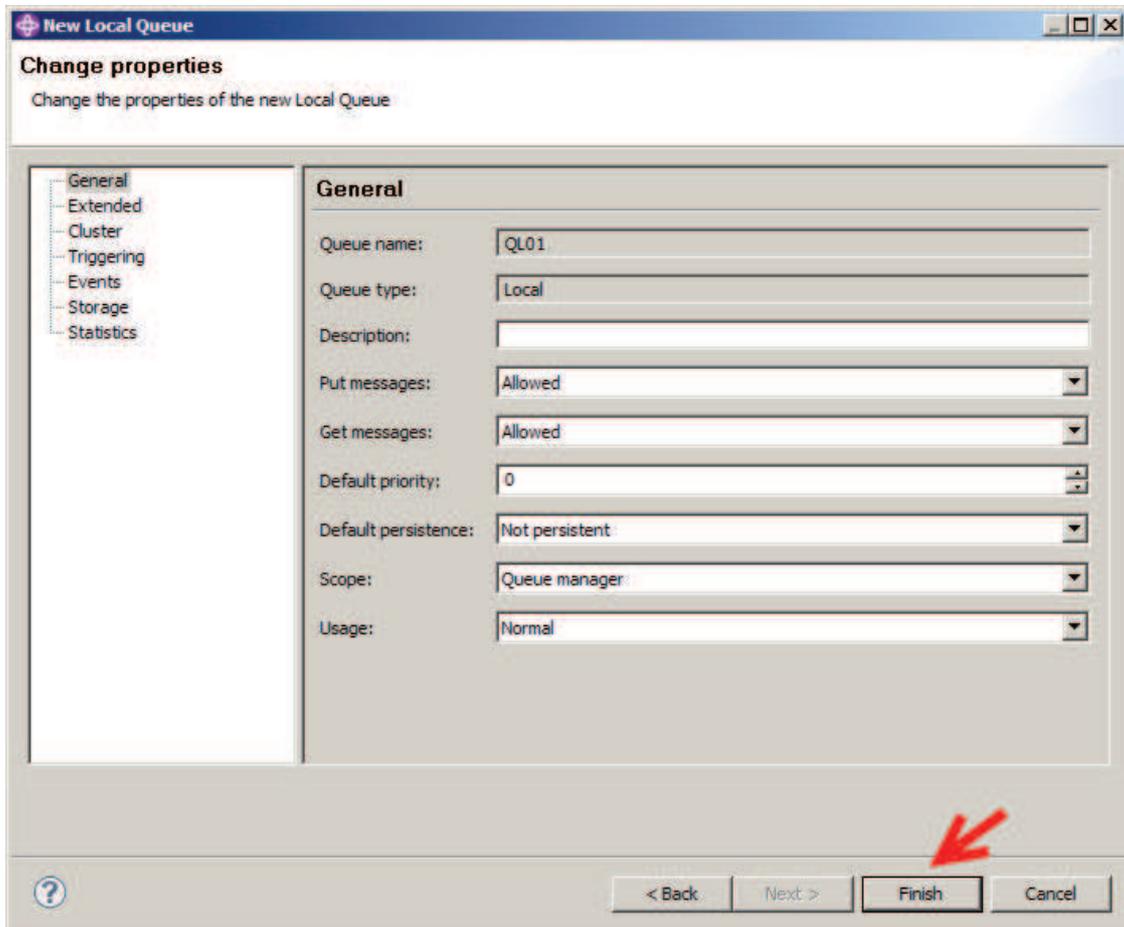
- __16. Now you will create a new queue. Within the WMQ75 queue manager **right-click** on **Queues** then select **New** then **Local Queue**.



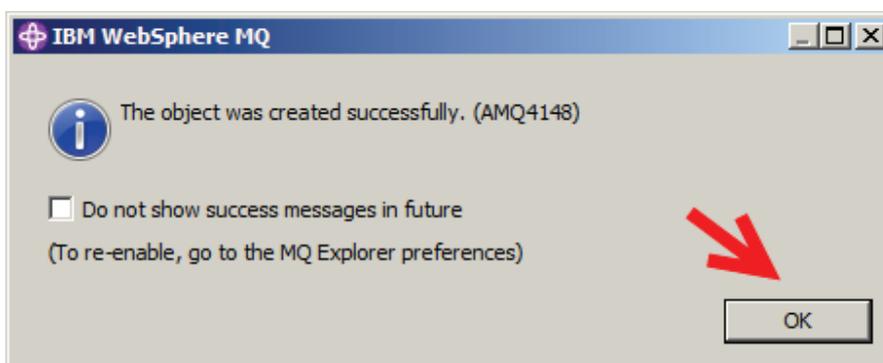
- __17. Name the new queue **QL01** then click on the **Next** button.



- __18. Note the variety of tabs on the left. There are many characteristics or properties for a queue. Explore some of them if so desired. In most cases you can utilize the defaults. For purposes of this lab you can accept all of the defaults – click **Finish** to create the queue.

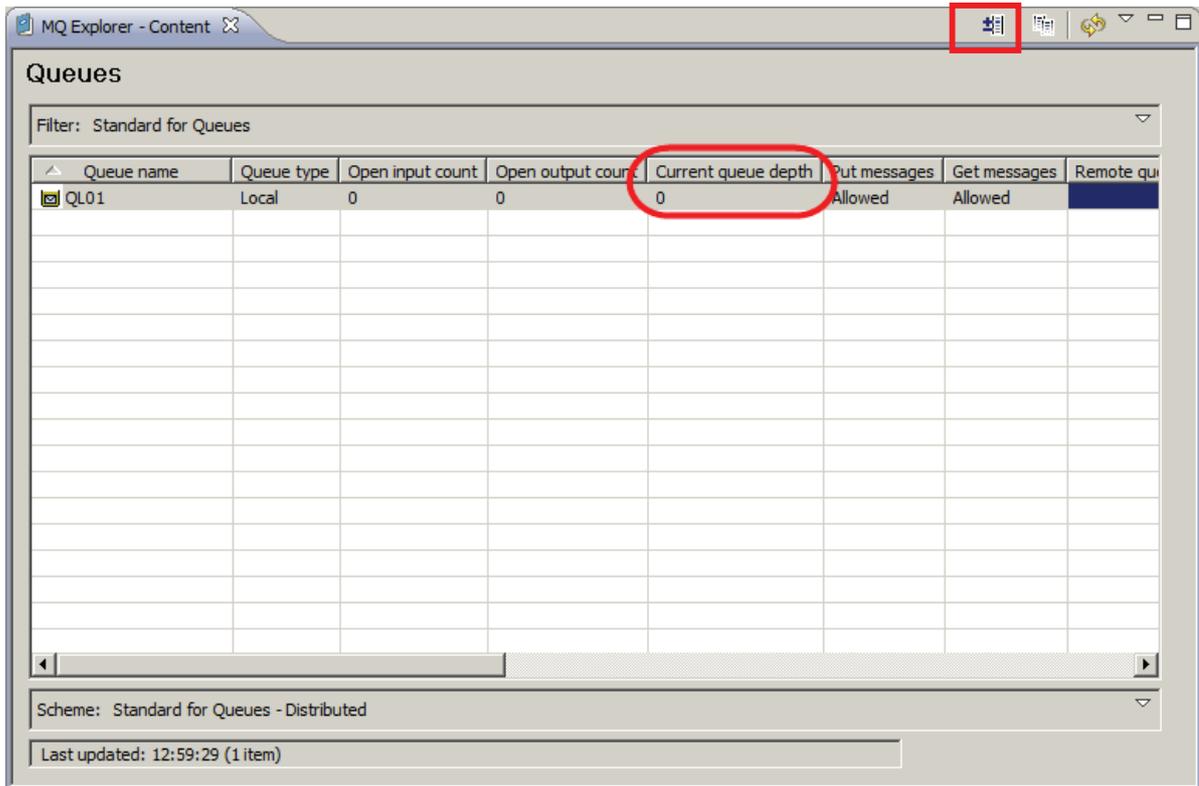


- __19. Dismiss the completion panel by clicking on the **OK** button.

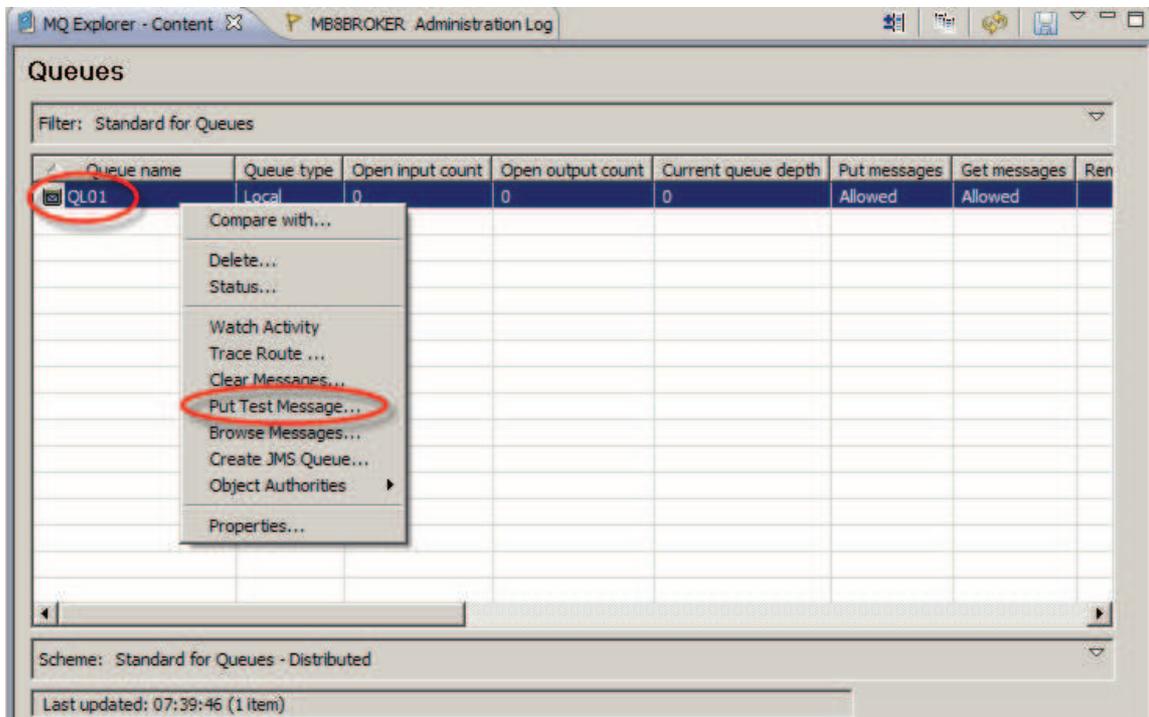


- __20. If the **SYSTEM.*** queues are displayed, you may click the toggle button in the top right corner to hide the **SYSTEM.*** queues.

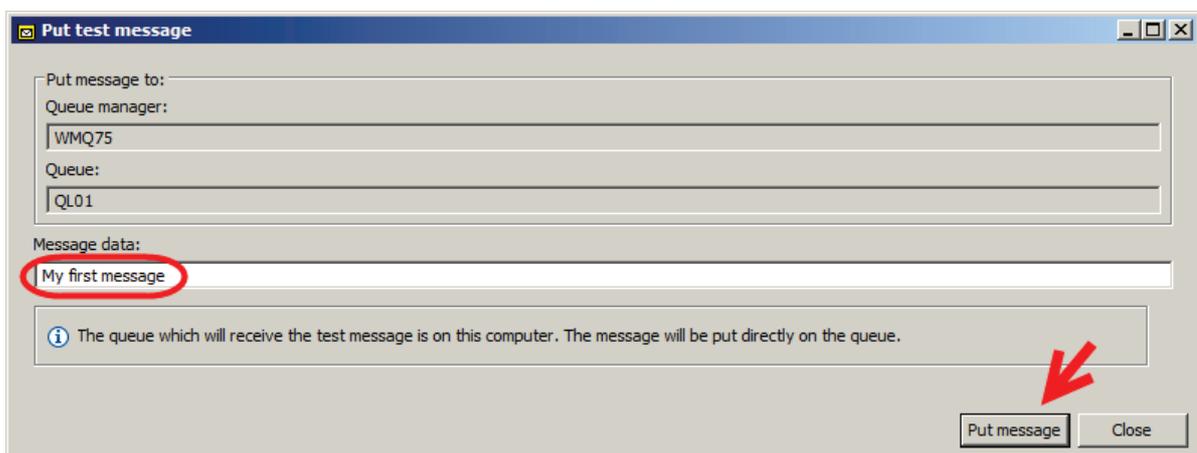
In the Content pane on the right-hand side the newly created queue is displayed along with its various properties. The **Current queue depth** property is a measure of how many messages are currently in the queue. Note that the current queue depth is 0 as you would expect since you just created the queue.



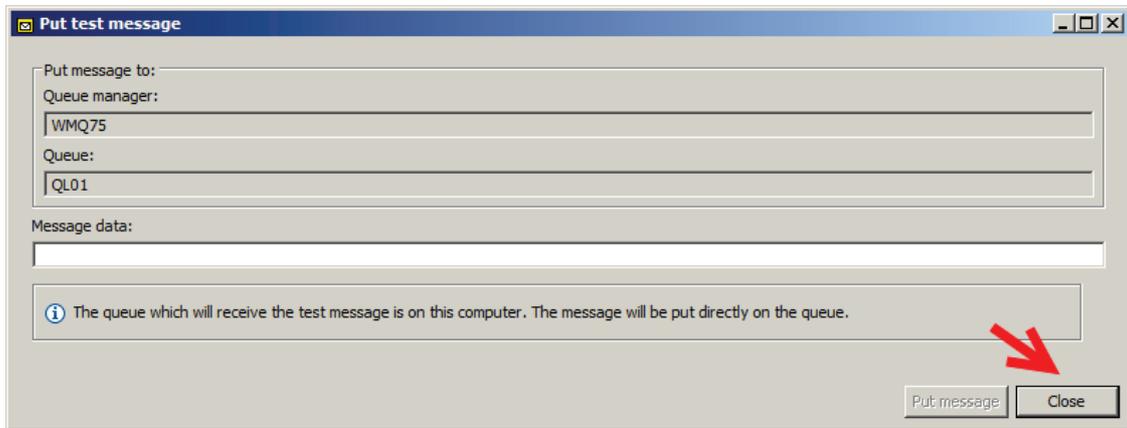
- __21. Next you will place a message in the queue. **Right-click** on the **QL01** queue and select **Put Test Message**.



