

WebSphere Lab Jam

Connectivity

WebSphere MQ

Lab Exercises



An IBM Proof of Technology

Catalog Number

Contents

LAB 1	INTRODUCTION TO WEBSHERE MQ	5
	1.1 CREATE A QUEUE MANAGER	5
	1.2 TESTING BASIC FUNCTIONALITY	9
	1.3 GROUPING QUEUE MANAGERS	18
	1.4 DISTRIBUTED QUEUING	33
	1.5 LAB CLEANUP.....	51
LAB 2	CONFIGURING THE WEBSHERE MQ JMS PROVIDER.....	59
	2.1 CREATE ADMINISTERED OBJECTS USING MQ EXPLORER	59
	2.2 CREATE A CONNECTION FACTORY FOR WEBSHERE MQ.....	62
	2.3 CREATE A DESTINATION FOR THE JMS APPLICATION TO PUT A MESSAGE TO.....	68
	2.4 WRITING A JMS MESSAGE USING A JAVA SAMPLE PROGRAM	73
LAB 3	PUBLISH / SUBSCRIBE ADMINISTRATION LAB	77
	3.1 LAB OVERVIEW.....	77
	3.2 USING MQ EXPLORER TO CREATE AND DISPLAY INFORMATION	77
	3.3 A FIRST LOOK AT THE MQ EXPLORER PUB / SUB TEST TOOLS.....	84
	3.4 ADMINISTERED SUBSCRIPTIONS	92
	3.5 TESTING PUBLICATIONS AND SUBSCRIPTIONS FROM THE COMMAND LINE	98
LAB 4	WEBSHERE MQ SECURITY LAB.....	104
	4.1 WEBSHERE MQ SECURITY LAB OVERVIEW	104
	4.2 REVIEW THE SECURITY ON THE SYSTEM.....	104
	4.3 USING RUNAS TO CHANGE RUNTIME AUTHORITY.	104
	4.4 POINT TO POINT (QUEUE) SECURITY	106
	4.5 PUB/SUB (TOPIC) SECURITY.....	110
	4.6 USING THE MQ EXPLORER TO MANAGE SECURITY.....	113
LAB 5	WEBSHERE MQ HTTP BRIDGE	117
	5.1 CREATE THE REQUIRED QUEUES	117
	5.2 START THE HTTP LISTENER	119
	5.3 TEST THE LISTENER USING cURL.....	122
	5.4 POSTING MESSAGES USING HTTP POST.....	122
	5.5 A VERY BASIC PUT	124
	5.6 A PUT WITH AN MQ PROPERTY SPECIFIED IN THE HEADERS	126
	5.7 REQUESTING INFORMATION TO BE RETURNED	129
	5.8 USING cURL TO BROWSE MESSAGES.....	131
	5.9 USING THE HTTP BRIDGE WITH JAVASCRIPT	132
	5.10 USING THE PUTQ JAVASCRIPT	135
APPENDIX A.	NOTICES	129
APPENDIX B.	TRADEMARKS AND COPYRIGHTS	131

THIS PAGE INTENTIONALLY LEFT BLANK

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.

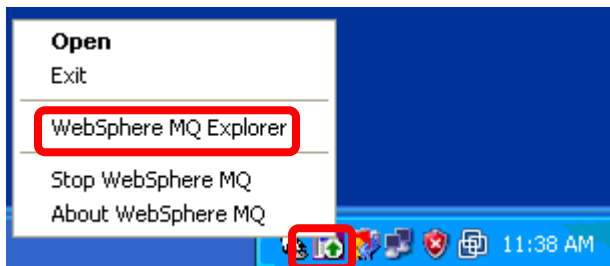
1.1 Create a Queue Manager

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

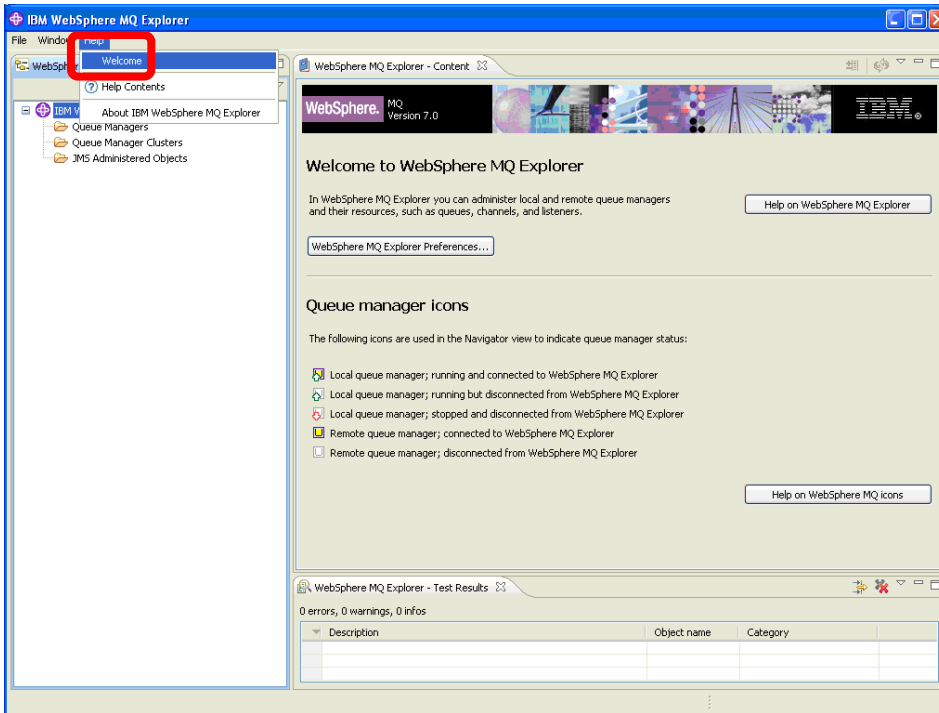
- __1. The indicated icon in the lower right hand corner represents WebSphere MQ on this system.



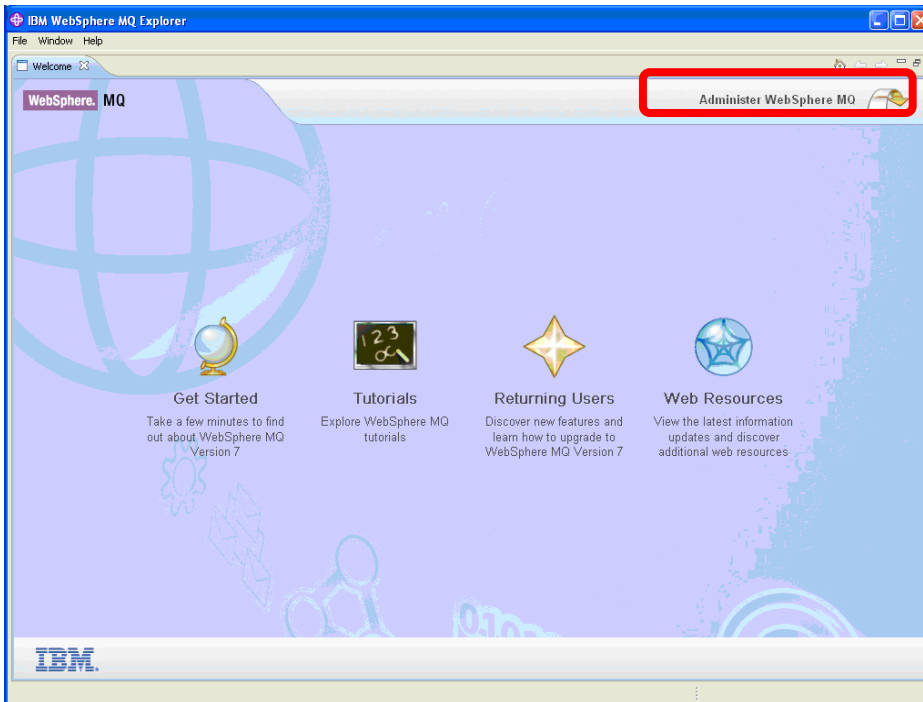
- __2. Start the MQ Explorer by **right-clicking** on the icon and selecting **WebSphere MQ Explorer**



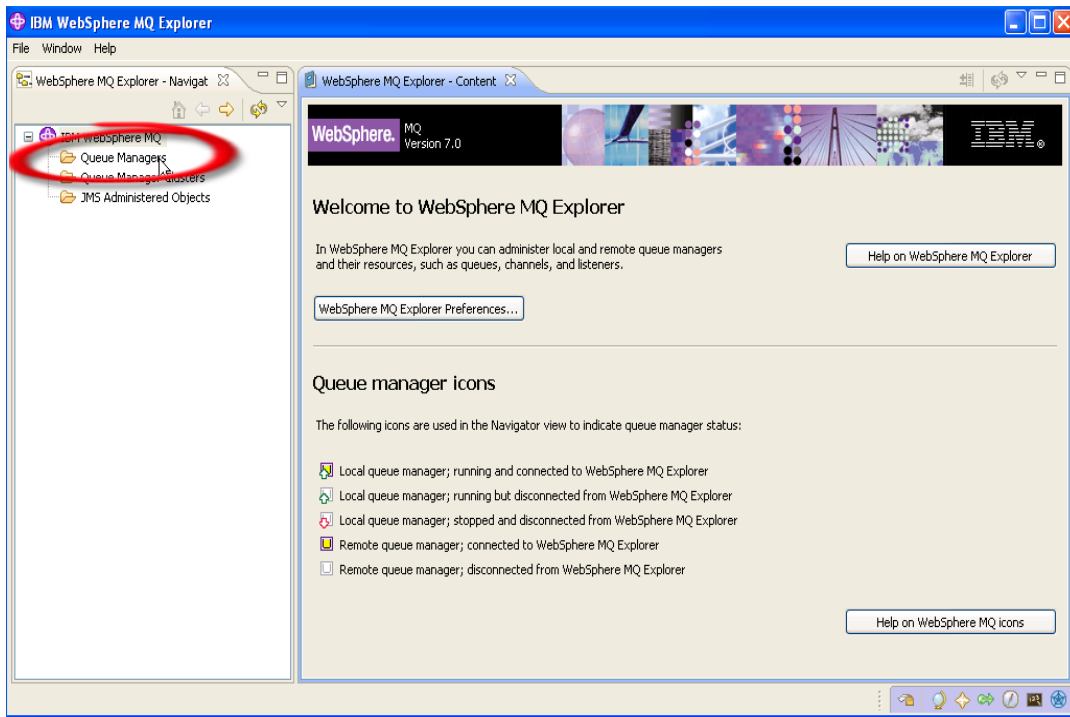
- ___3. The welcome screen provides a nice selection of resources for the product. On the Menu Bar, click on **Help** and select **Welcome** to bring it into view



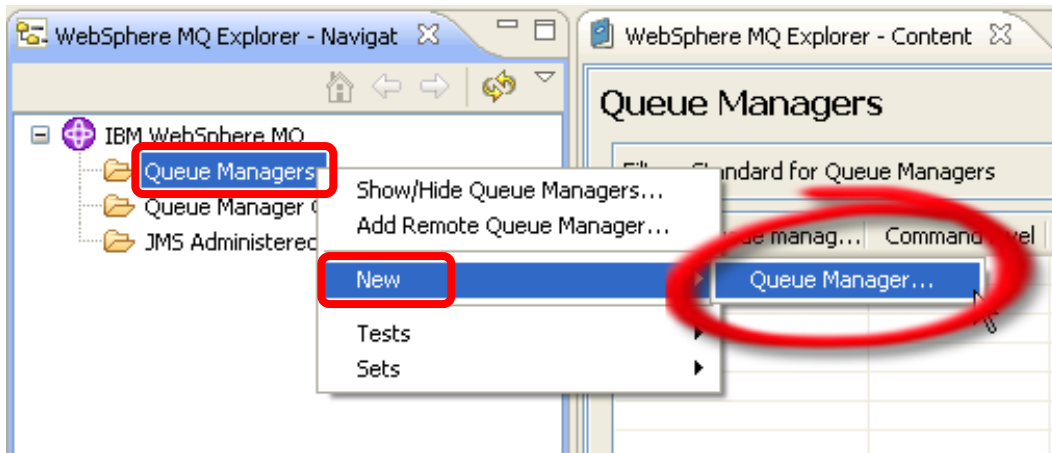
- ___4. Note the various options on the Welcome screen and explore them if you would like. Then return to the MQ Explorer workbench by clicking the **Administer WebSphere MQ** icon in the upper right-hand corner. The first time you launch MQ Explorer after an install of WebSphere MQ this Welcome screen will be displayed automatically.



- ___5. The left-hand pane in MQ Explorer is called the **Navigation pane**. In the navigation pane, locate the **Queue Managers** folder.