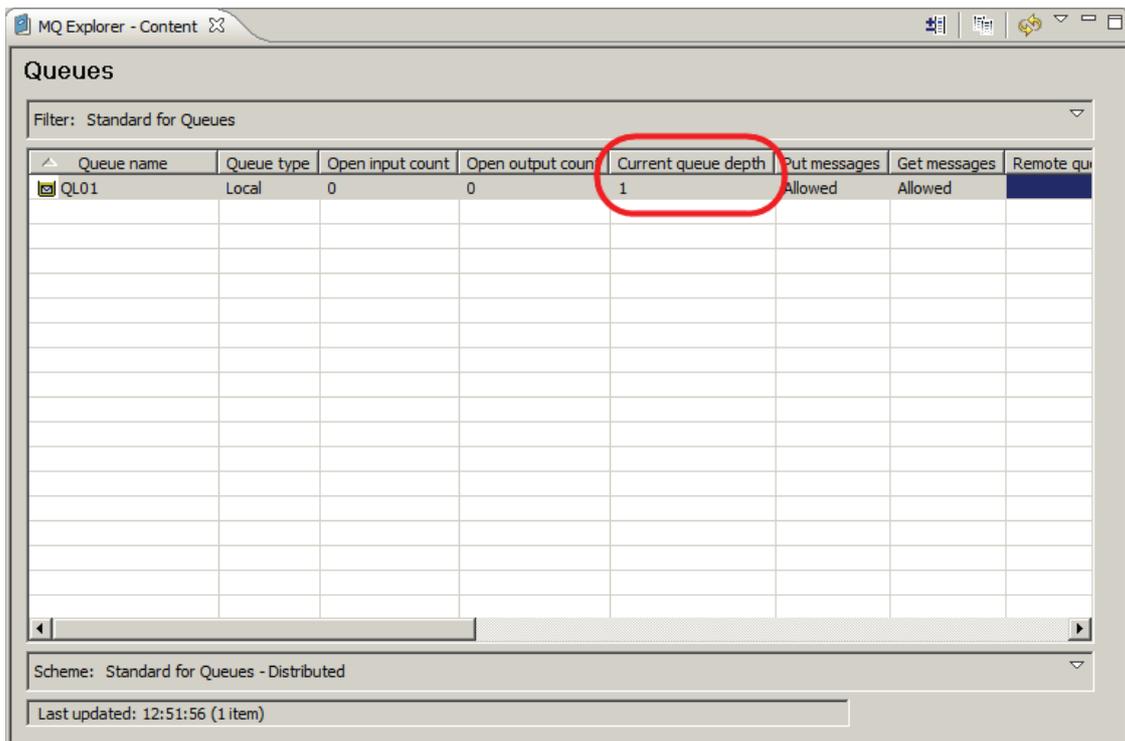
- __22. Enter a test message by typing something of your choosing in the **Message data** field and click on the **Put message** button.



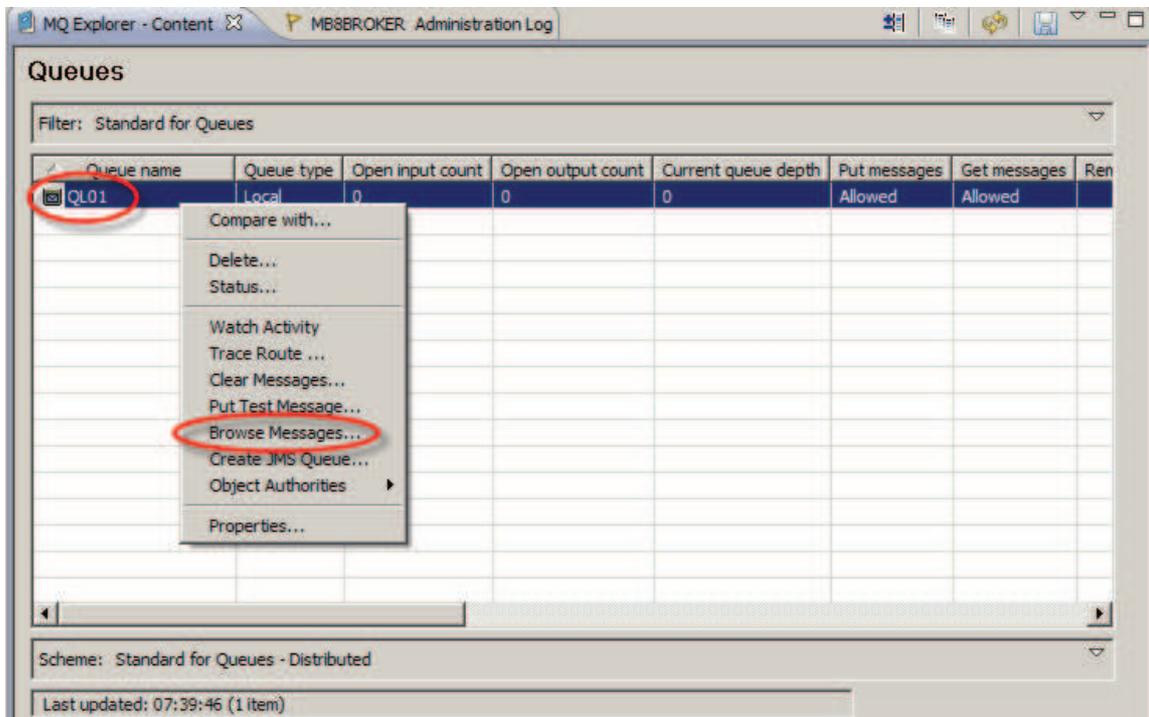
__23. Close the Put test message panel by clicking on the **Close** button.



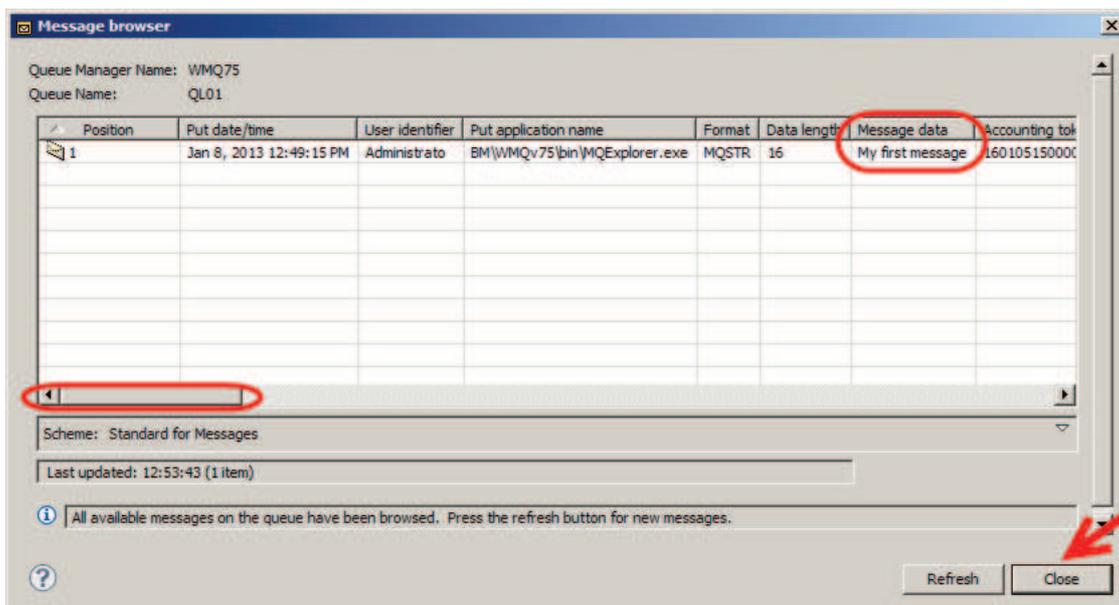
When you return to the display of queues, note that the **Current queue depth count is now 1**. This represents the test message you just placed into the queue.



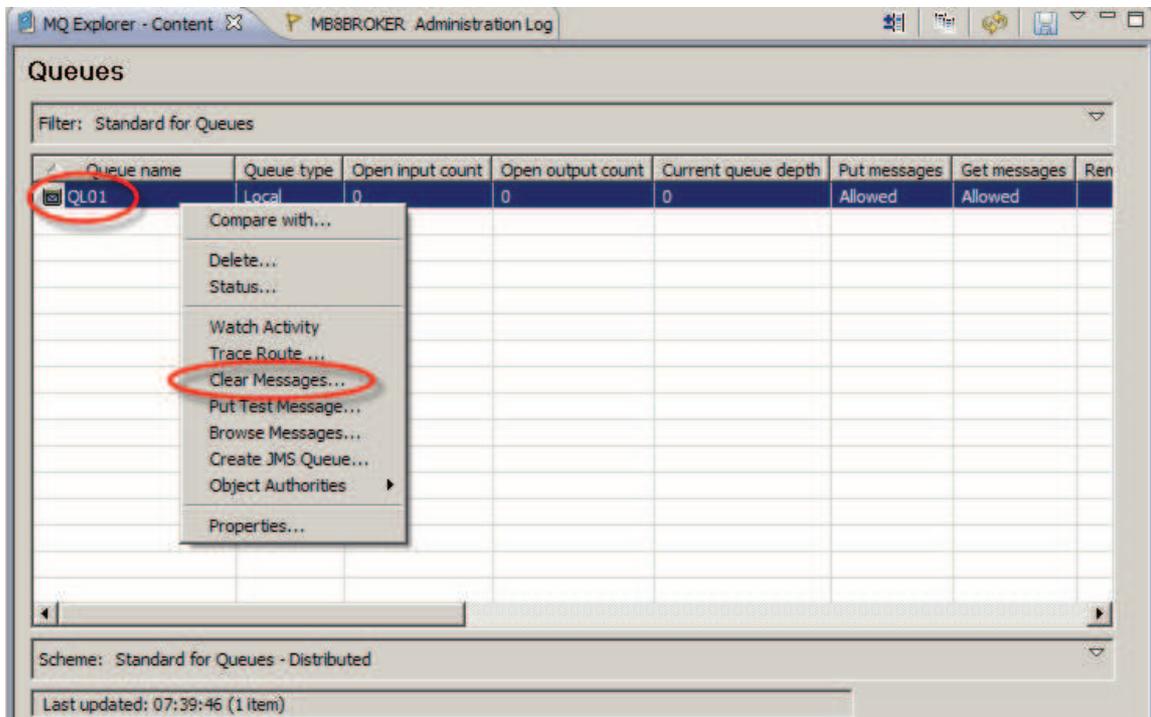
- __24. You can view details about the messages on a queue – **right-click** on queue **QL01** and select **Browse Messages**.



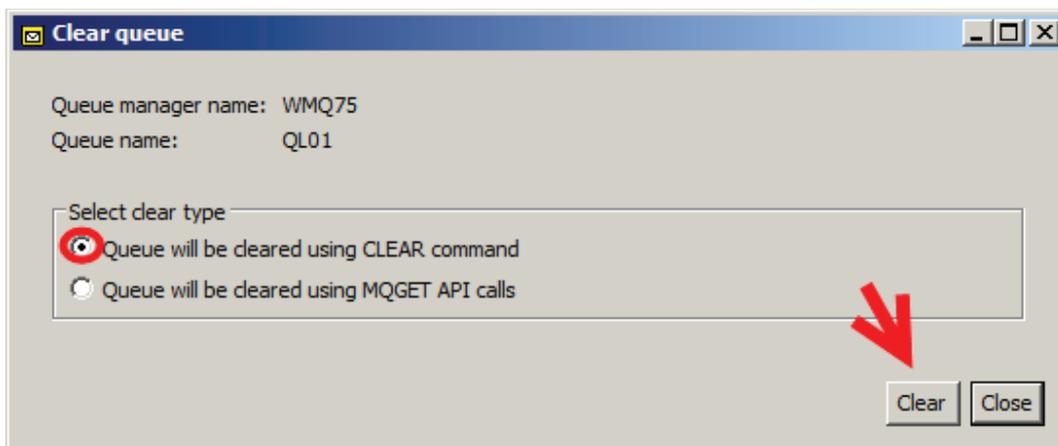
- __25. Here is the resulting display. Note that there is a **scroll bar** at the bottom that allows you to view additional data that is not currently in view. There will be one such line displayed for each message in the queue. Click on the **Close** button.



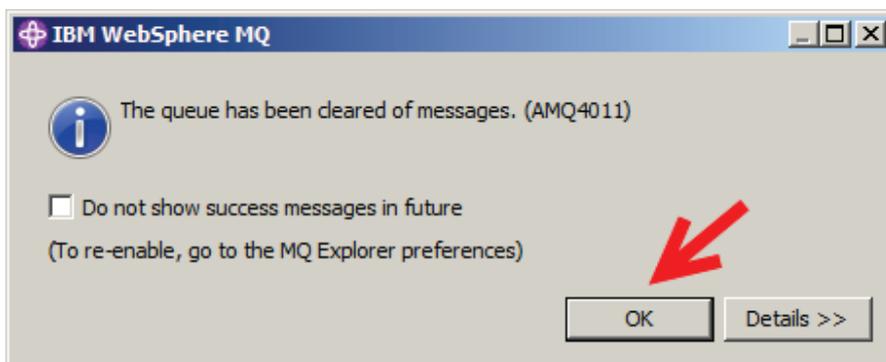
- __26. You will now delete your test message from the queue. **Right-click** on queue **QL01** and select **Clear Messages**.....note that this action will remove all messages from the queue.



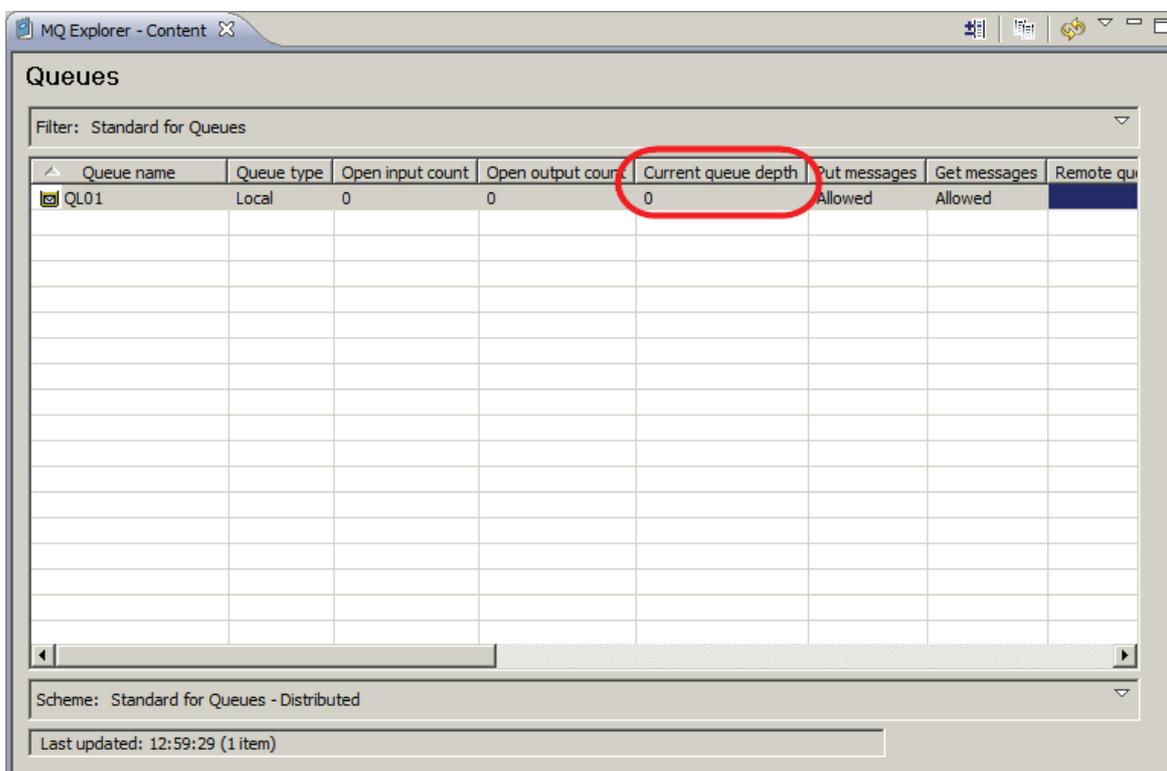
- __27. Accept the default selection of using the **CLEAR** command. This option cannot be used if any application has the queue open. In that case the second option would be required. Click on the **Clear** button.



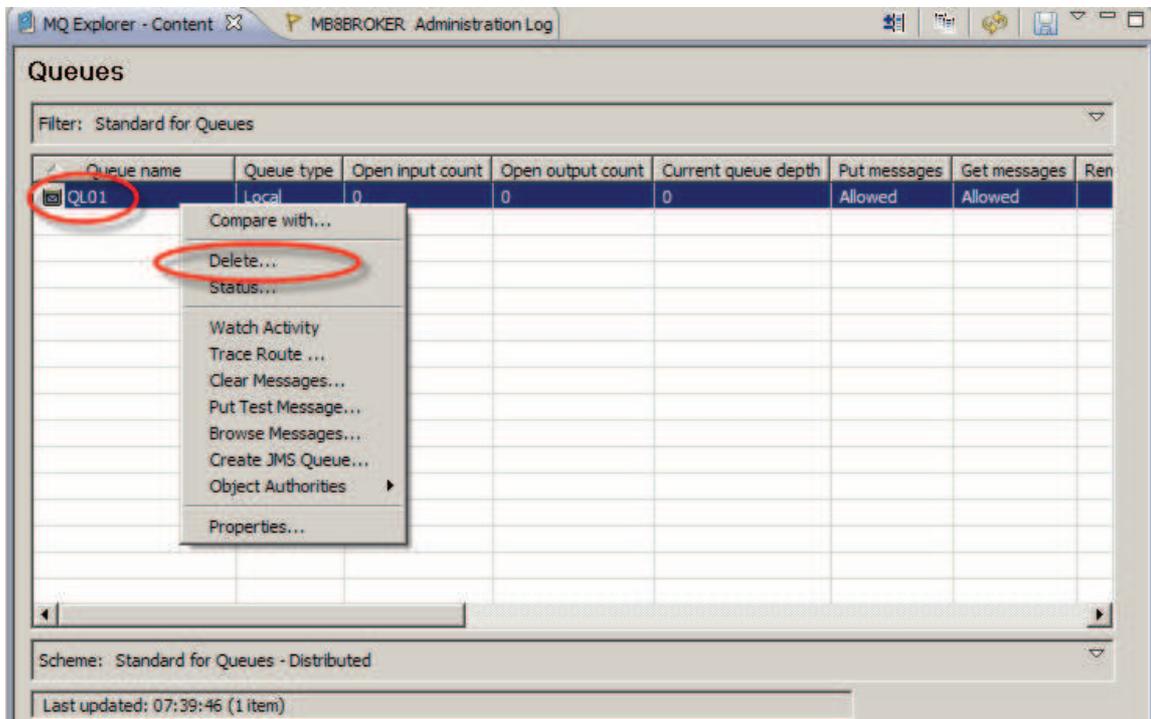
__28. Click on the **OK** button to dismiss the confirmation panel.



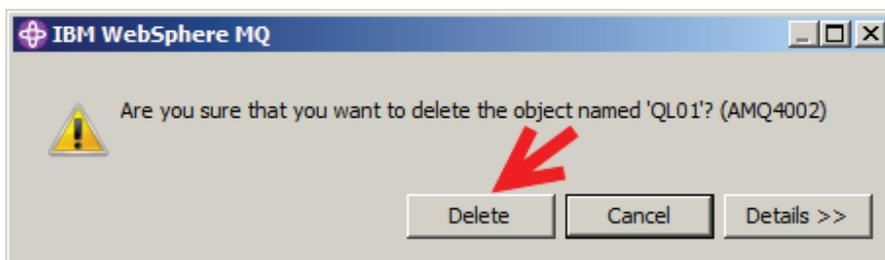
You can now see that the queue is indeed empty.



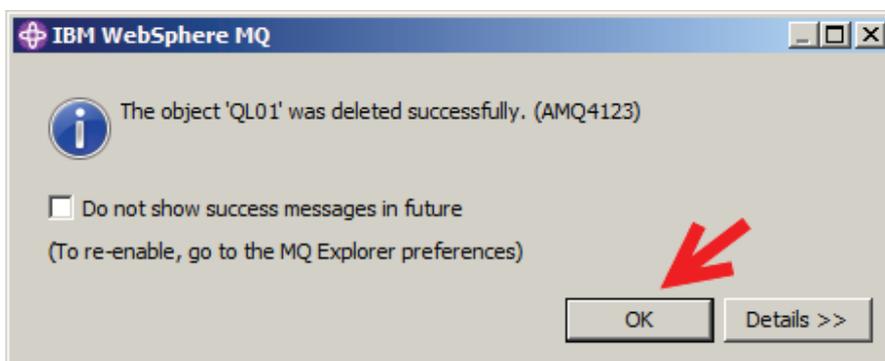
__29. You will now delete the queue. **Right-click** on the **QL01** queue and select **Delete...**



__30. Confirm the request by clicking on the **Delete** button.



__31. Dismiss the confirmation panel by clicking on the **OK** button.



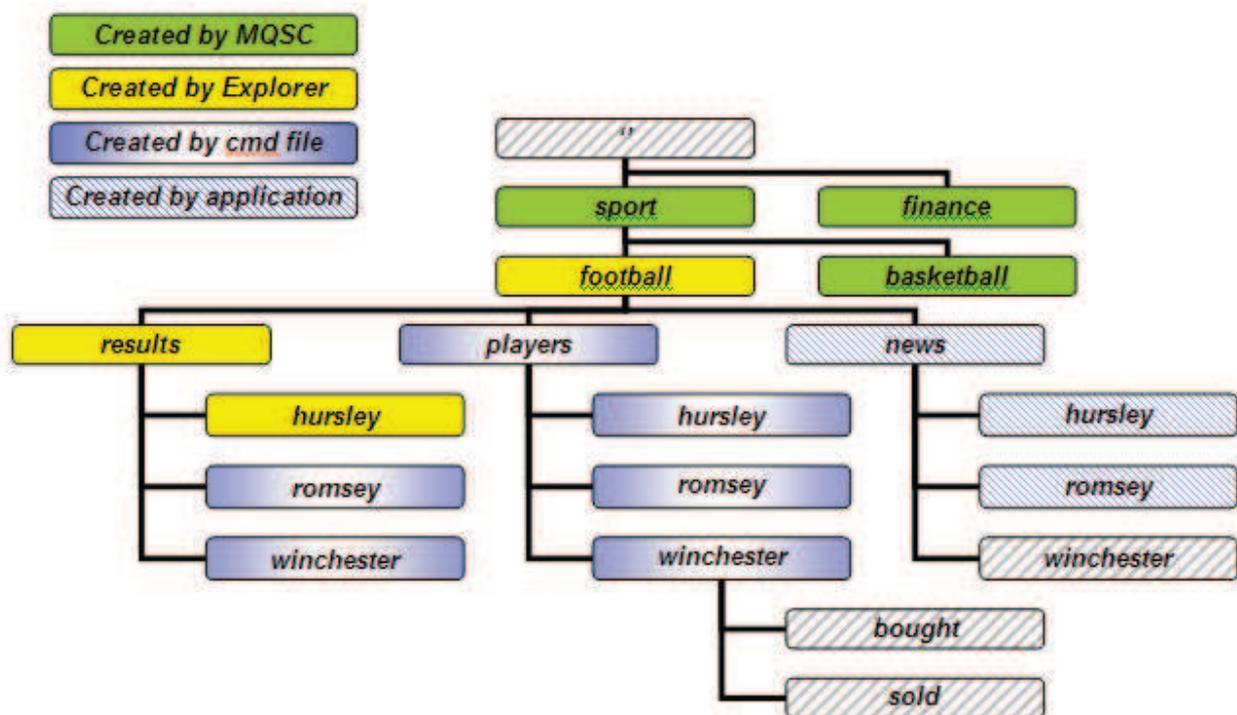
This concludes this portion of Lab 1.

Lab 3 Publish / Subscribe Administration

3.1 Lab Overview

This lab will demonstrate the administration of Publish / Subscribe-related objects using the WebSphere MQ Explorer. Also, you will use the MQ Explorer Test Publication and Test Subscription features to demonstrate publishing and subscribing to TOPIC strings.

You will be working with the hierarchical structure of TOPIC STRINGS represented in the picture below.



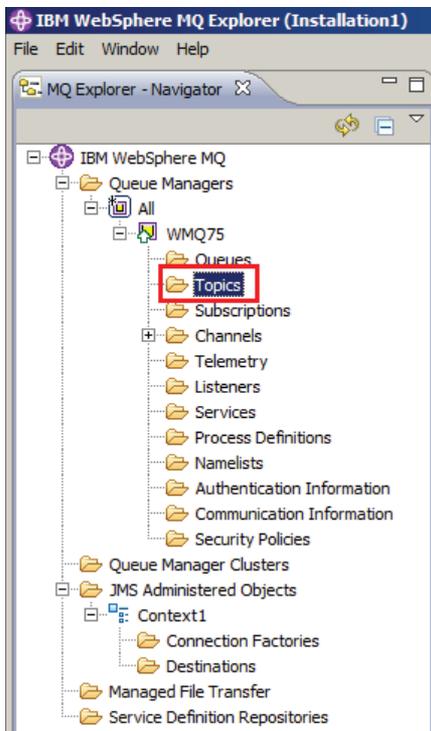
3.2 Using WebSphere MQ Explorer to create and display information

- __1. If the WebSphere MQ Explorer is not already running, you can launch it from the shortcut on the desktop.



- __2. Start the WebSphere MQ Explorer by double-clicking on the **WebSphere MQ Explorer** shortcut.

- ___3. Click on the **Topics** folder in the Navigator pane (left side). Notice that you have no topic objects defined yet! We have provided a script for this purpose.



- ___4. To run the provided script, find the shortcut called **PubSub Lab Setup** on the Windows® desktop. Double-click the shortcut to run the script.



If you receive an Open File – Security Warning popup, click on **Run** to allow the script to execute.



- __5. The script should run very quickly, leaving the following command window open. Verify that the command was successful. Then press Enter to close the command window.

```
AMQ8690: WebSphere MQ topic created.
6 : DEFINE TOPIC<SPORT.FOOTBALL.PLAYERS.WINCHESTER> +
  : TOPICSTR('sport/football/players/winchester') +
  : REPLACE +
  : PUB<ASPCRENT> +
  : SUB<ASPCRENT>
AMQ8690: WebSphere MQ topic created.
7 : DEFINE TOPIC<SPORT.FOOTBALL.NEWS> +
  : TOPICSTR('sport/football/news') +
  : REPLACE +
  : PUB<ENABLED> +
  : SUB<ENABLED>
AMQ8690: WebSphere MQ topic created.
7 MQSC commands read.
No commands have a syntax error.
All valid MQSC commands were processed.

C:\Student\Lab_PubSubAdmin\resources>pause
Press any key to continue . . .
```

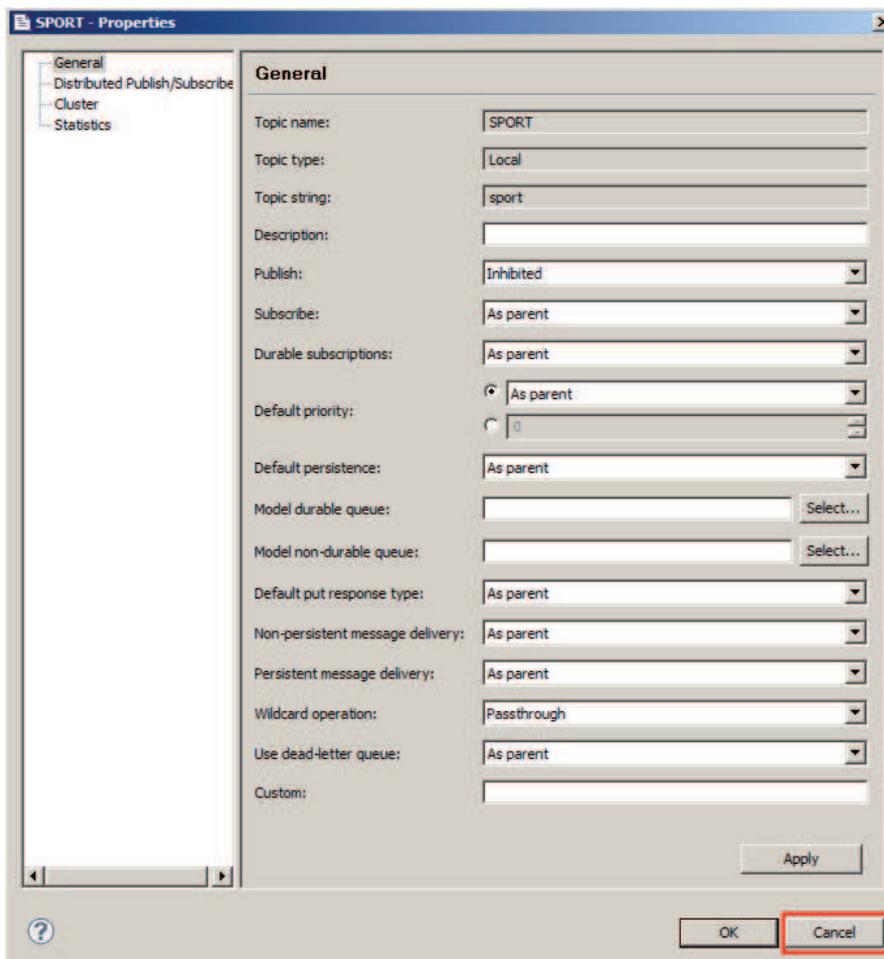
- __6. You should now see the following WebSphere MQ Topic objects displayed in the Content pane in the MQ Explorer. Observe the Topic objects (under the Topic name column) and their corresponding Topic string values. Also observe the Publish and Subscribe enablement status on the right.