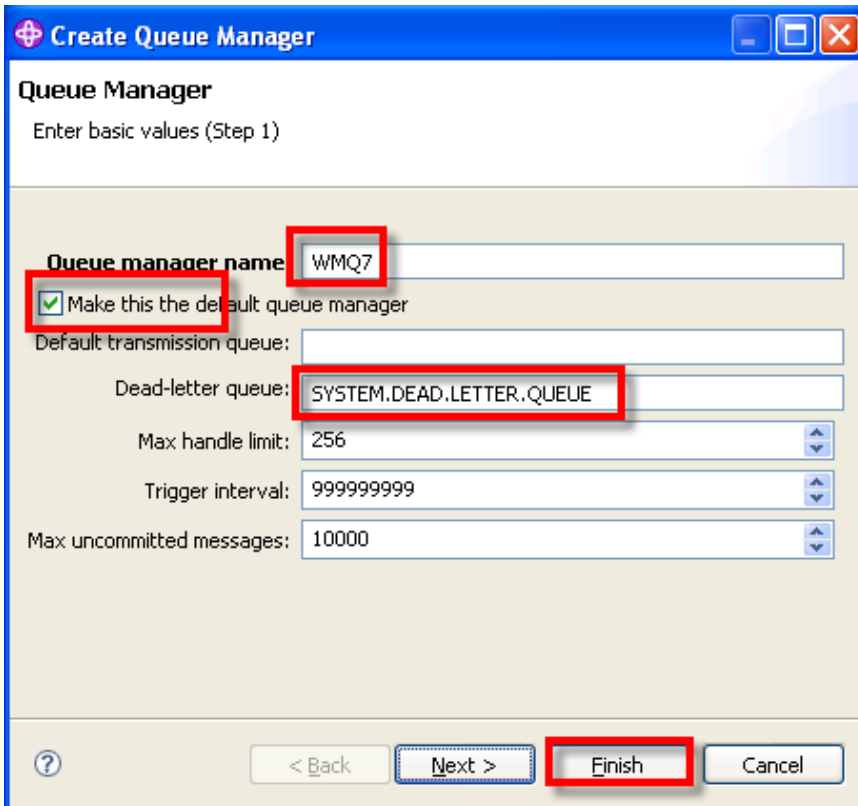
- ___6. Right-click on **Queue Managers** and select **New** then **Queue Manager...**



- __7. Create a new queue manager and name it **WMQ7**. 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!!!! 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!!!!

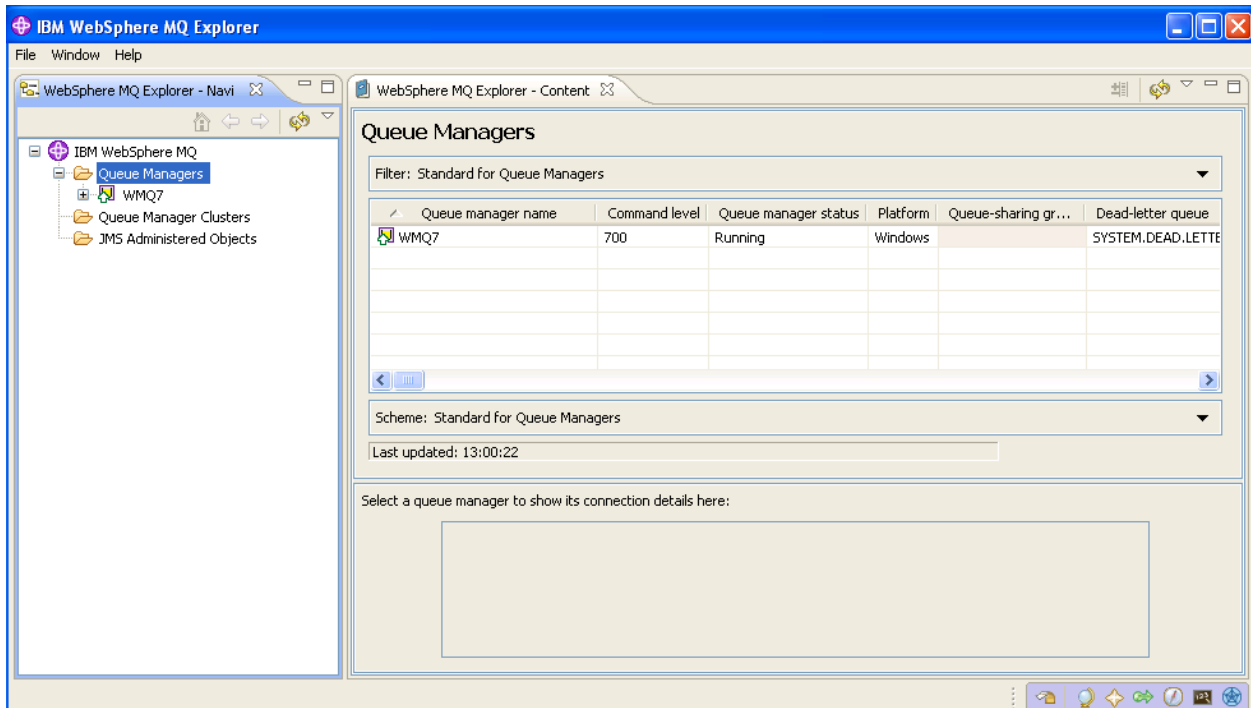
Specify **SYSTEM.DEAD.LETTER.QUEUE** as the **dead letter queue**, then click the **Finish** button.



- __8. The create queue manager process takes a few seconds during which time the following panel is displayed...



- __9. The newly created queue manager is displayed in MQ Explorer. In the Content pane on the right-hand side you can see details about the queue manager, such as its status, the name of the dead letter queue, etc.



This concludes this portion of Lab 1.

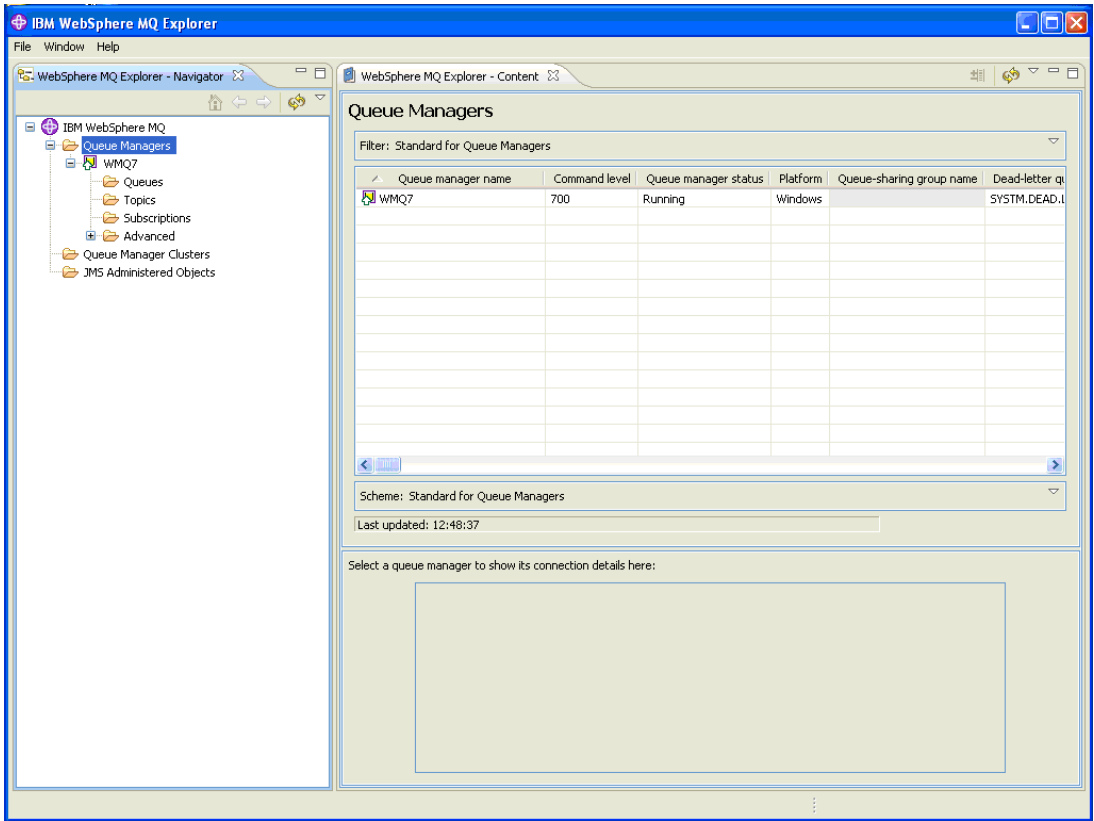
1.2 Testing Basic Functionality

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.

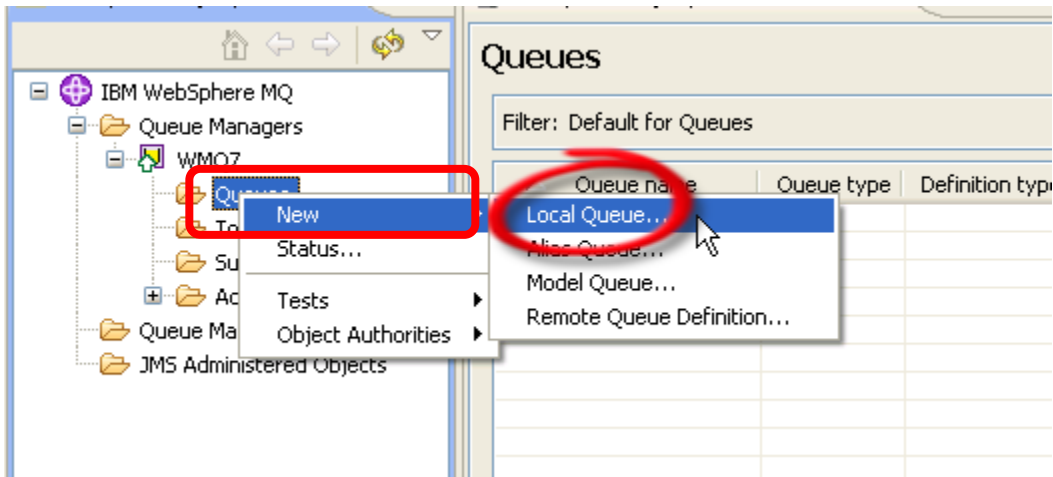
- __1. Expand the new **WMQ7** queue manager by clicking on the “+” symbol in front of it in the navigation tree.



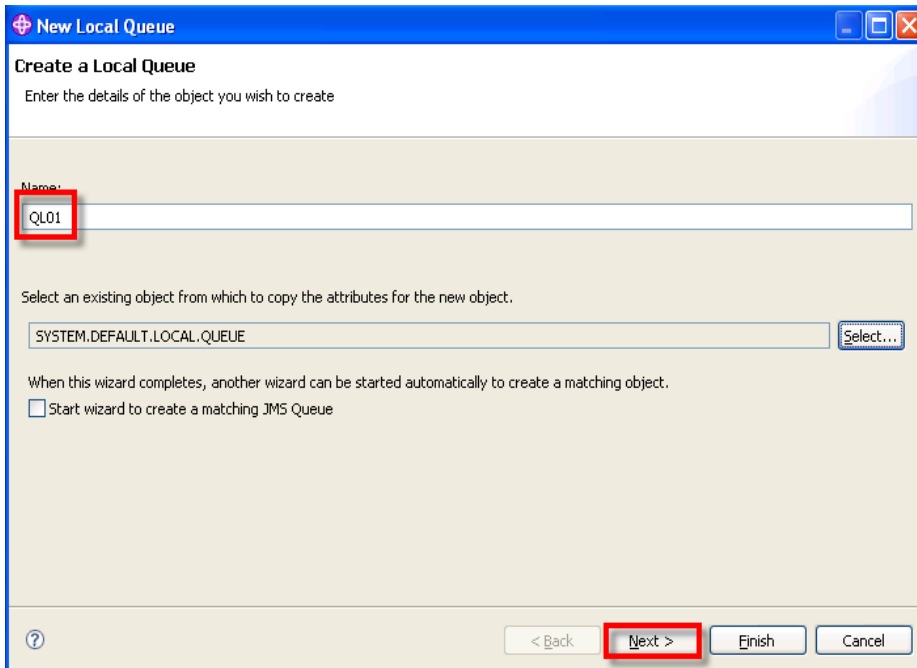
- 2. Observe the tree structure that is displayed in the Navigator pane. There are separate folders for *Queues*, *Topics* and *Subscriptions*, as well as one labeled *Advanced*.



- 3. Now you will create a new queue. Within the WMQ7 queue manager **right-click** on **Queues** then select **New** then **Local Queue**.