Topic name	Topic type	Topic string	Description	Publish	Subscribe
FINANCE	Local	finance		Inhibited	Inhibited
SPORT	Local	sport		Inhibited	As parent
SPORT.BASKETBALL	Local	sport/basketball		As parent	As parent
SPORT.FOOTBALL	Local	sport/football		Inhibited	As parent
SPORT.FOOTBALL.NEWS	Local	sport/football/news		Allowed	Allowed
SPORT.FOOTBALL.PLAYERS	Local	sport/football/players		Allowed	Allowed
SPORT.FOOTBALL.PLAYERS.HURSLEY	Local	sport/football/players/hursley		As parent	As parent
SPORT.FOOTBALL.PLAYERS.ROMSEY	Local	sport/football/players/romsey		As parent	As parent
SPORT.FOOTBALL.PLAYERS.WINCHESTER	Local	sport/football/players/winchester		As parent	As parent
SPORT.FOOTBALL.RESULTS	Local	sport/football/results		Inhibited	As parent
SPORT.FOOTBALL.RESULTS.HURSLEY	Local	sport/football/results/hursley		Allowed	Allowed
SPORT.FOOTBALL.RESULTS.ROMSEY	Local	sport/football/results/romsey		Allowed	Allowed
SPORT.FOOTBALL.RESULTS.WINCHESTER	Local	sport/football/results/winchester		Allowed	Allowed

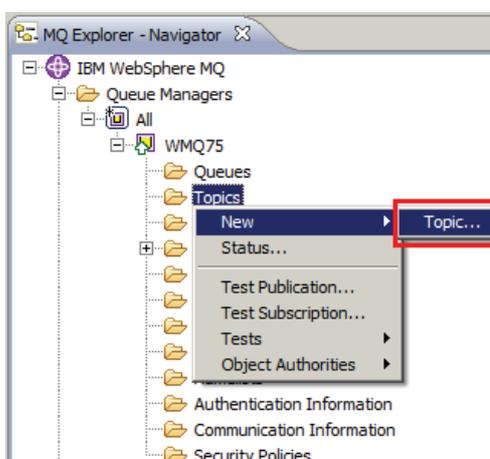
- __7. Double-click on the **SPORT** Topic object.

Topic name	Topic type	Topic string	Description
FINANCE	Local	finance	
SPORT	Local	sport	
SPORT.BASKETBALL	Local	sport/basketball	
SPORT.FOOTBALL	Local	sport/football	
SPORT.FOOTBALL.NEWS	Local	sport/football/news	
SPORT.FOOTBALL.PLAYERS	Local	sport/football/players	
SPORT.FOOTBALL.PLAYERS.HURSLEY	Local	sport/football/players/hursley	
SPORT.FOOTBALL.PLAYERS.ROMSEY	Local	sport/football/players/romsey	
SPORT.FOOTBALL.PLAYERS.WINCHESTER	Local	sport/football/players/winchester	
SPORT.FOOTBALL.RESULTS	Local	sport/football/results	
SPORT.FOOTBALL.RESULTS.HURSLEY	Local	sport/football/results/hursley	
SPORT.FOOTBALL.RESULTS.ROMSEY	Local	sport/football/results/romsey	
SPORT.FOOTBALL.RESULTS.WINCHESTER	Local	sport/football/results/winchester	

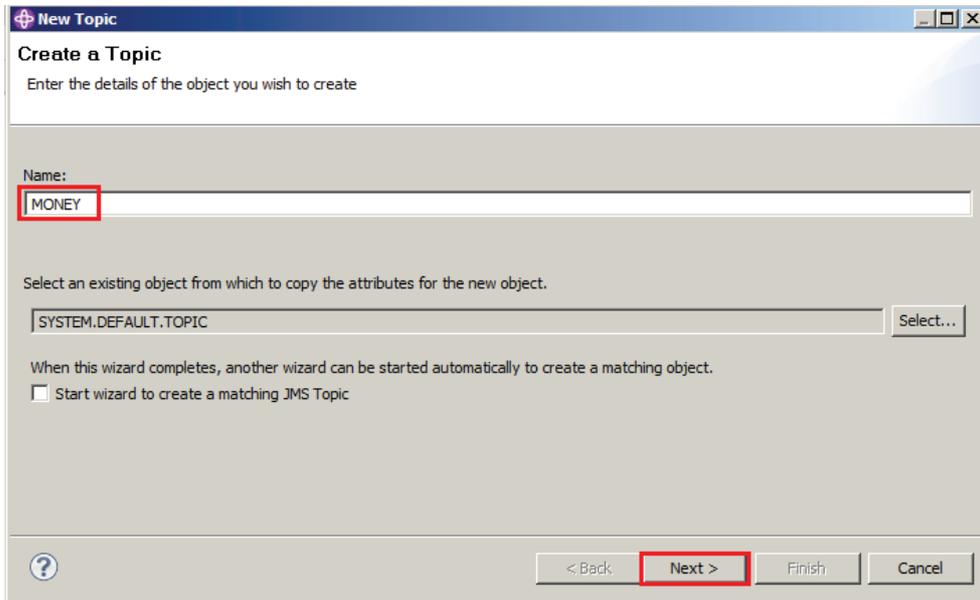
- __8. Observe the various properties of the Topic object. Explore the various values available on the pull-downs if you wish. Close the window by clicking **Cancel**.



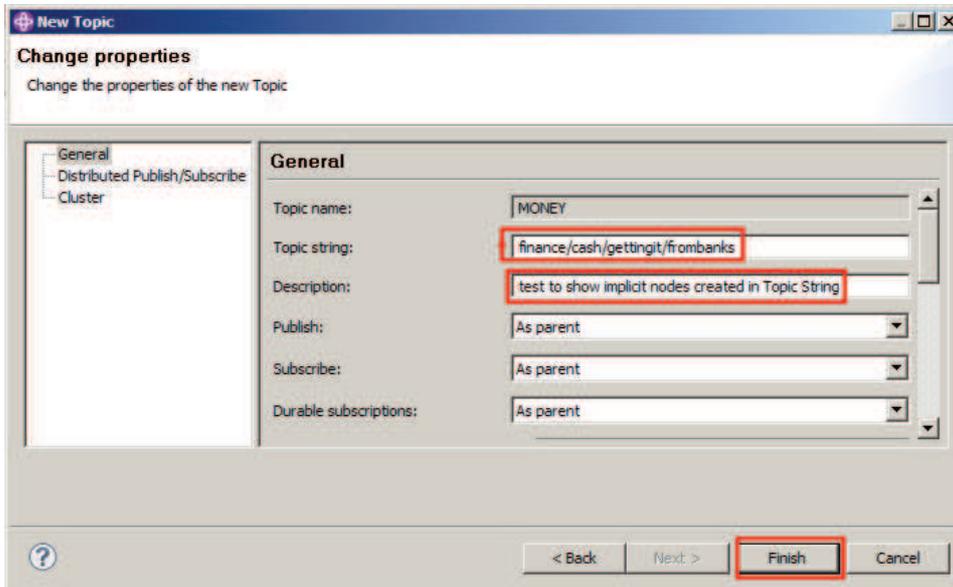
- __9. From the Navigation pane select **Topics** under queue manager WMQ75. Right-click on **Topics** and then select **New** → **Topic** from the context menu.



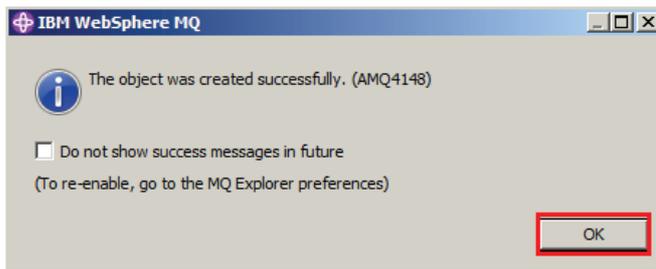
__10. Enter **MONEY** in the Name field; allow the other fields to default. Then click “**Next**”.



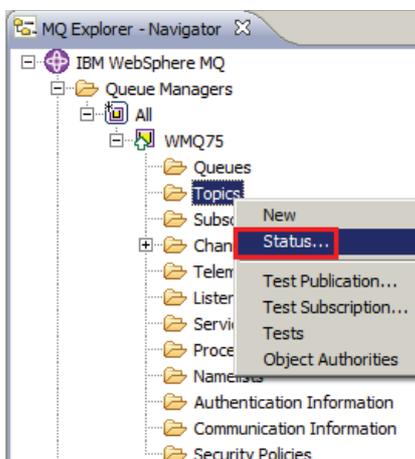
__11. Enter “**finance/cash/gettingit/frombanks**” (without the quote marks) in the **Topic string** field. Enter a description in the description field and then click “**Finish**”



- __12. Close the confirmation box by clicking **OK**.



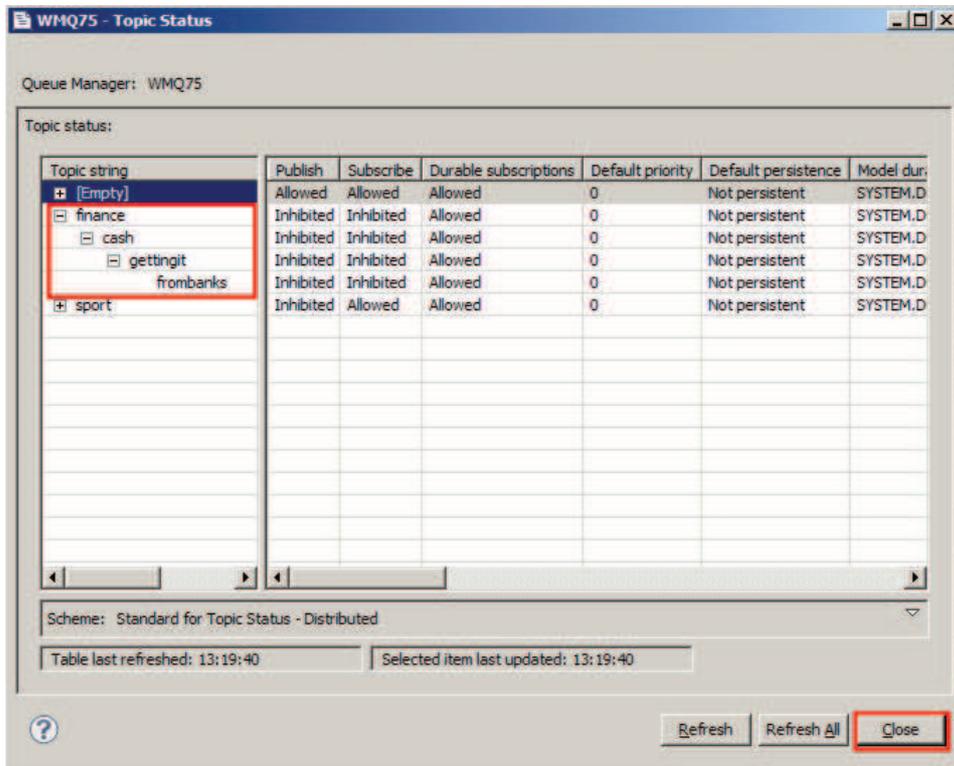
- __13. Back in the WebSphere MQ Explorer, right click on **Topics** (under WMQ75) and then select **Status**.



- __14. Now click on the “+” symbol to the left of **finance** – a level of the hierarchy opens; repeat on the “+” in front of **cash**, then **gettingit**, then **frombanks** which is the bottom of the “tree”.

In the Topic Status notice that all the intermediate nodes have been created and that they have inherited properties from the parent **finance**. These intermediate nodes have no related Topic Objects and so cannot have their properties altered by MQSC or WebSphere MQ Explorer.

__15. Close the Topic Status view by clicking on **Close**.

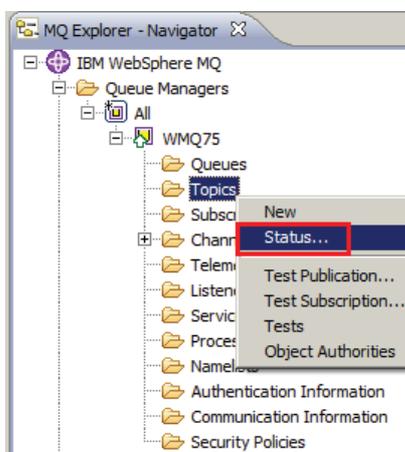


This concludes this portion of Lab 3.

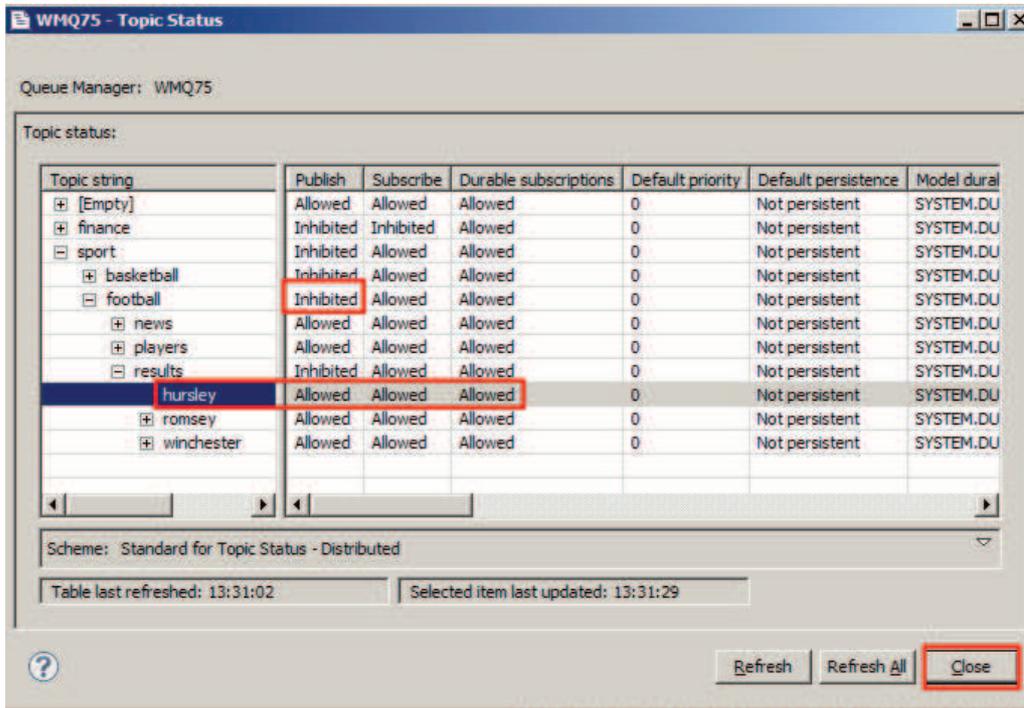
3.3 A First look at the MQ Explorer Pub/Sub Test tools

In this section you will be using the tools that come with WebSphere MQ Explorer that allow you to test publishing to and subscribing to topics.

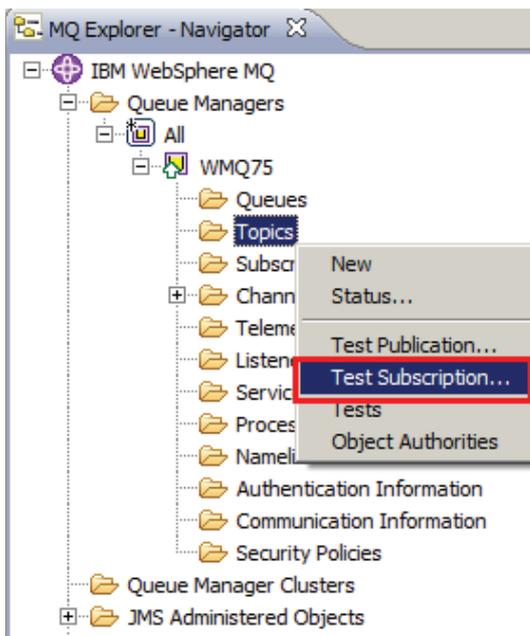
- ___16. You will now be working with some pre-defined topics. From the WebSphere MQ Explorer, display topic status by right-clicking on the **Topics** folder and selecting **Status...**



- __17. You are going to focus on the **sport** topic tree. Expand the “+” symbols on the **sport** topic tree and you should see something similar to the screen capture below. Notice that Publish is allowed for the topic string **sport/football/results/hursley**. Also, following the tree “up”, you should notice that the topic string **sport/football** has the publish attribute **Inhibited**. Remember this as you complete the next steps of this lab. Click **Close** to dismiss the status window.

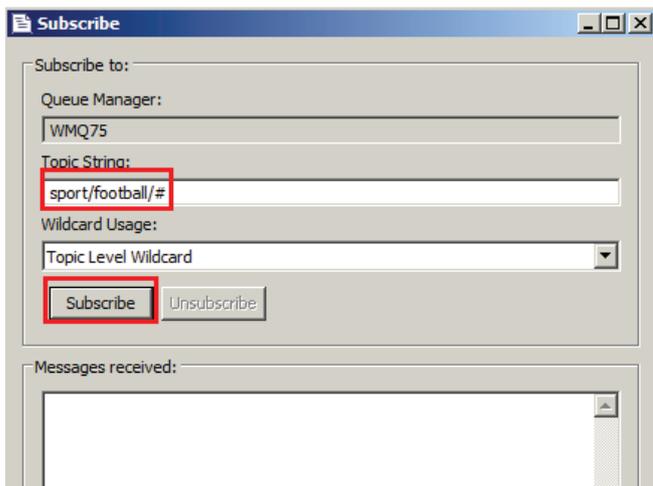


- __18. Start a test subscription window by right-clicking on **Topics** and selecting **Test Subscription**.



- __19. Type in the topic string **sport/football/#** and press the **Subscribe** button.

The “#” symbol is called the *multi-level wildcard*. The string **sport/football/#** indicates a subscription to all publications sent to the sport/football topic or any of its children. The Test Tool window remains open and the **Unsubscribe** button becomes active. Publications received will be displayed in the **Messages received** box.



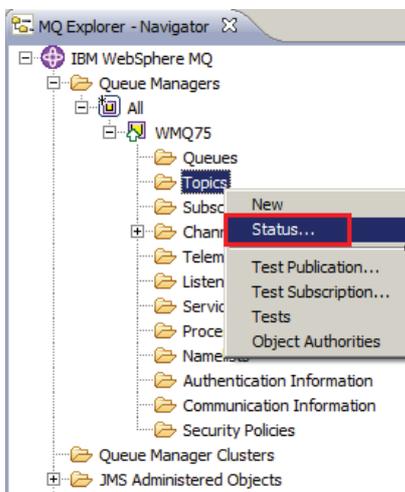
- __20. **Minimize** the Subscribe window by clicking on the minimize button.



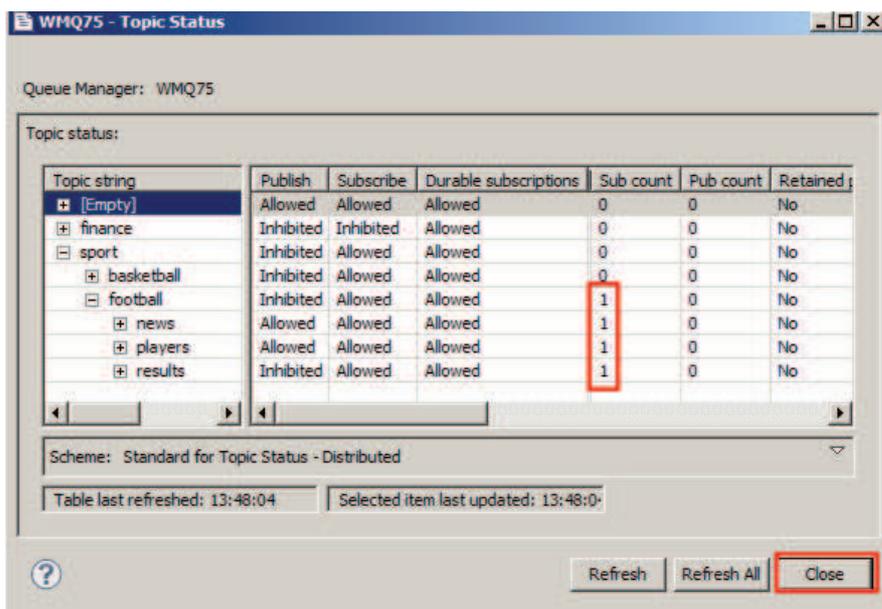
- __21. The Subscribe window will “park” itself at the bottom left part of the Windows desktop. You will restore this window in a later step of this lab.



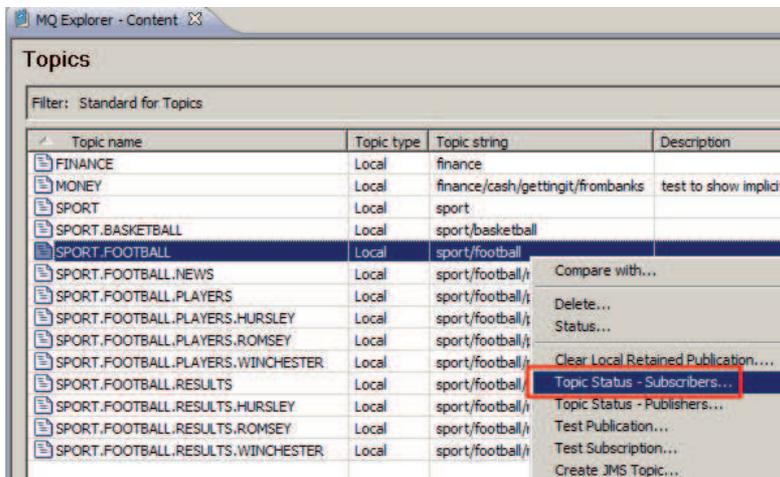
__22. Right click on the Topics folder then select **Status...**



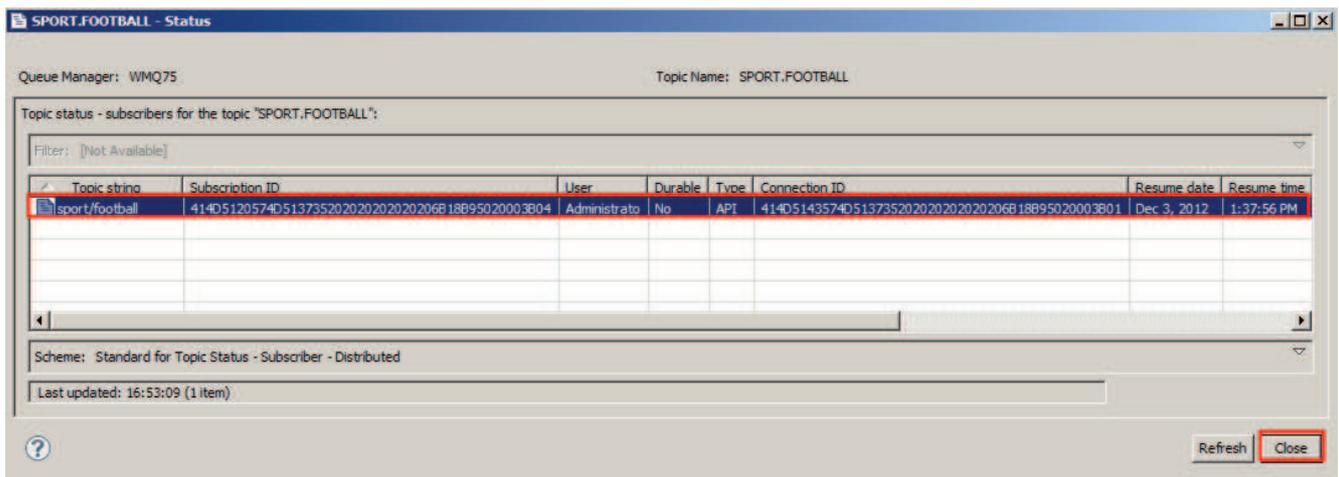
__23. Expand the **sport** tree. Observe the Subscription counts; sport/football and its children have a positive subscription count. You will need to scroll to the right to find the **Sub count** column. Close the topic status window.



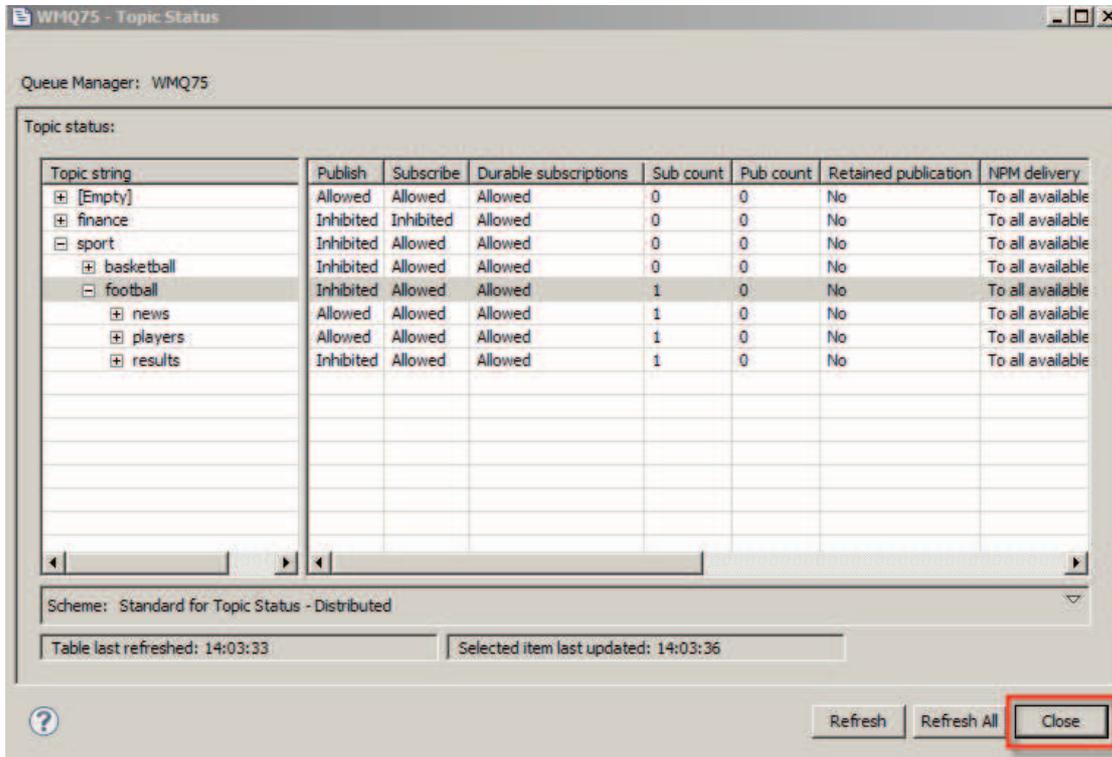
- __24. From the Topic display, select the **SPORT.FOOTBALL** row, right-click and select **Topic Status – Subscribers**. This gives detailed information about subscribers to this Topic object.



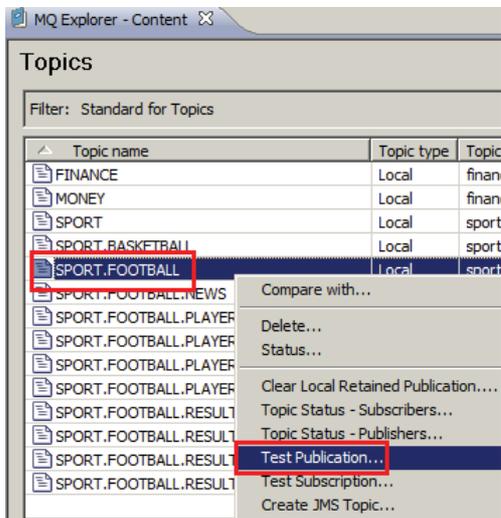
- __25. Observe the detailed display and then close the status screen by clicking on the **Close** button.



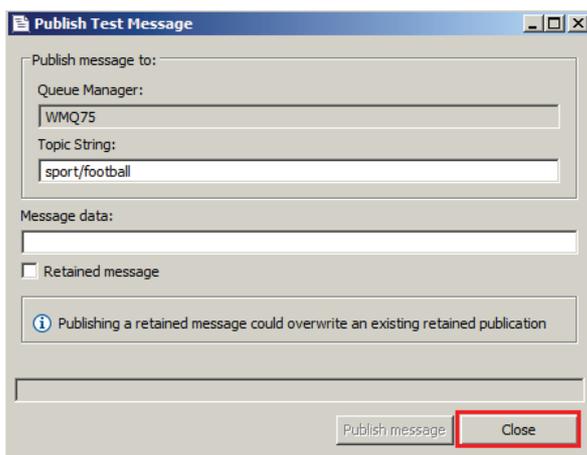
__26. Close the **Topic Status** window by clicking **Close**.



- __27. Now you will publish a message. Returning to the Topic list, select the **SPORT.FOOTBALL** row, right-click and then select **Test Publication**.



- __28. This dialog will publish a message to the topic string **sport/football**.