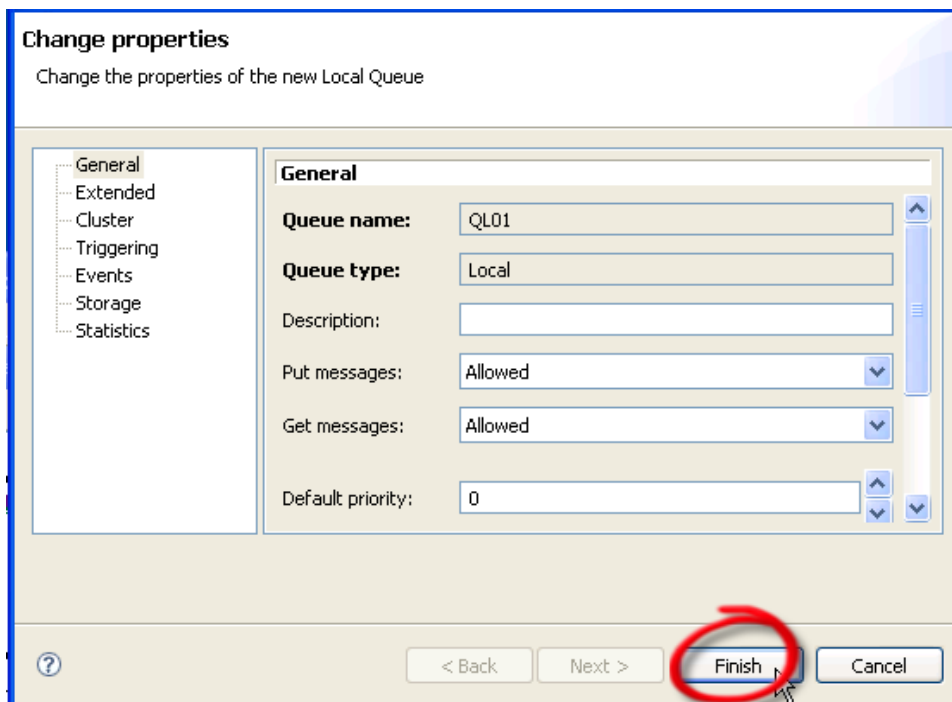


- ___4. Name the new queue **QL01** then click on the **Next** button.



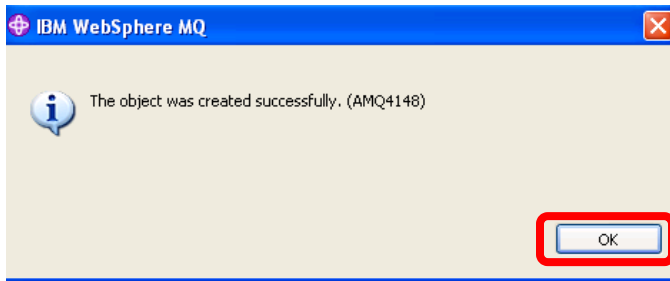
The screenshot shows the 'New Local Queue' wizard window. The title bar reads 'New Local Queue'. The main heading is 'Create a Local Queue' with the instruction 'Enter the details of the object you wish to create'. The 'Name:' field is highlighted with a red box and contains the text 'QL01'. Below this, there is a section 'Select an existing object from which to copy the attributes for the new object.' with a dropdown menu showing 'SYSTEM.DEFAULT.LOCAL.QUEUE' and a 'Select...' button. A checkbox labeled 'Start wizard to create a matching JMS Queue' is unchecked. At the bottom, the '< Back' button is disabled, the 'Next >' button is highlighted with a red box, and the 'Finish' and 'Cancel' buttons are also visible.

- ___5. 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.

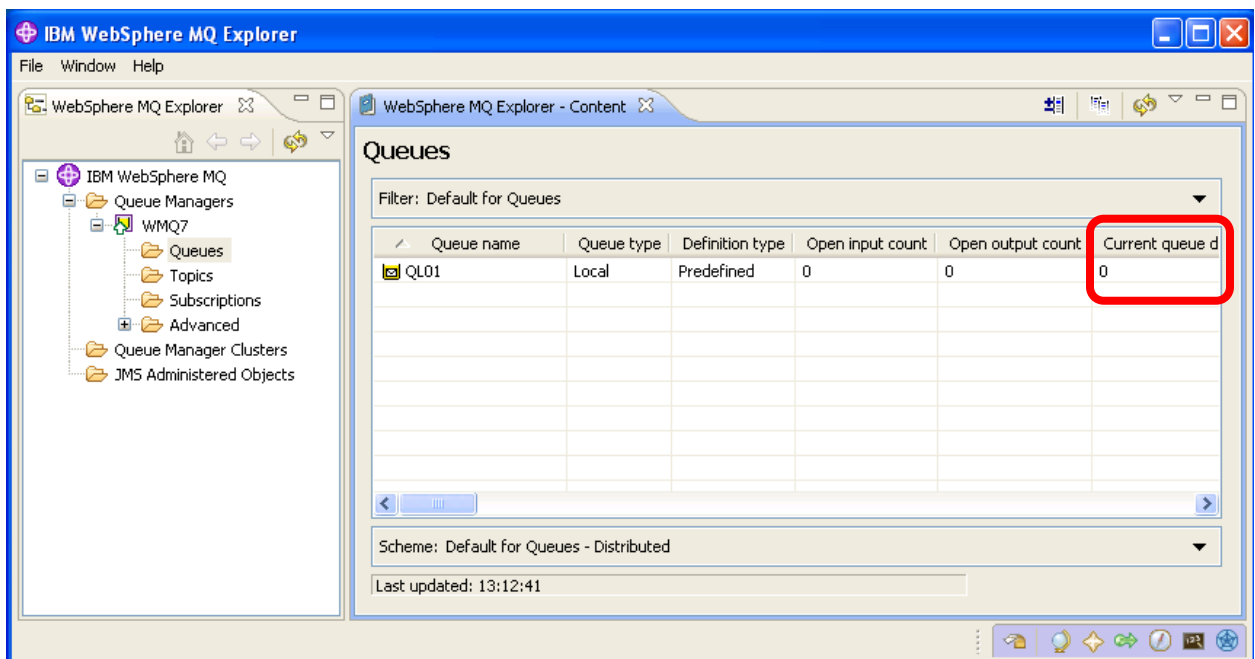


The screenshot shows the 'Change properties' dialog for the new Local Queue. The title bar reads 'Change properties' with the instruction 'Change the properties of the new Local Queue'. On the left, there is a tree view with tabs: General, Extended, Cluster, Triggering, Events, Storage, and Statistics. The 'General' tab is selected. The main area shows the following properties: 'Queue name:' with the value 'QL01', 'Queue type:' with the value 'Local', 'Description:' with an empty text box, 'Put messages:' with a dropdown menu set to 'Allowed', 'Get messages:' with a dropdown menu set to 'Allowed', and 'Default priority:' with the value '0'. At the bottom, the '< Back' button is disabled, the 'Next >' button is disabled, the 'Finish' button is highlighted with a red circle, and the 'Cancel' button is also visible.

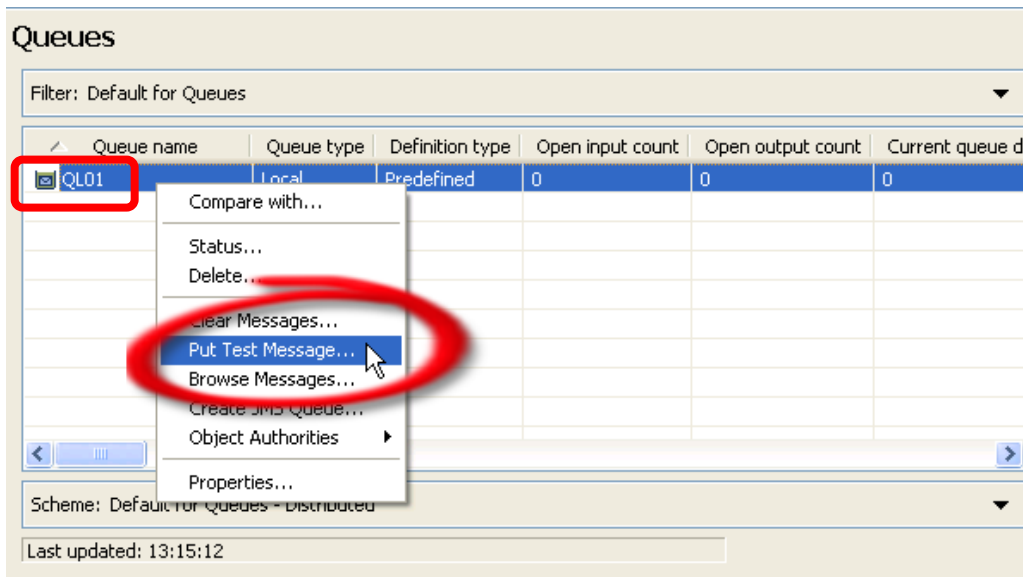
__6. Dismiss the completion panel by clicking on the **OK** button.



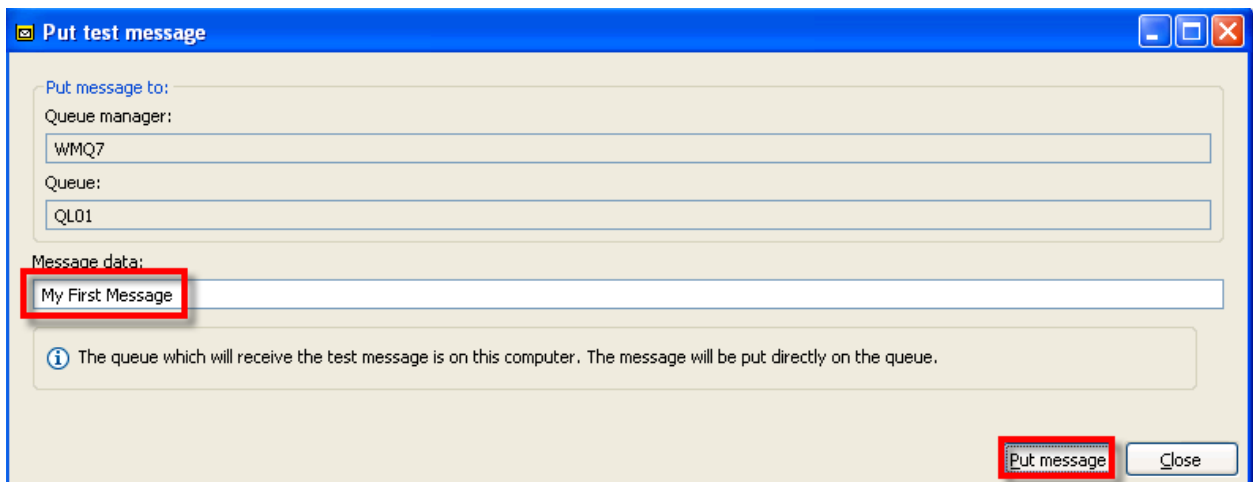
__7. In the Content pane on the right-hand side the newly created queue is displayed along with its various properties. The **Current queue depth** property is a measure of how many messages are currently in the queue. Note that the current queue depth is 0 as you would expect since you just created the queue.



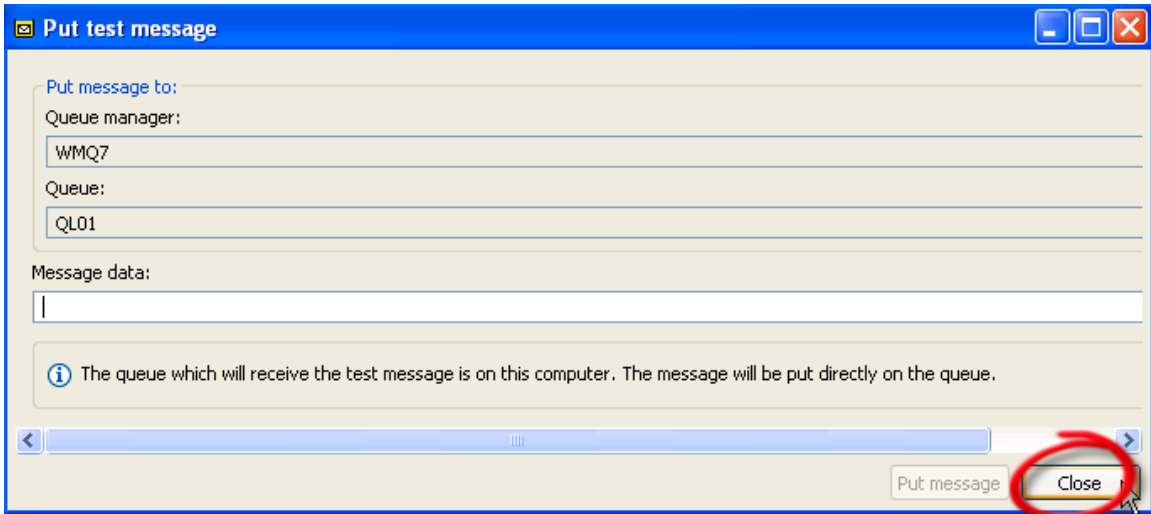
- __8. Next you will place a message in the queue. **Right-click** on the **QL01** queue and select **Put Test Message**.