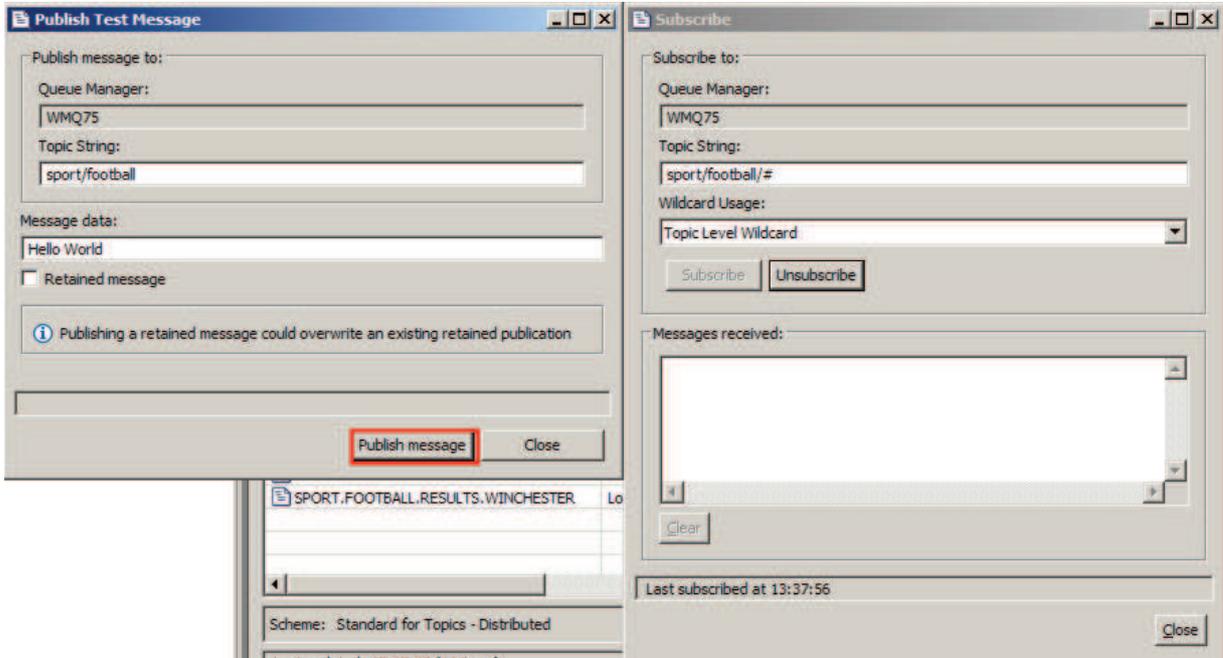
- __29. Before entering a message and sending it, you will arrange the windows on the screen.

- __30. Locate the **Subscribe** Test Tool window where you previously subscribed to **sport/football/#**. Click on the Restore window button to restore the window. Now position the **Publish Test Message** and the restored **Subscribe** test tool windows so they both are visible. Then return focus to the **Publish** window.

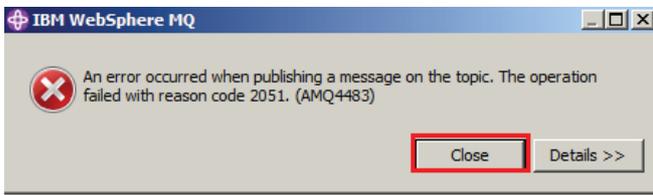


- __31. Type a message such as **Hello World** and then press **Publish Message**.

- __32. An error occurs because the topic object for **sport/football** is **publish-inhibited**; you'll recall that we saw that this was set earlier in the lab. But this will not inhibit our subscribers; we used the multi-level wildcard to subscribe to topics at and below "sport/football" in the topic tree, so we will be subscribing to items published lower in the hierarchy.



- __33. Click on Close to dismiss the error popup.

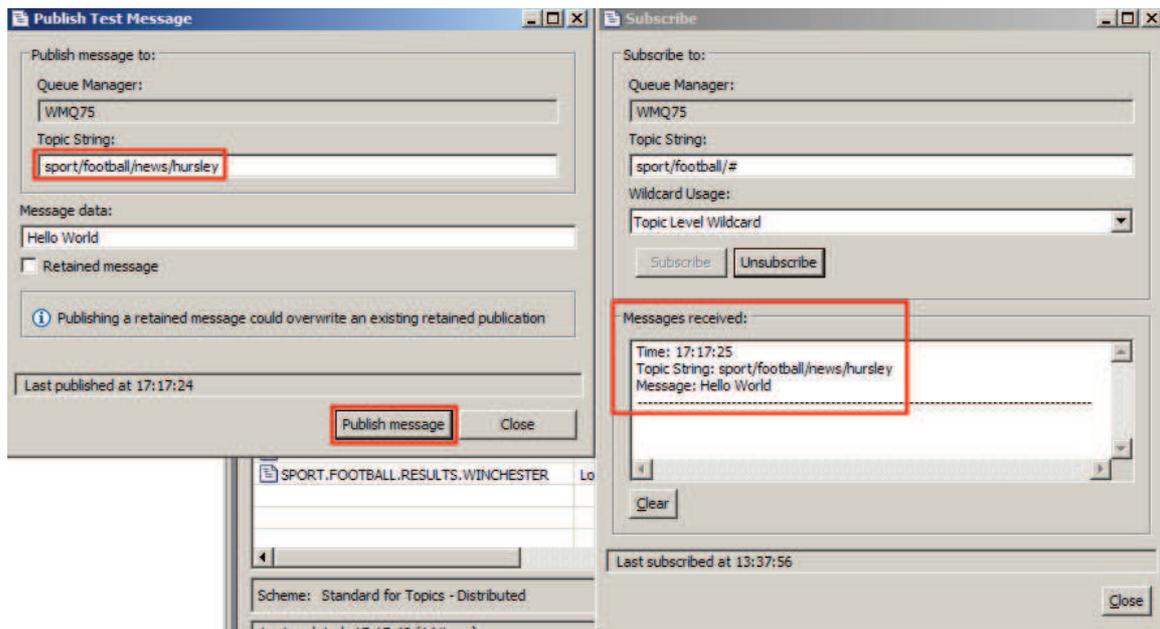


- __34. In the Publish Test Message window, overwrite the topic string to **sport/football/news/hursley** and click on **Publish message**. You have published and subscribed your first message!

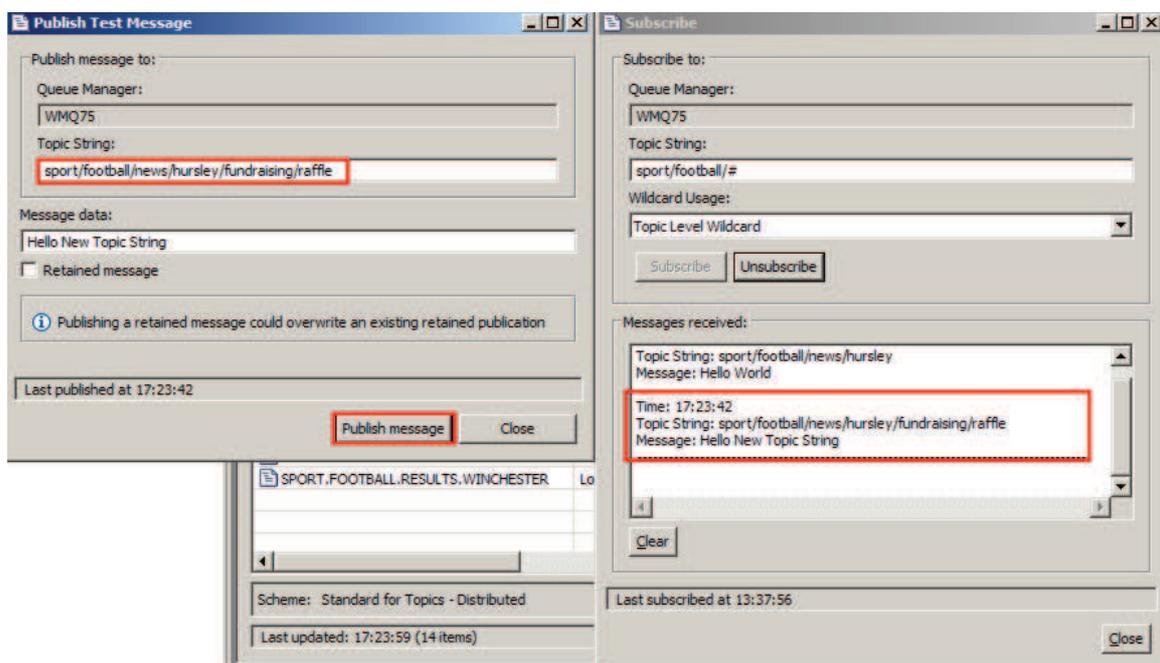


Dynamically Created Topic Objects

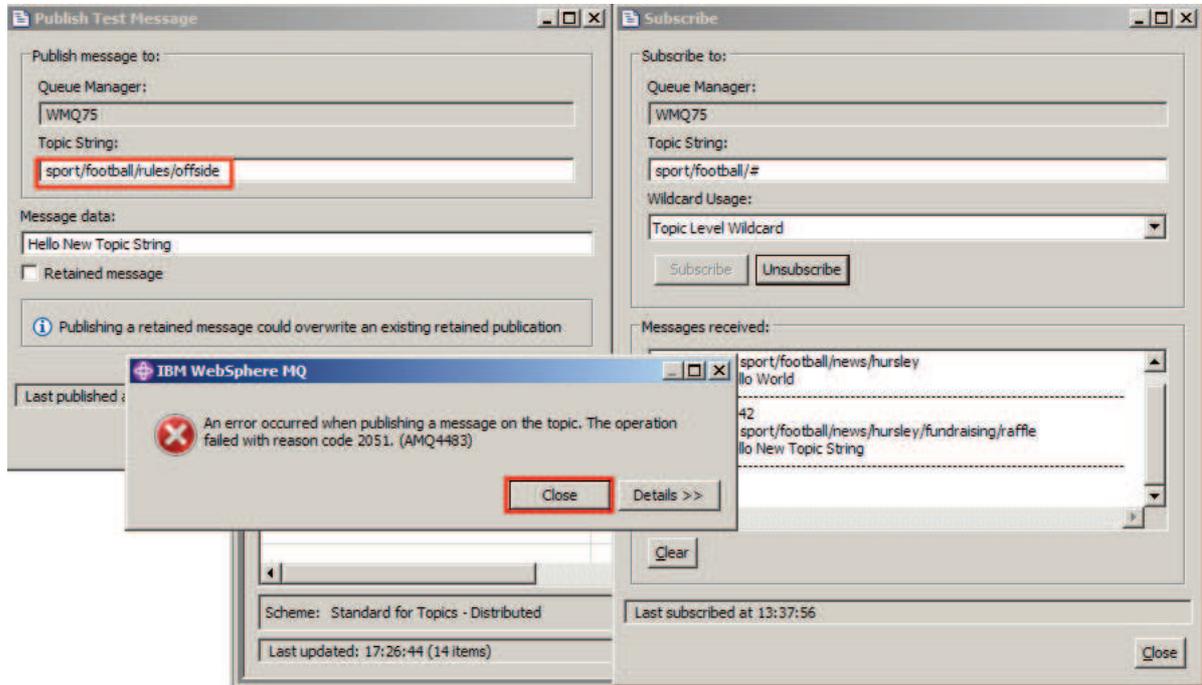
These dynamically created Topic objects are temporary and only exist for a limited amount of time before the Queue Manager removes them; for example if you restart the Queue Manager, they will no longer exist..



35. Now try publishing to **sport/football/news/hursley/fundraising/raffle**. The message is sent to the subscriber. New levels of the hierarchy have been created automatically.



- ___36. Now try publishing to **sport/football/rules/offside**. The publish attempt failed! That is because the node in the topic tree that is dynamically created automatically inherits the properties of the parent **sport/football** – which has its Publish attribute Inhibited. Click on **Close** to close the error popup.



- __37. Return to the Topic object display. Once again open the Topic status list and expand the **sport** hierarchy. You will see the automatically created elements. Click on **Close** to close the status window.

WMQ75 - Topic Status

Queue Manager: WMQ75

Topic status:

Topic string	Publish	Subscribe	Durable subscriptions	Default persistence	Model durable queue	Model non-durable c
[Empty]	Allowed	Allowed	Allowed	Not persistent	SYSTEM.DURABLE....	SYSTEM.NDURABLE.
finance	Inhibited	Inhibited	Allowed	Not persistent	SYSTEM.DURABLE....	SYSTEM.NDURABLE.
sport	Inhibited	Allowed	Allowed	Not persistent	SYSTEM.DURABLE....	SYSTEM.NDURABLE.
basketball	Inhibited	Allowed	Allowed	Not persistent	SYSTEM.DURABLE....	SYSTEM.NDURABLE.
football	Inhibited	Allowed	Allowed	Not persistent	SYSTEM.DURABLE....	SYSTEM.NDURABLE.
news	Allowed	Allowed	Allowed	Not persistent	SYSTEM.DURABLE....	SYSTEM.NDURABLE.
hursley	Allowed	Allowed	Allowed	Not persistent	SYSTEM.DURABLE....	SYSTEM.NDURABLE.
fundraising	Allowed	Allowed	Allowed	Not persistent	SYSTEM.DURABLE....	SYSTEM.NDURABLE.
raffie	Allowed	Allowed	Allowed	Not persistent	SYSTEM.DURABLE....	SYSTEM.NDURABLE.
players	Allowed	Allowed	Allowed	Not persistent	SYSTEM.DURABLE....	SYSTEM.NDURABLE.
hursley	Allowed	Allowed	Allowed	Not persistent	SYSTEM.DURABLE....	SYSTEM.NDURABLE.
romsey	Allowed	Allowed	Allowed	Not persistent	SYSTEM.DURABLE....	SYSTEM.NDURABLE.
winchester	Allowed	Allowed	Allowed	Not persistent	SYSTEM.DURABLE....	SYSTEM.NDURABLE.
results	Inhibited	Allowed	Allowed	Not persistent	SYSTEM.DURABLE....	SYSTEM.NDURABLE.
hursley	Allowed	Allowed	Allowed	Not persistent	SYSTEM.DURABLE....	SYSTEM.NDURABLE.
romsey	Allowed	Allowed	Allowed	Not persistent	SYSTEM.DURABLE....	SYSTEM.NDURABLE.
winchester	Allowed	Allowed	Allowed	Not persistent	SYSTEM.DURABLE....	SYSTEM.NDURABLE.
rules	Inhibited	Allowed	Allowed	Not persistent	SYSTEM.DURABLE....	SYSTEM.NDURABLE.
offside	Inhibited	Allowed	Allowed	Not persistent	SYSTEM.DURABLE....	SYSTEM.NDURABLE.
sportootball	Allowed	Allowed	Allowed	Not persistent	SYSTEM.DURABLE....	SYSTEM.NDURABLE.

Scheme: Standard for Topic Status - Distributed

Table last refreshed: 17:29:58 Selected item last updated: 17:29:58

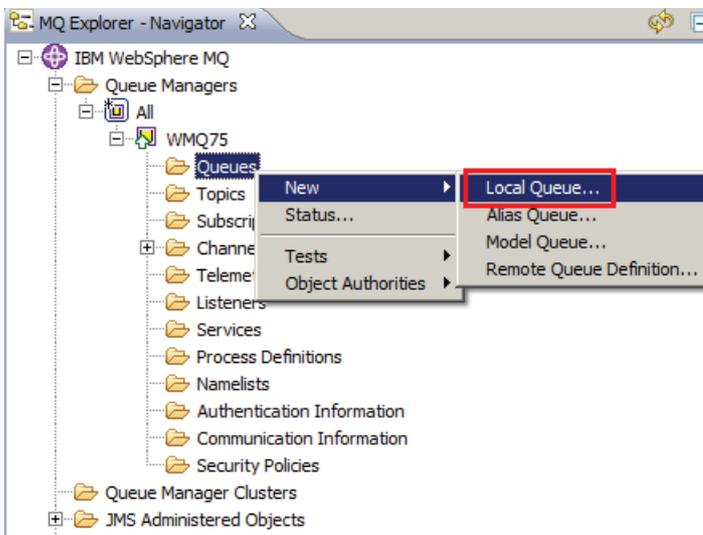
Refresh Refresh All **Close**

This concludes this portion of Lab 3.