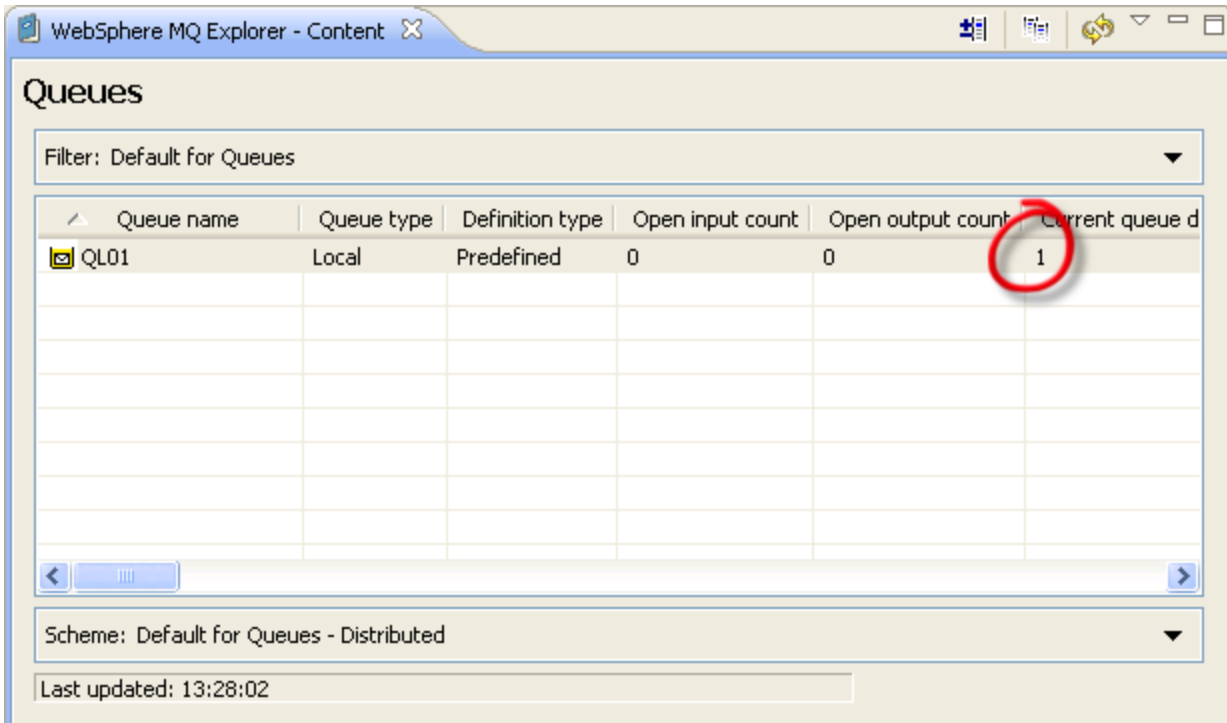
- __9. Enter a test message by typing something of your choosing in the **Message data** field and click on the **Put message** button.



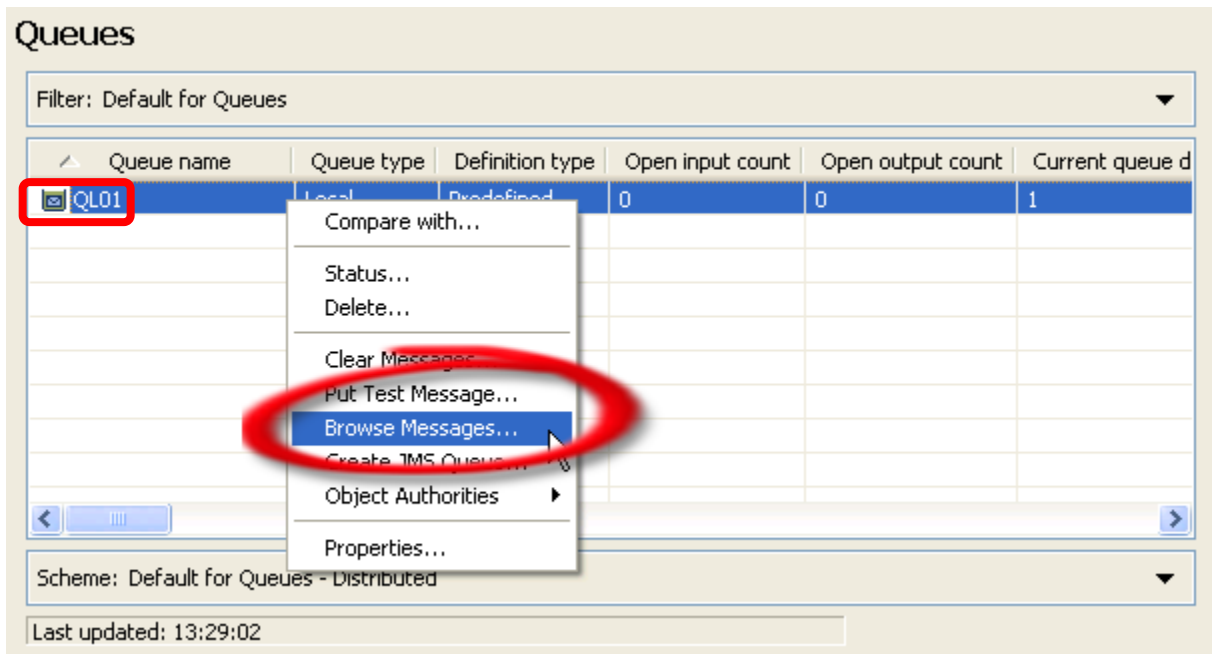
__10. Close the Put test message panel by clicking on the **Close** button.



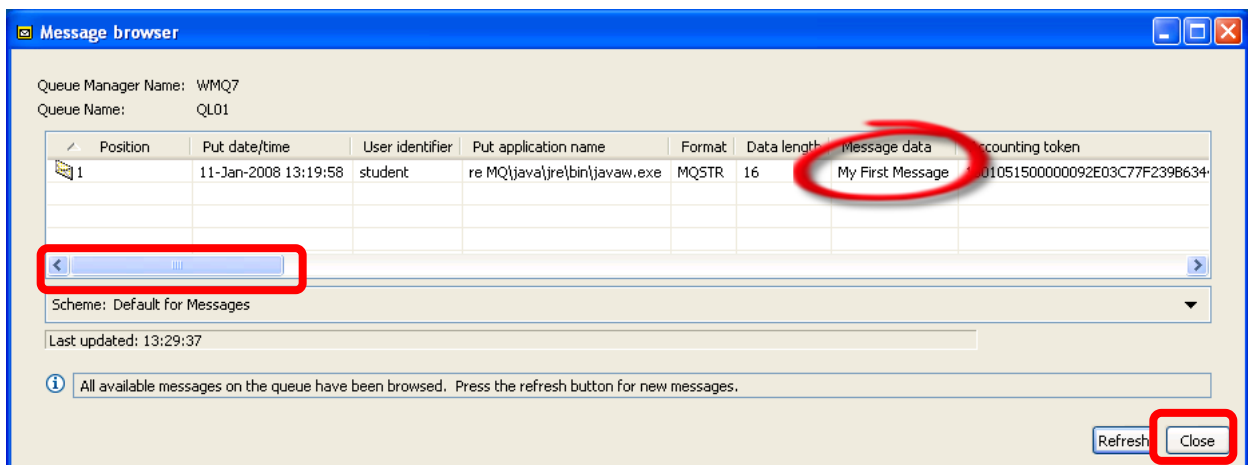
When you return to the display of queues, note that the **Current queue depth count is now 1**. This represents the test message you just placed into the queue.



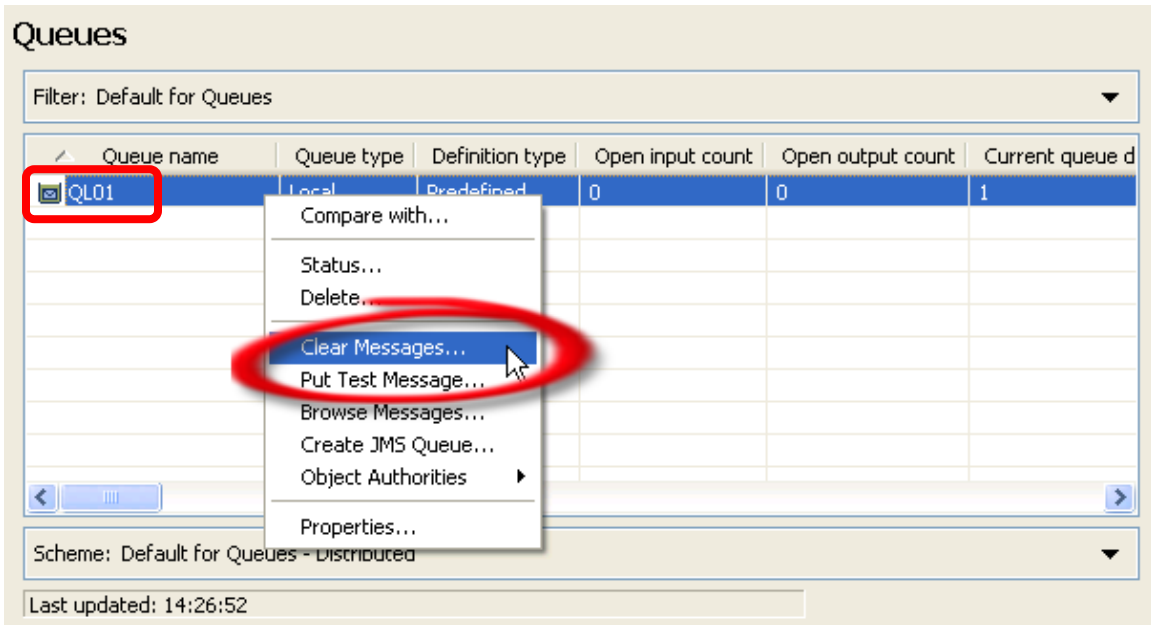
- __11. You can view details about the messages on a queue – **right-click** on queue **QL01** and select **Browse Messages**.



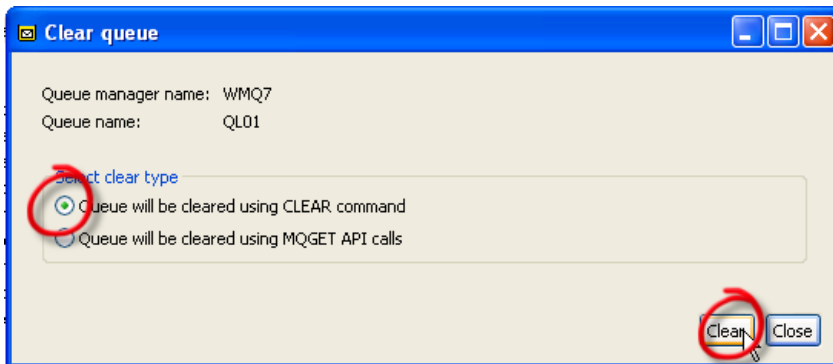
- __12. Here is the resulting display. Note that there is a **scroll bar** at the bottom that allows you to view additional data that is not currently in view. There will be one such line displayed for each message in the queue. Click on the **Close** button.



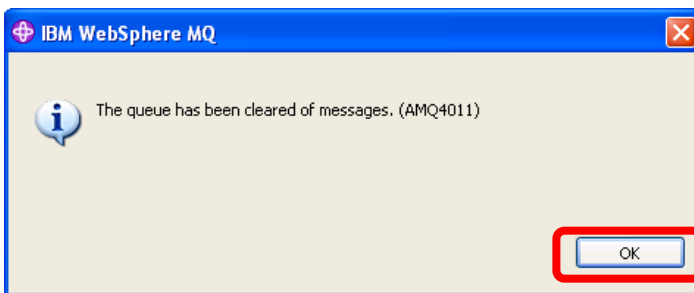
- __13. You will now delete your test message from the queue. **Right-click** on queue **QL01** and select **Clear Messages**.....note that this action will remove all messages from the queue.



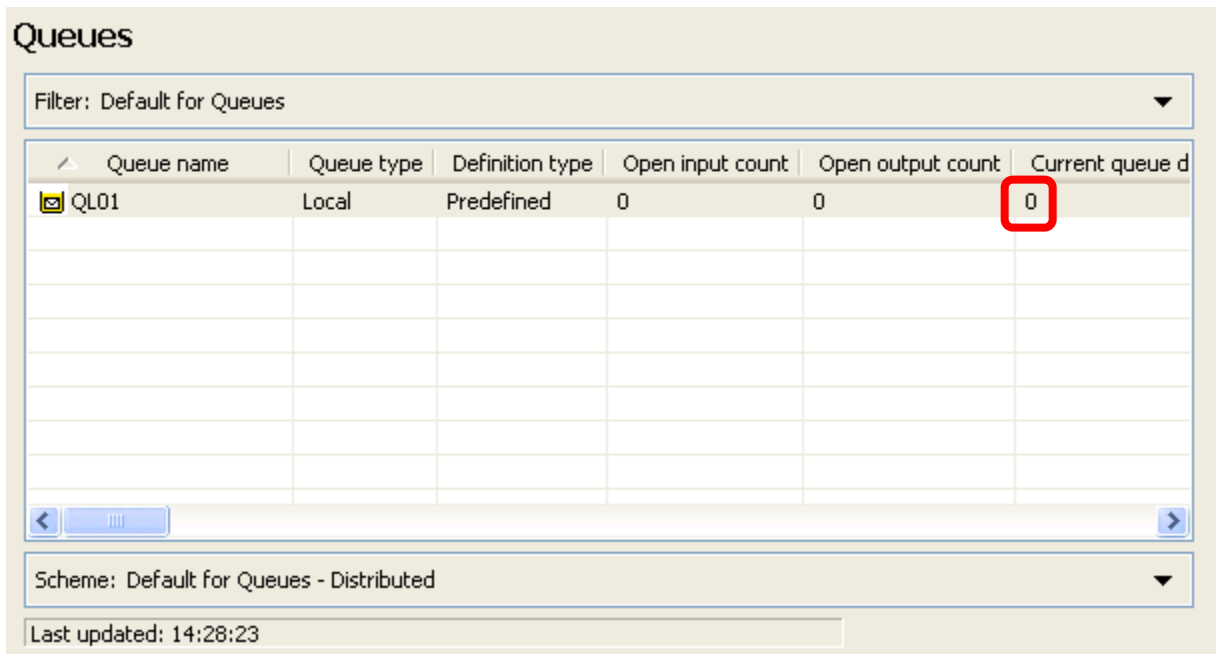
- __14. Accept the default selection of using the **CLEAR command**. This option cannot be used if any application has the queue open. In that case the second option would be required. Click on the **Clear** button.



- __15. Click on the **OK** button to dismiss the confirmation panel.



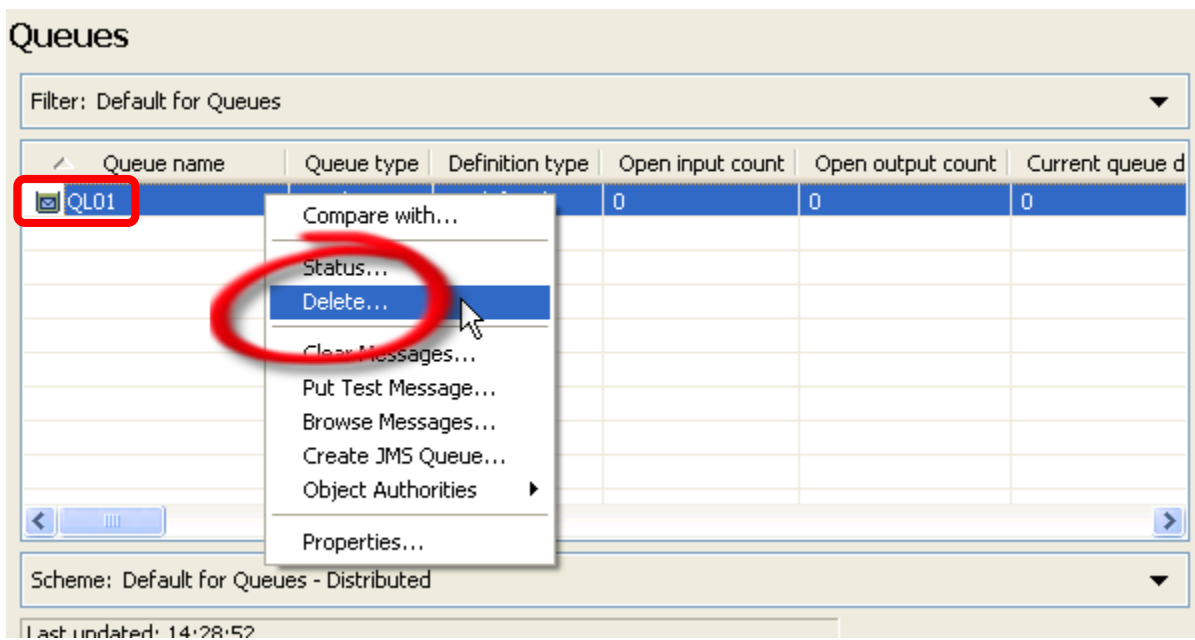
You can now see that the queue is indeed empty.



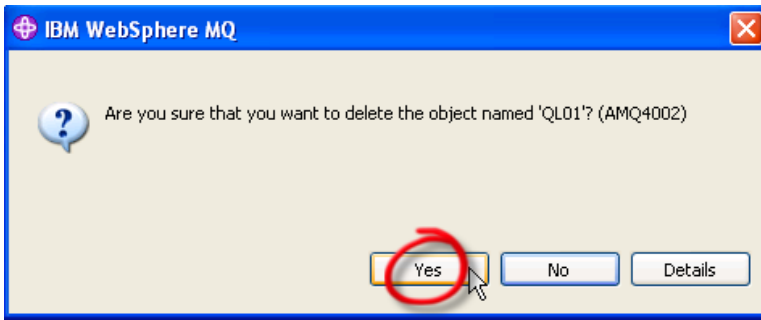
The screenshot shows the 'Queues' console with a table of queue information. The 'Current queue d' column for the 'QL01' queue is highlighted with a red box, showing the value '0'. The table has the following columns: Queue name, Queue type, Definition type, Open input count, Open output count, and Current queue d.

Queue name	Queue type	Definition type	Open input count	Open output count	Current queue d
QL01	Local	Predefined	0	0	0

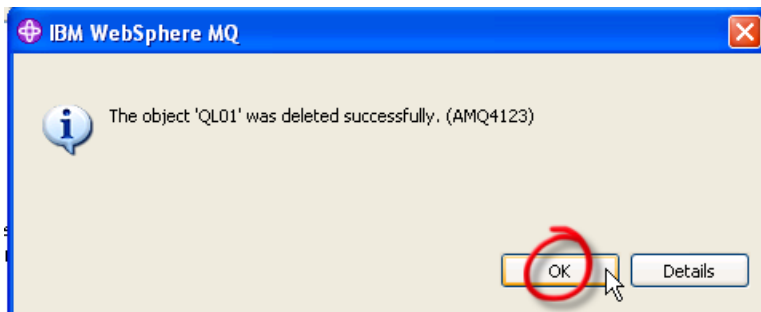
- __16. You will now delete the queue. **Right-click** on the **QL01** queue and select **Delete...**



__17. Confirm the request by clicking on the **Yes** button.



__18. Dismiss the confirmation panel by clicking on the **OK** button.



This concludes this portion of Lab 1.

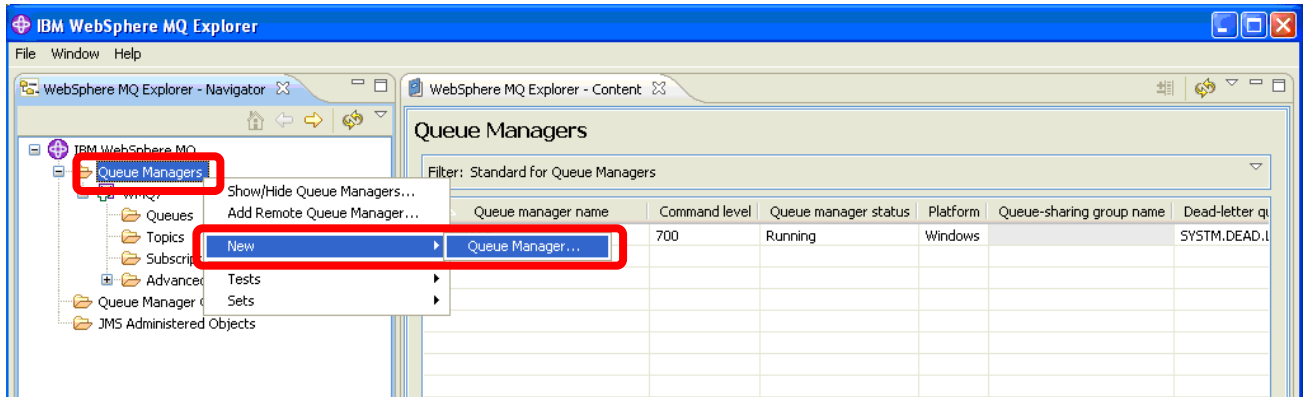
1.3 Grouping Queue Managers

In this portion of the lab you will use a feature of MQ Explorer that allows grouping of queue managers into Sets. This offers a convenient way to view and/or manage a subset of queue managers. A particular queue manager may be a member of one Set or many Sets or no Set at all. The only restriction is that a Set may not contain another Set.

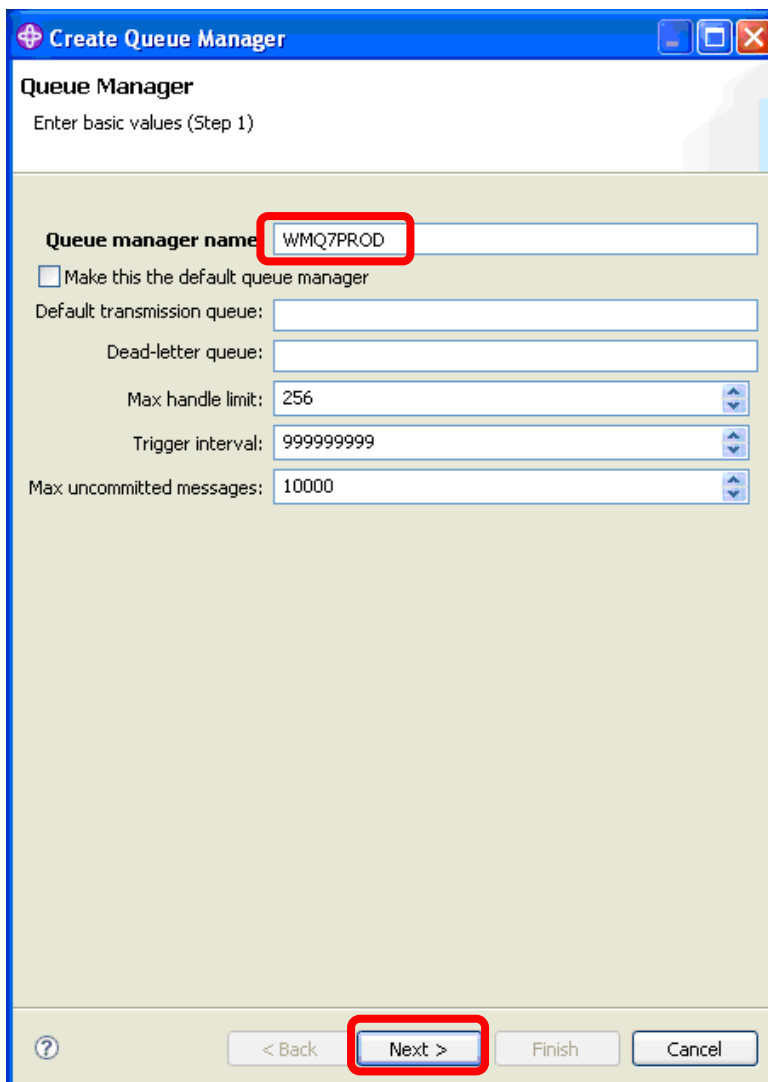
Using the MQ Explorer, you will now create the following queue manager and assign it the indicated port number for the listener:

WMQ7PROD → port number = 1415

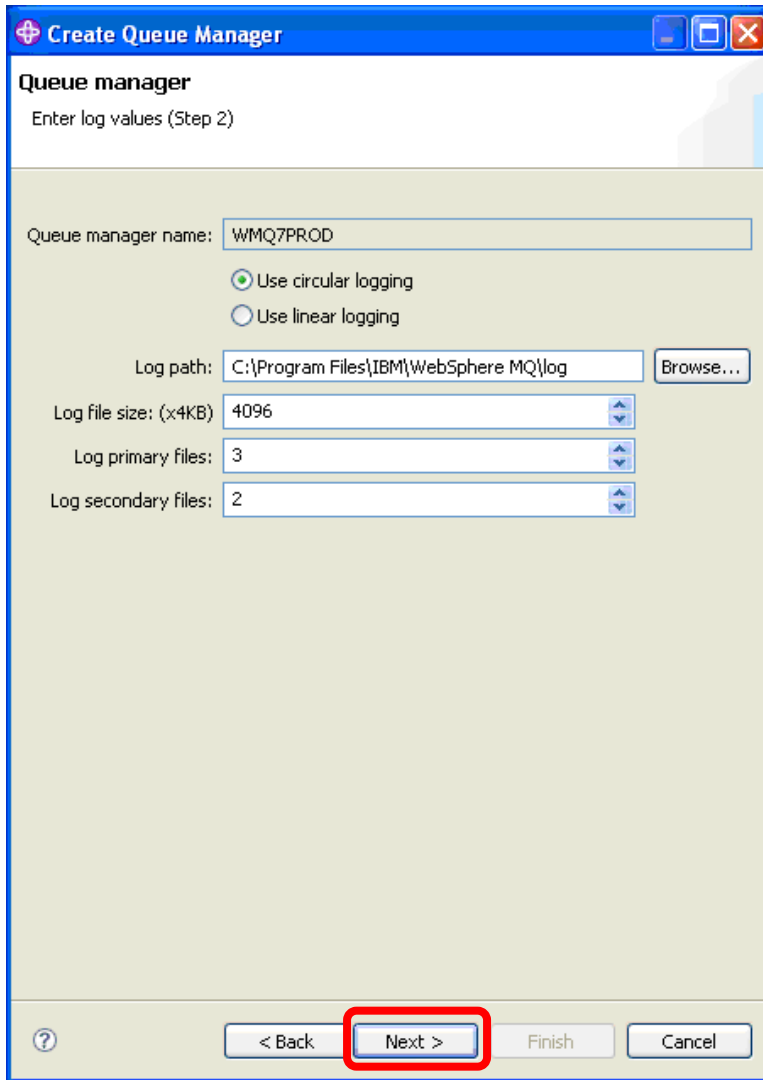
- __1. Right-click on **Queue Managers** and select **New**, then **Queue Manager**