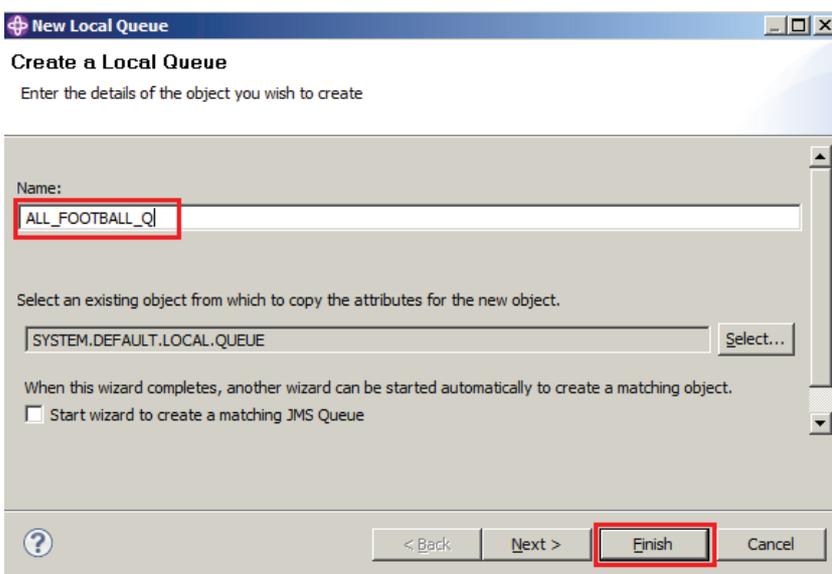
3.2 Administered Subscriptions

While it is typical for subscribers to register their own subscriptions, it is possible to administratively register a subscription using WebSphere MQ Explorer. This is a subscription to a topic string that delivers messages to a queue. This can be very useful because it is a way for a legacy program which was designed as a point-to-point application to read a queue associated with a topic; in this way it can participate in publish/subscribe without changing the program. You will now explore how such a subscription can be set up and used.

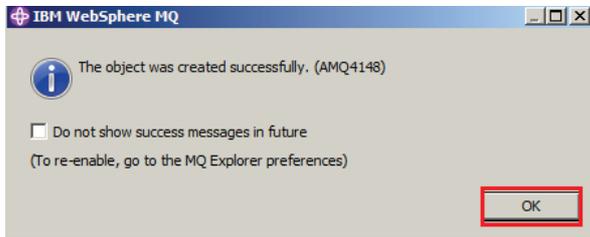
__38. In the WebSphere MQ Explorer, select **Queues**. Right-click and select **New → Local Queue**.



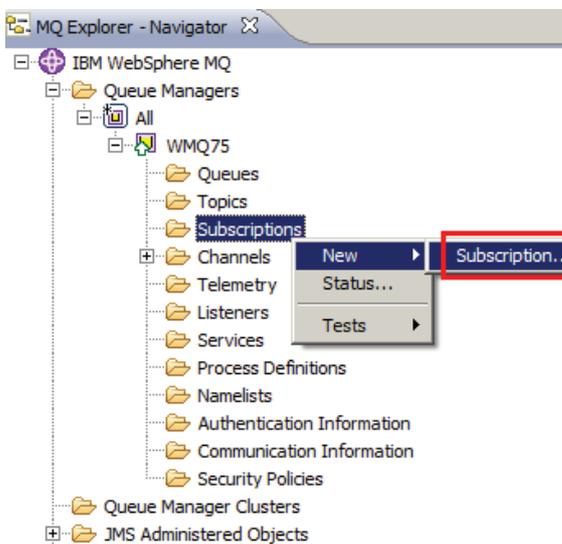
__39. Name the queue **ALL_FOOTBALL_Q** and press **Finish**.



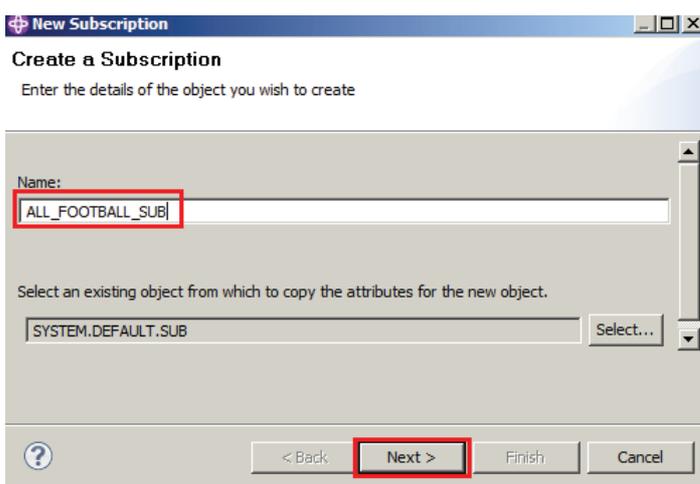
__40. Click **OK** to close the confirmation dialog.



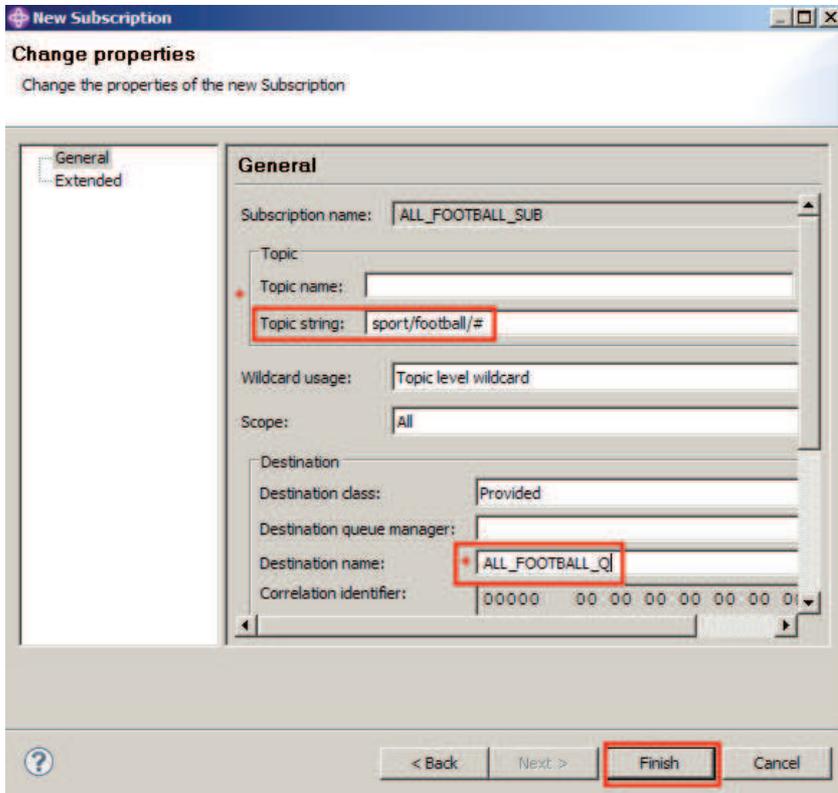
__41. Select **Subscriptions**, right-click and select **New** → **Subscription**.



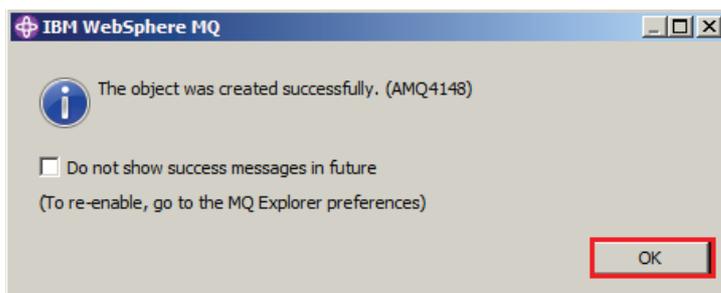
__42. Type **ALL_FOOTBALL_SUB** as the subscription name and then click **Next**.



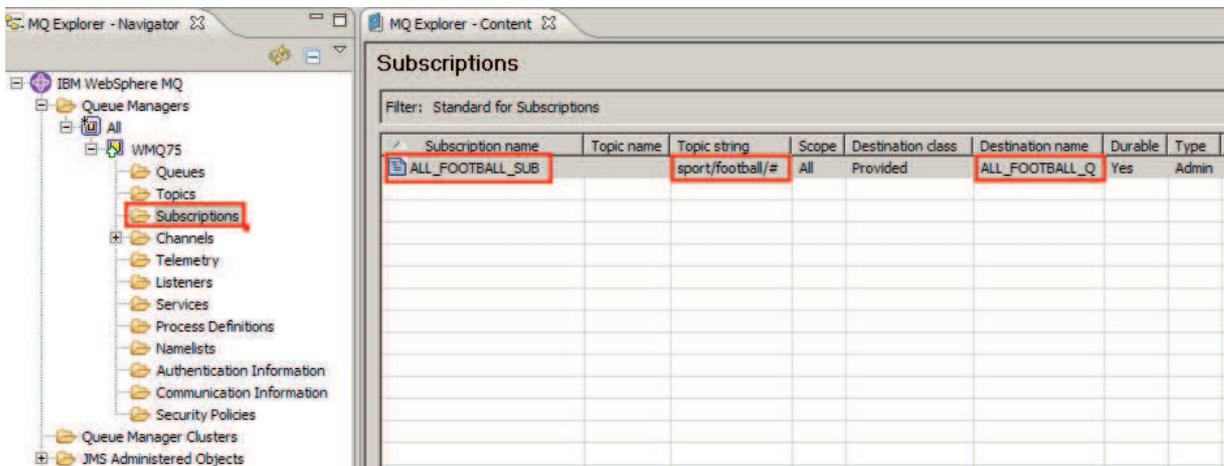
- __43. Leave the Topic Name blank, and enter **sport/football/#** as the Topic string. Leave the Destination Queue Manager blank and enter **ALL_FOOTBALL_Q** in the Destination Name. Then click **Finish**.



- __44. Click **OK** to close the confirmation window.

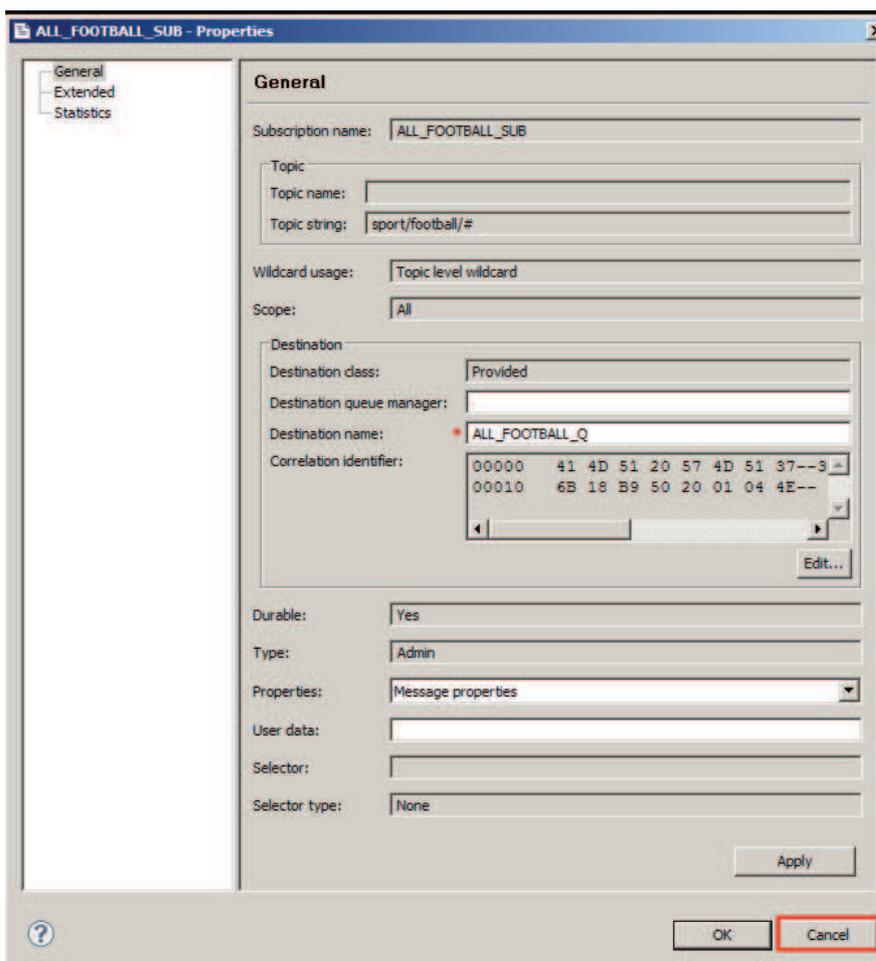


__45. The new administrative subscription appears.



Double-click on the new subscription to see its attributes.

__46. The attributes of the new subscription are displayed.



This subscription will now route all qualifying messages to the local queue **ALL_FOOTBALL_Q**.

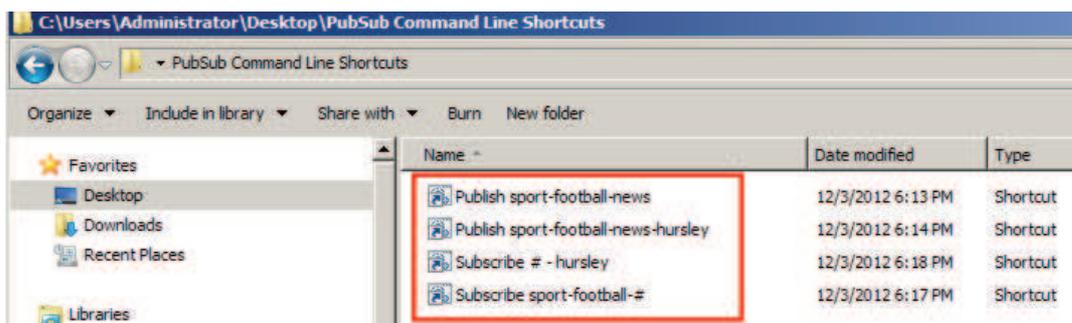
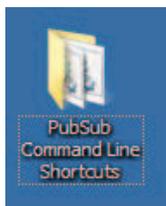
__47. Close the properties window by clicking on **Cancel**.

This concludes this portion of Lab 3.

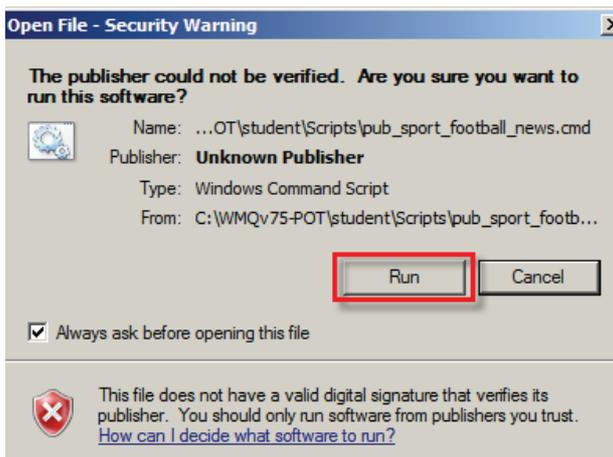
3.5 Testing Publications and Subscriptions from the command line

You will now use two more sample programs that are supplied with WebSphere MQ to further test WebSphere MQ publish and subscribe capabilities, called amqspub and amqssub.