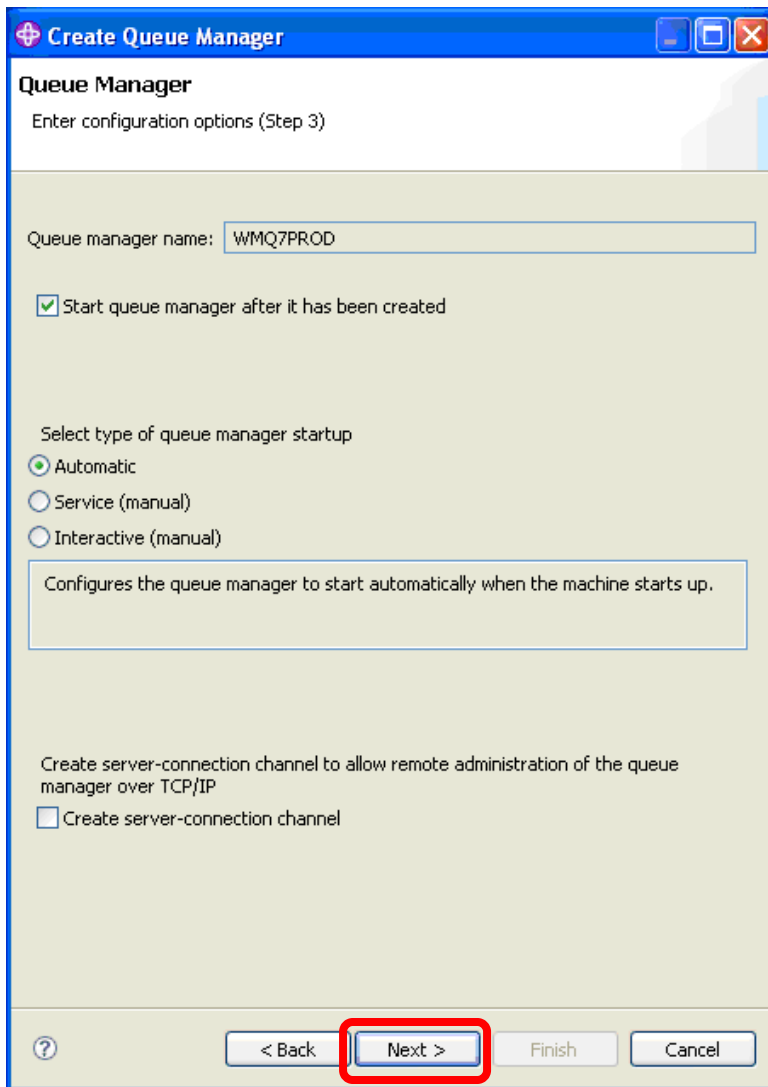
- __2. Enter **WMQ7PROD** as the name and click the **Next** button.



__3. Click the **Next** button to accept these defaults....



__4. Click the **Next** button to accept these defaults....



Create Queue Manager

Queue Manager

Enter configuration options (Step 3)

Queue manager name: WMQ7PROD

Start queue manager after it has been created

Select type of queue manager startup

Automatic

Service (manual)

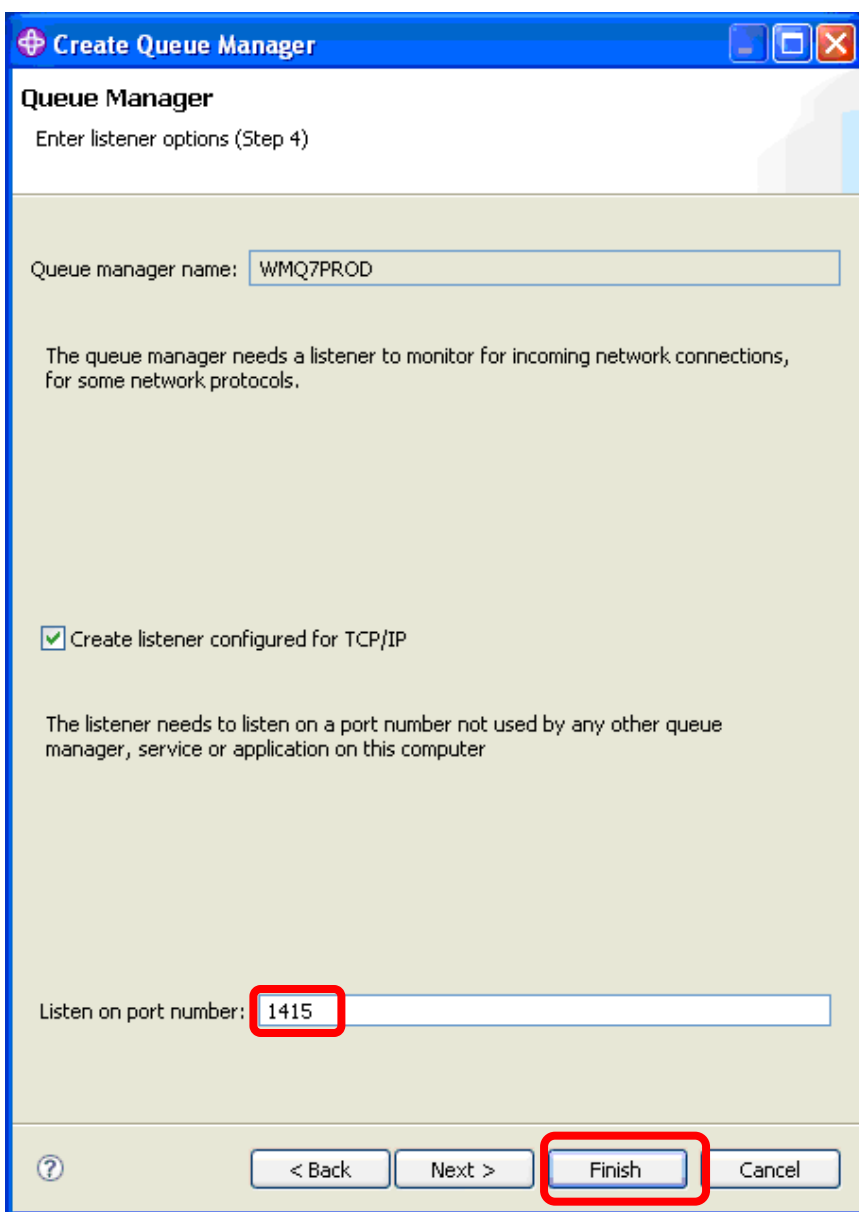
Interactive (manual)

Configures the queue manager to start automatically when the machine starts up.

Create server-connection channel to allow remote administration of the queue manager over TCP/IP

Create server-connection channel

- __5. Enter **1415** as the **port number** and click the **Finish** button.



Create Queue Manager

Queue Manager
Enter listener options (Step 4)

Queue manager name: WMQ7PROD

The queue manager needs a listener to monitor for incoming network connections, for some network protocols.

Create listener configured for TCP/IP

The listener needs to listen on a port number not used by any other queue manager, service or application on this computer

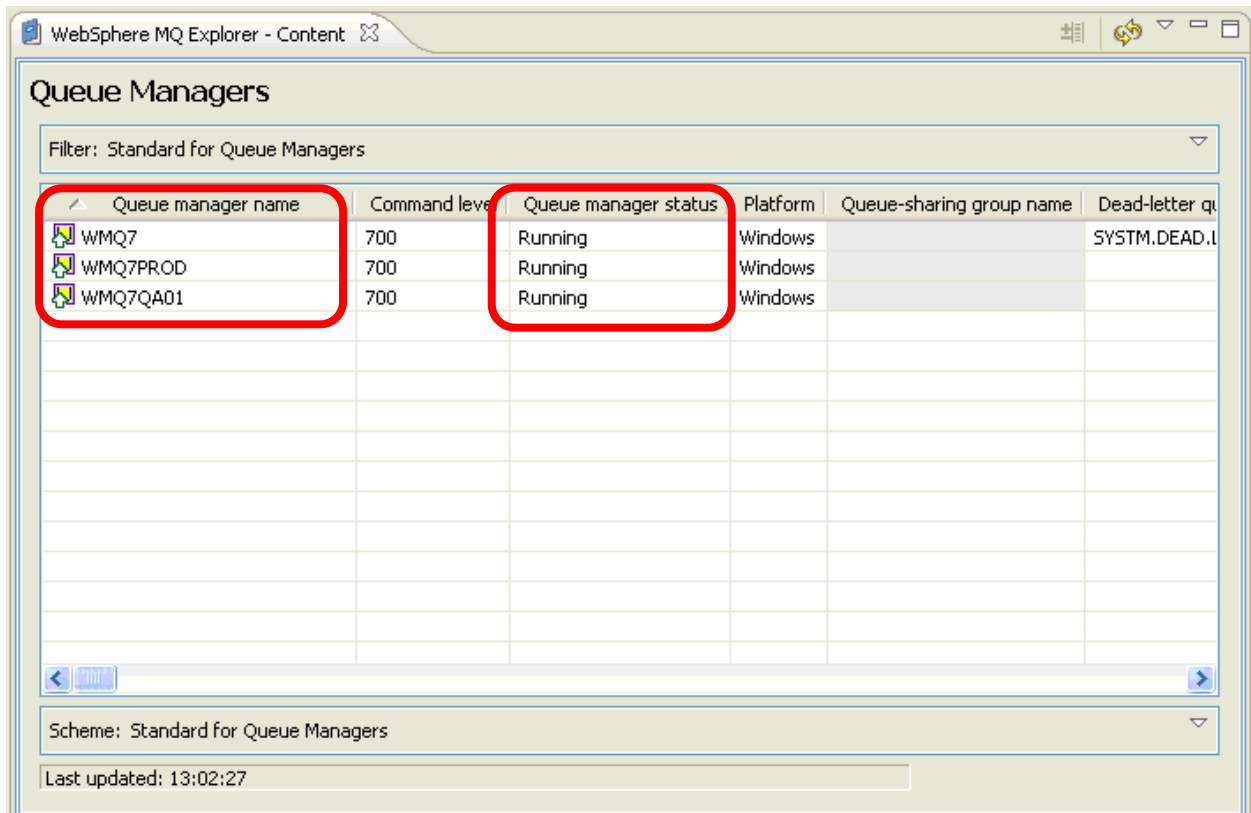
Listen on port number: 1415

< Back Next > **Finish** Cancel

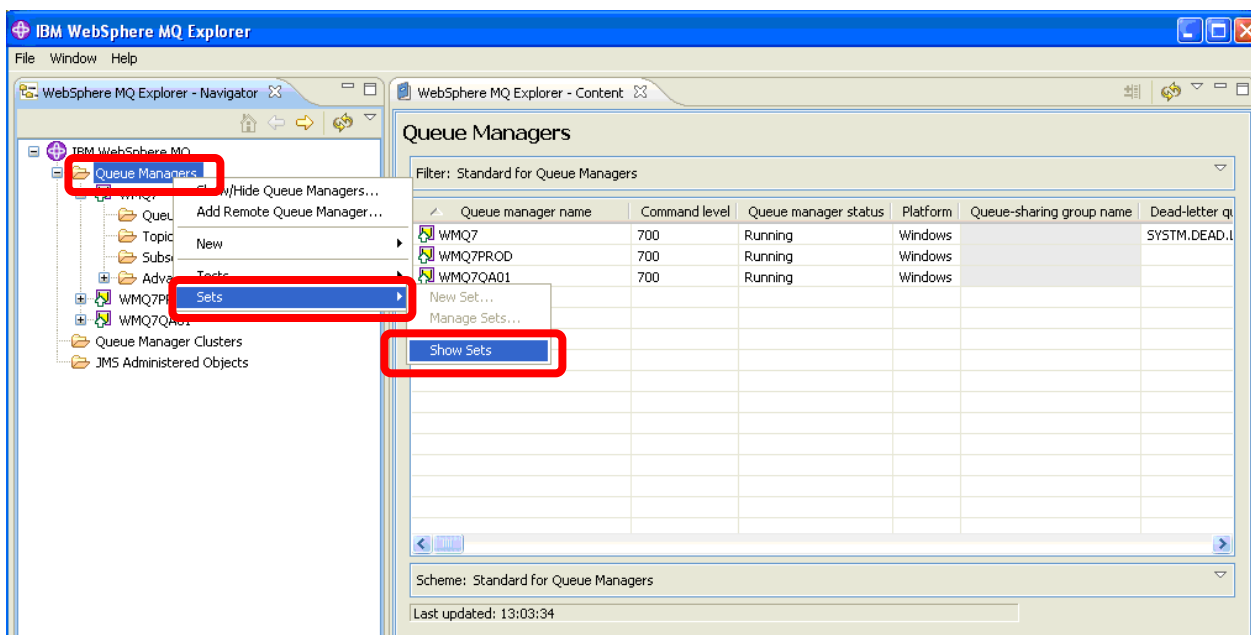
Using MQ Explorer and the steps shown above, create another queue manager and assign it the indicated port number for the listener:

WMQ7QA01 → port number = 1416

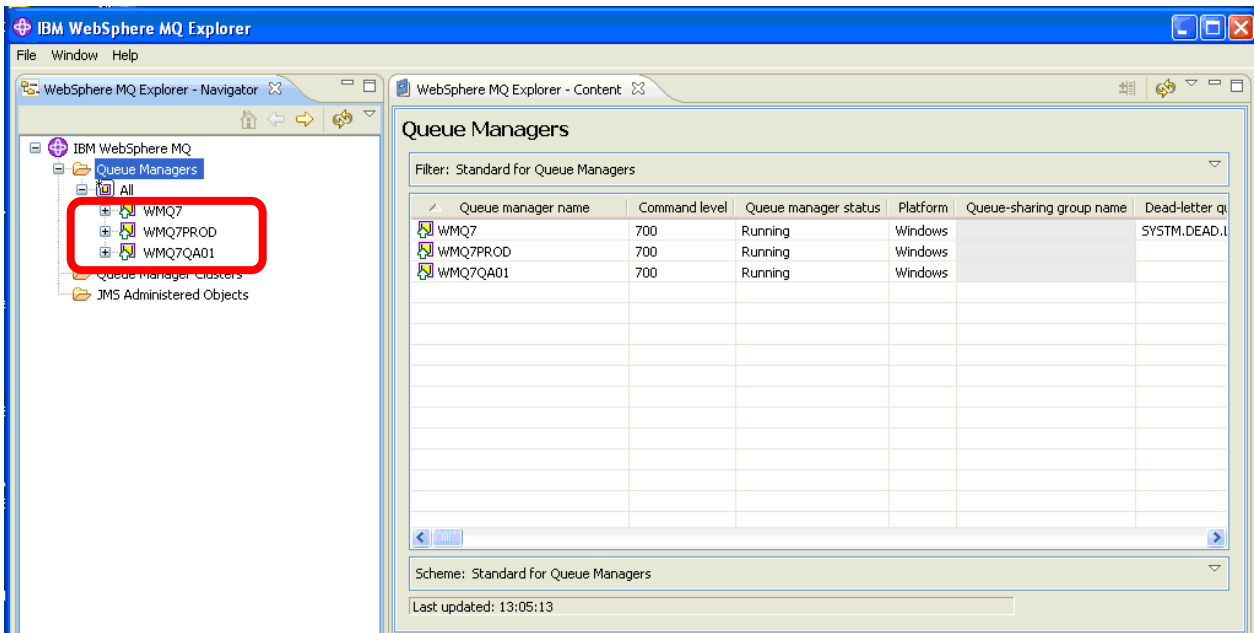
- ___6. Check the Content pane and make sure all three of the queue managers have been created, and that the **Queue manager status** for each is **Running**.



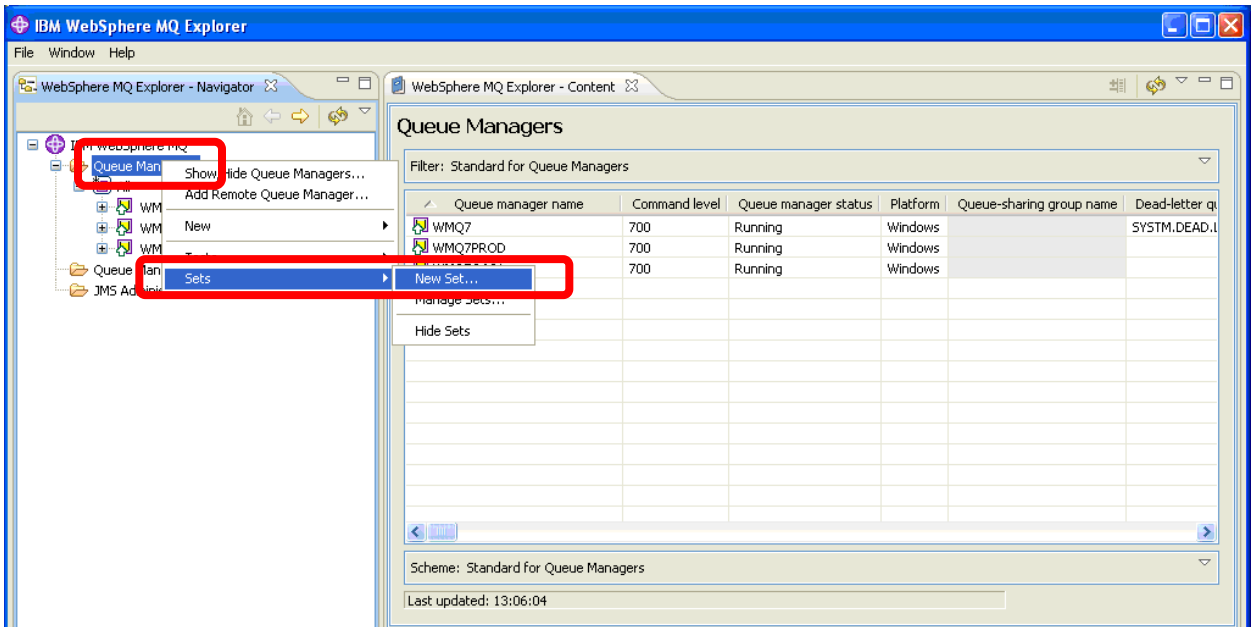
- ___7. To include Sets in the MQ Explorer display right-click on **Queue Managers** and select **Sets** then **Show Sets**



You now see that you have a **default set** called **All** that contains every queue manager that is defined.



__8. To define a new Set, right-click on **Queue Managers** and select **Sets** then **New Set**.



- ___9. Enter **PROD.QMGRS** as the name, leave the **Select type** as **Manual** and click on the **Next** button.

New Set

Create a Set
Select the name and type of the new set

Name:
PROD.QMGRS

Select type

Manual. Objects can be added to and removed from the set using menu actions, and using drag and drop

Automatic. Objects are automatically made members of the set if they match its filter conditions

< Back **Next >** Finish Cancel

- ___10. Click the check box for **WMQ7PROD** to indicate that this queue manager is to be included in the Set and click the **Finish** button

New Set

Create a Set
Select the queue managers that will be members of this set

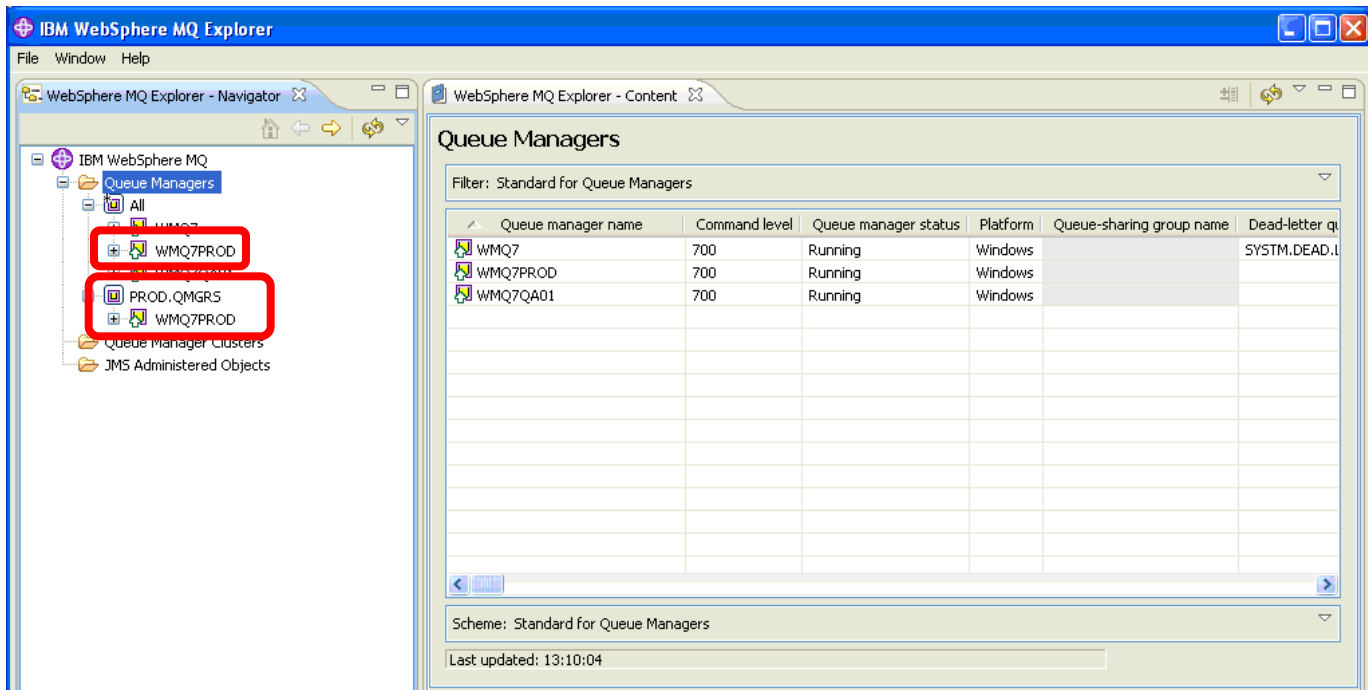
Queue manager name	Connection type	Connection name	Channel name	Channel definition table	Refresh interval
<input type="checkbox"/> WMQ7	Local				15
<input checked="" type="checkbox"/> WMQ7PROD	Local				15
<input type="checkbox"/> WMQ7PROD	Local				15

Last updated: 13:09:04

Select All Select None

< Back Next > **Finish** Cancel

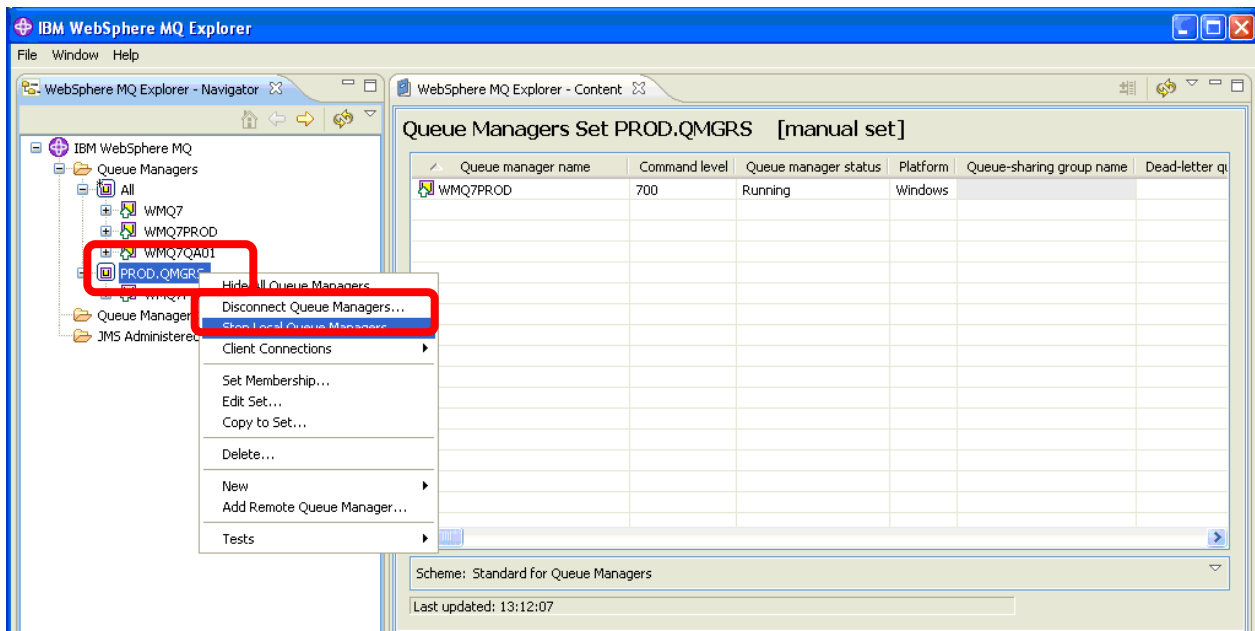
- ___11. Click on the plus sign to **expand** the **PROD.QMGRS** Set. Note that WMQ7PROD is a member of both the All Set and the PROD.QMGRS Set.