- __48. A folder on the desktop contains four shortcuts that will start two instances of a publishing sample and two subscribers. Open the folder and then double-click each shortcut to launch the programs.



If you receive an **Open File – Security Warning**, click on **Run**.

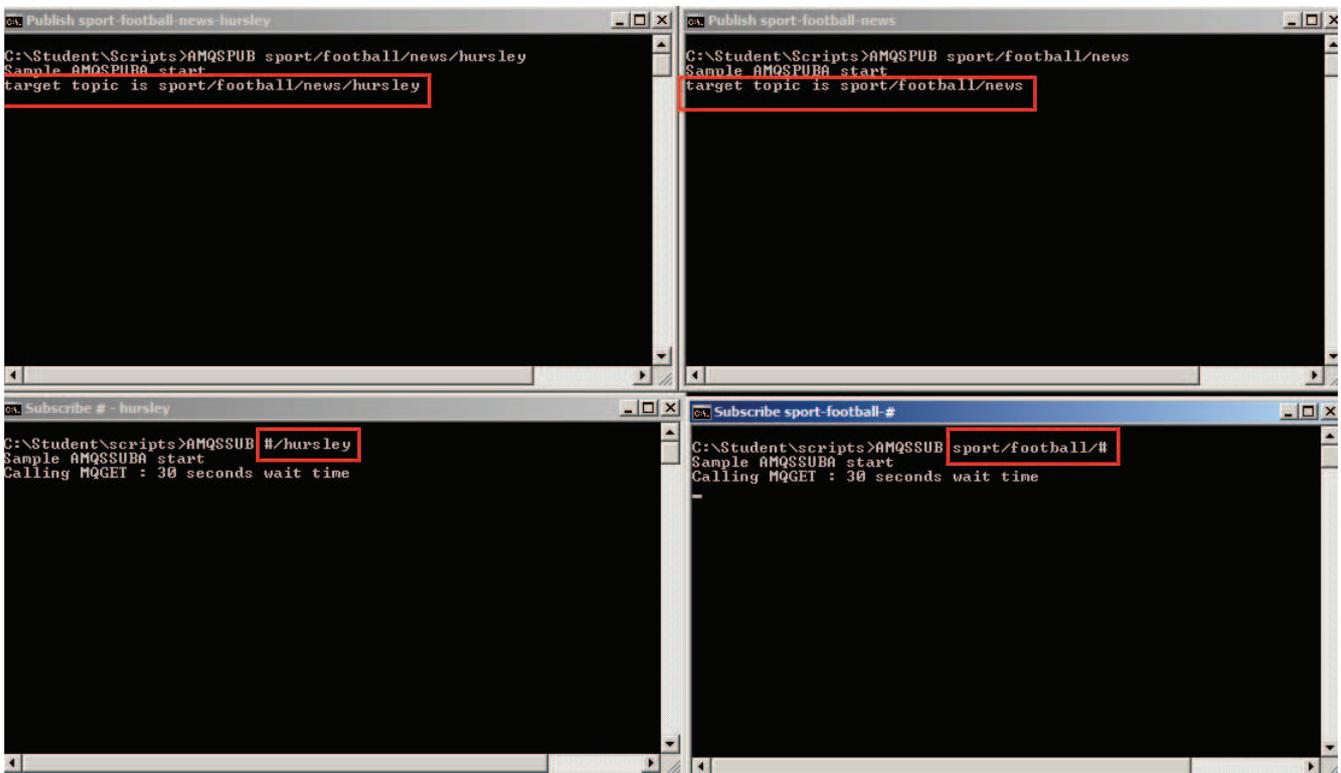


- __49. Arrange the windows as shown, so that you can see all of them at the same time. The top two windows are the *topic publishers* (amqspub). Each time you type text into either window, the windows on the bottom, the *topic subscribers* (amqssub) will receive the text as published messages because the topic string that they are subscribing to matched the one being used by the publishers.



Subscriber Timeout

The amqssub program times out after 30 seconds if no messages arrive. If your window times out, just double-click the command again to restart the program..

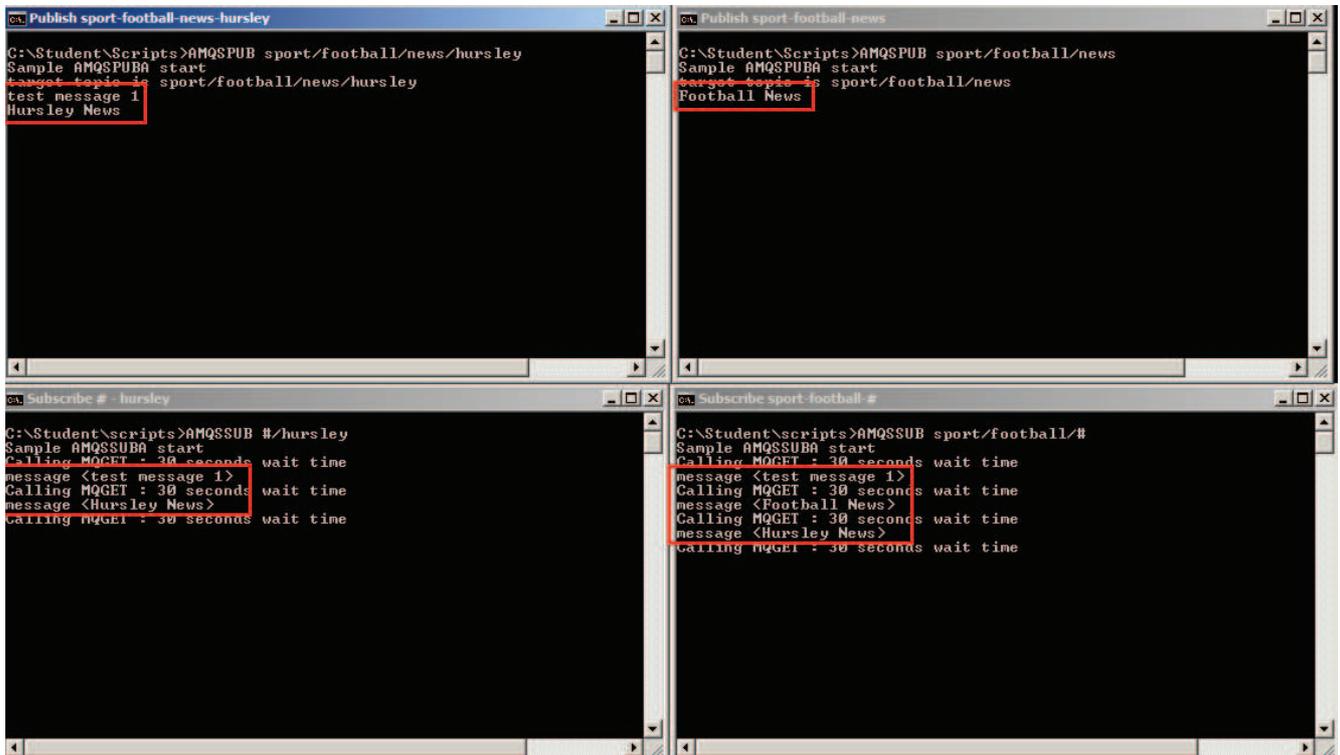


- __50. Now in the top left window (publishing to **sport/football/news/hursley**) enter **“test message 1”** and press Enter. The message should appear in *both* subscribing windows because the published message matched both subscriptions.

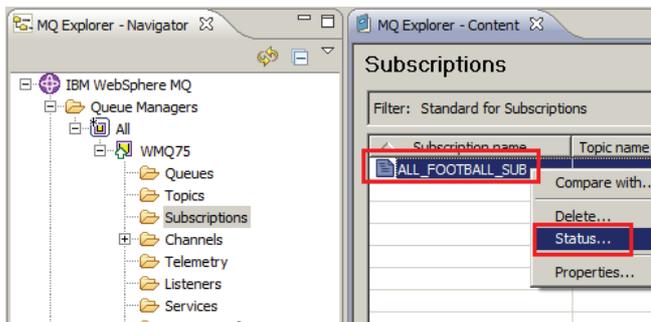
Again in the top left window type the text **Hursley News** and then press **Enter**.

In the top right window type the text **Football News** and press **Enter**. Notice that the **sport/football/#** subscription gets both publications. This is because when you subscribed you used a *multi-level wildcard* (#) to indicate that you were interested in messages published to the **sport/football** topic or any of its children, so you will get both messages.

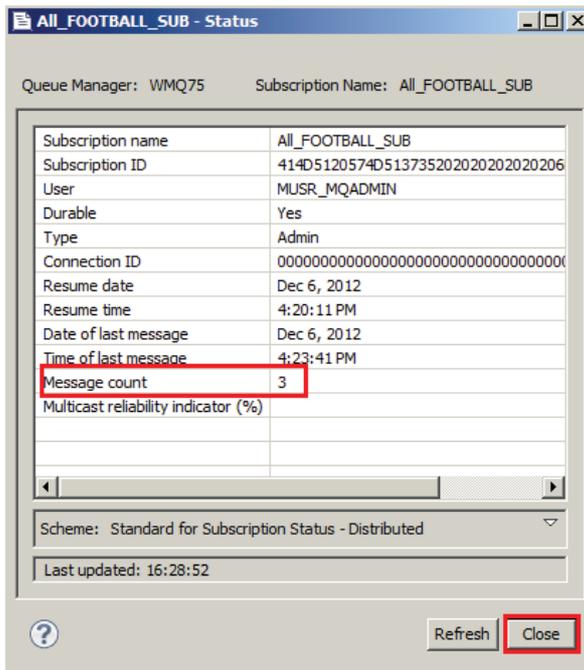
On the other hand, the **#hursley** subscription gets only one.



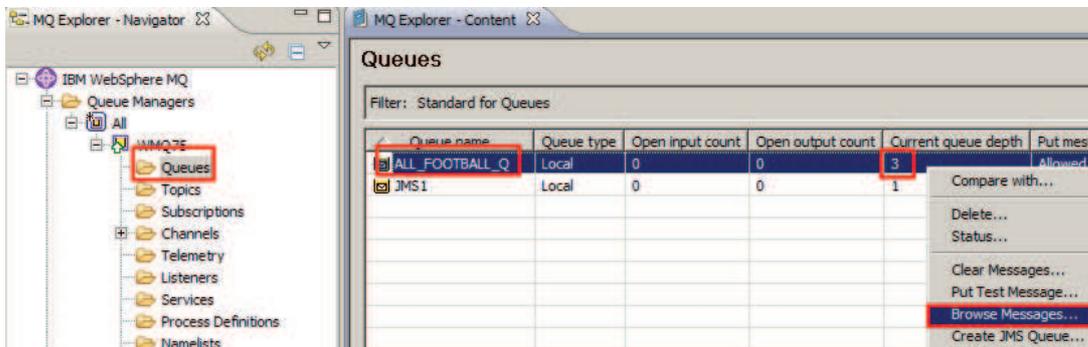
- __51. Return to the Subscriptions view in the WebSphere MQ Explorer. Select the **ALL_FOOTBALL_SUB** subscription, right-click and select **Status**.



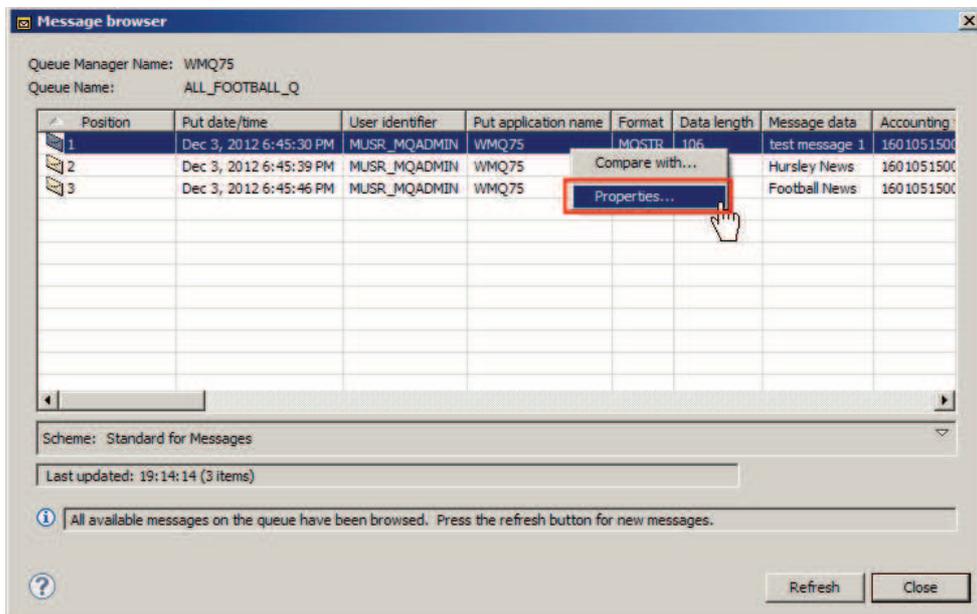
- __52. The message count should have a count of the messages that were published on this topic. Click on **Close** to close the status window.



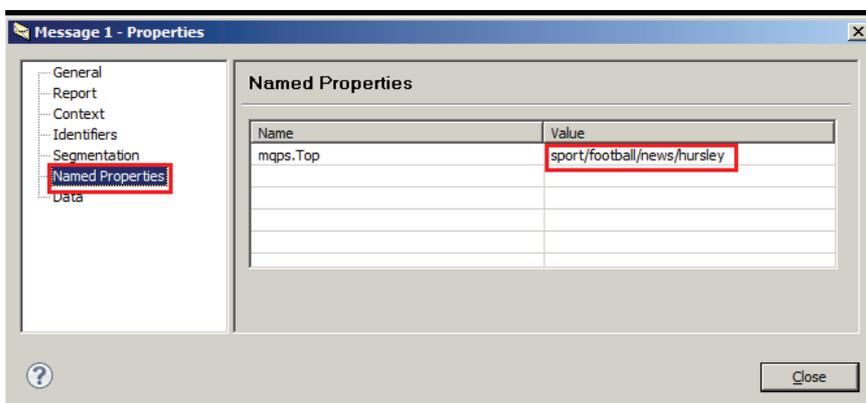
- __53. Right-click on the queue **ALL_FOOTBALL_Q** and select **Browse Messages**.



- __54. You should see three messages on the queue (or as many as you put in the amqspub test). Select one of the messages, right-click and choose **Properties**.



- __55. Click on the **Named Properties** tab. From this display you can see the originating topic string.



- __56. Close the four or five open command windows as you will no longer need them.

This concludes Lab 3.