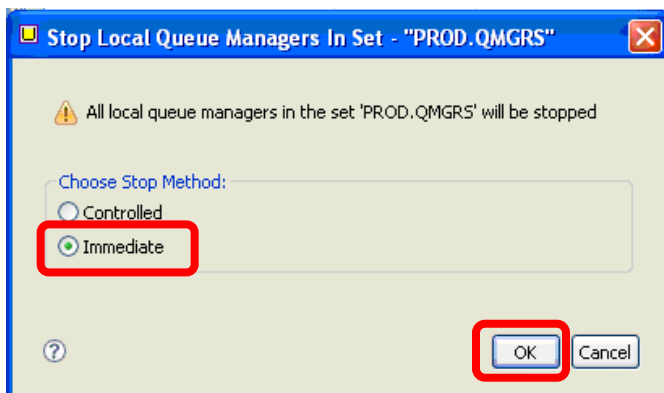
Once a Set has been created you can take an action against the entire Set of queue managers by issuing a single request. For example you can:

- Show/Hide All
- Connect/Disconnect All
- Start/Stop All Local
- Run Default/Custom Tests

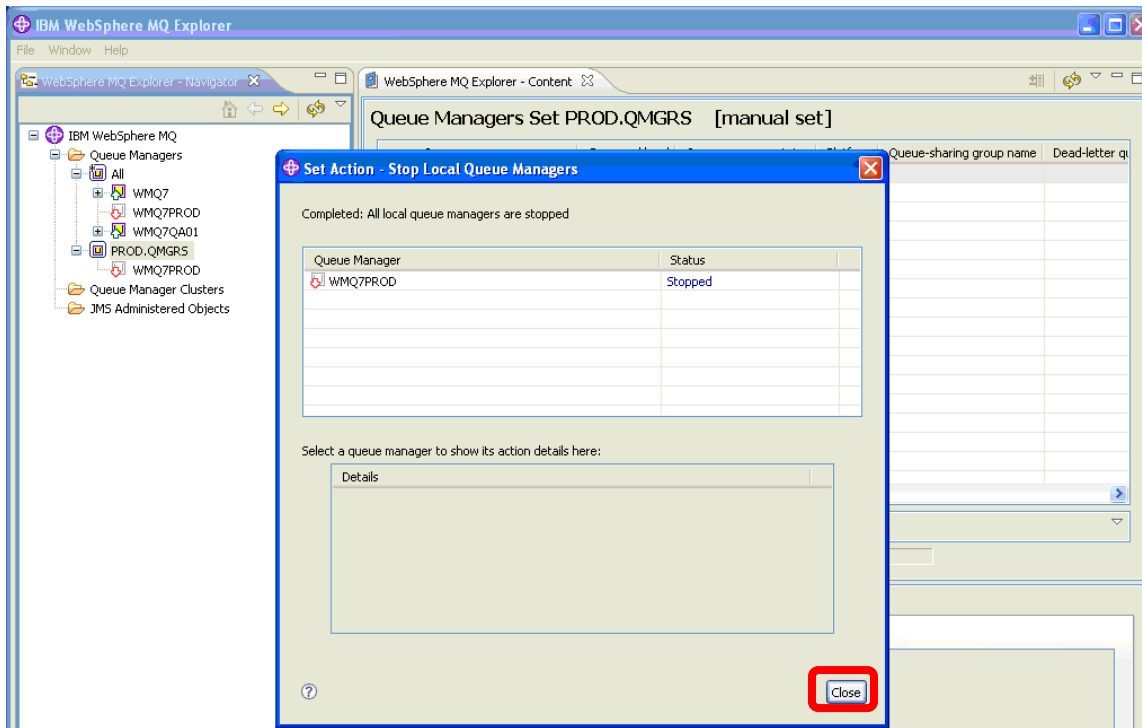
- __12. To take some joint action against a Set of queue managers right-click on the Set and choose the desired action in the pulldown list. Right-click on **PROD.QMGRS** and select **Stop Local Queue Managers...**



- __13. Choose **Immediate** for the **Stop Method** (not a good option for a production queue manager!) and click **OK**.

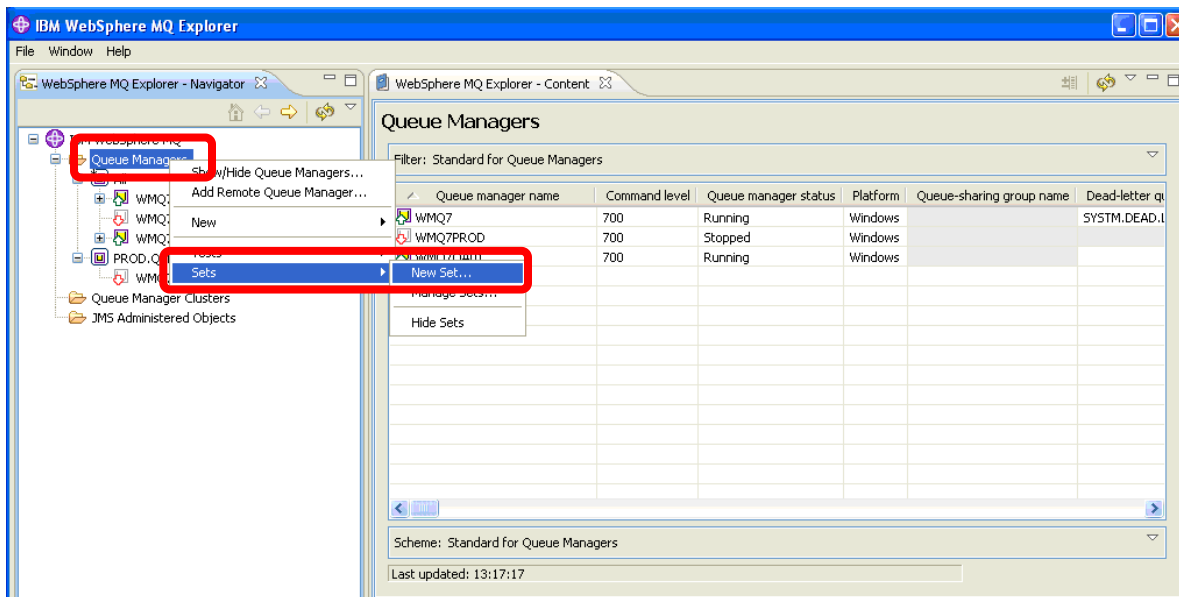


- ___14. After a few seconds the queue manager will show a status of **stopped**. Click the **Close** button to remove the Set Action panel.

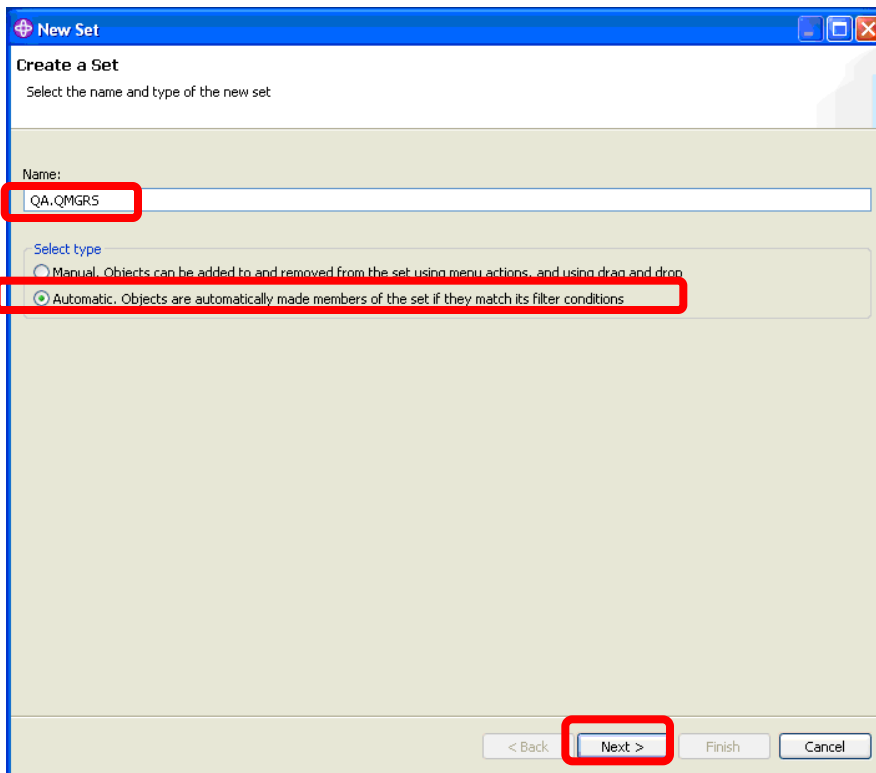


In the previous example, you performed your queue manager grouping in a manual fashion. Grouping can also be done automatically based on a filter you specify. To illustrate this you will create another Set and define it so that it will automatically add the appropriate queue manager(s) to the Set based on a portion of the queue manager name.

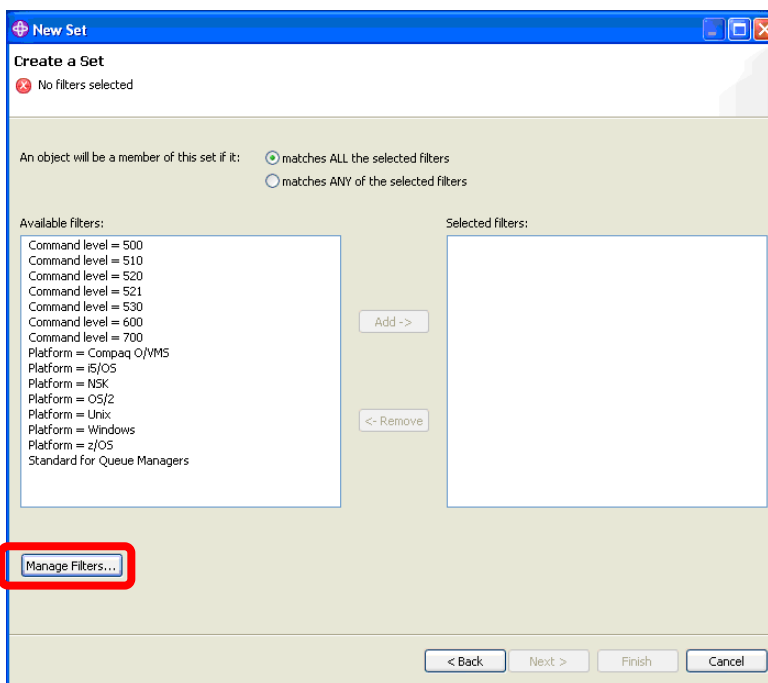
- ___15. Right-click on **Queue Managers** and select **Sets** then **New Set**.



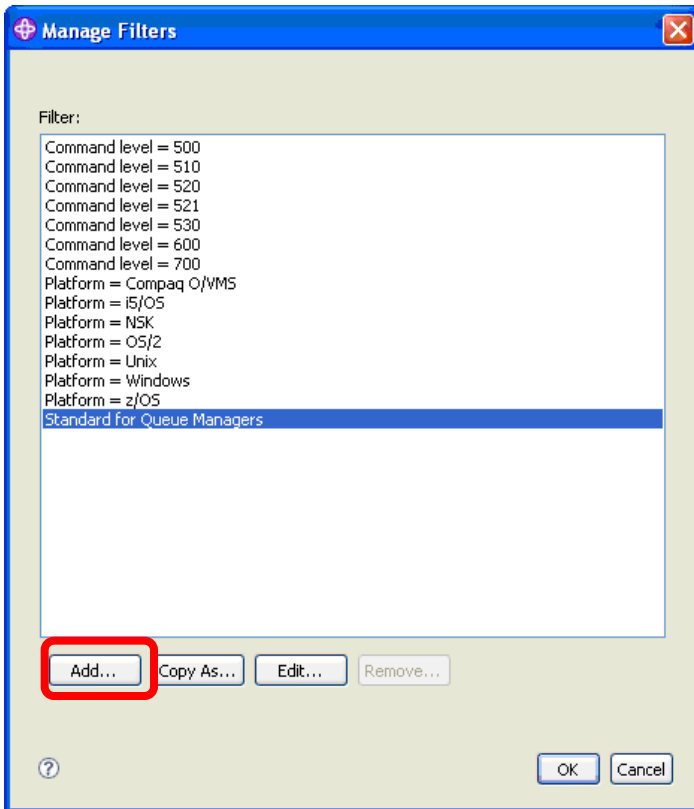
- __16. Name the Set **QA.QMGRS**, select **Automatic** and click the **Next** button.



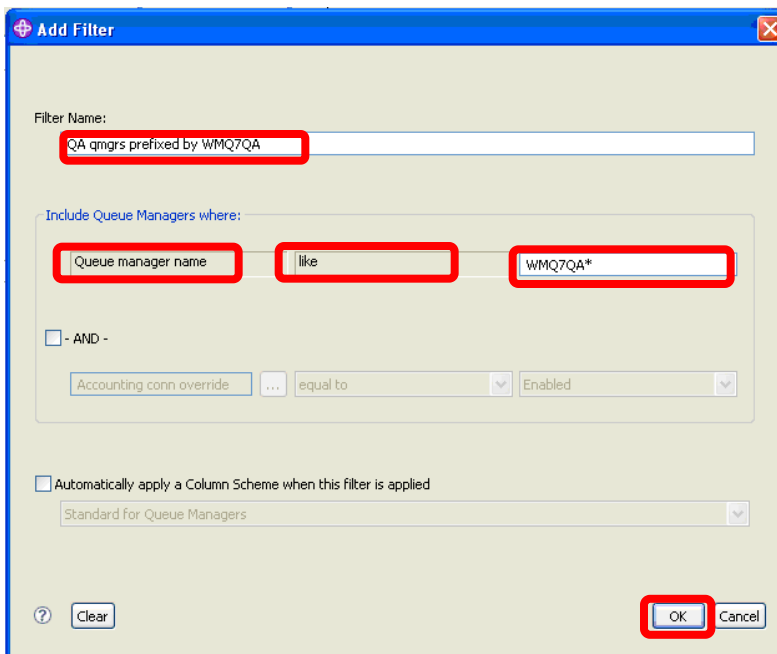
- __17. This panel shows some predefined filters that can be used to choose queue managers for this Set. You can see that you could filter based on WebSphere MQ version and release level or on the operating system platform the queue manager is running on. You are going to create a new filter. Click on the **Manage Filters** button



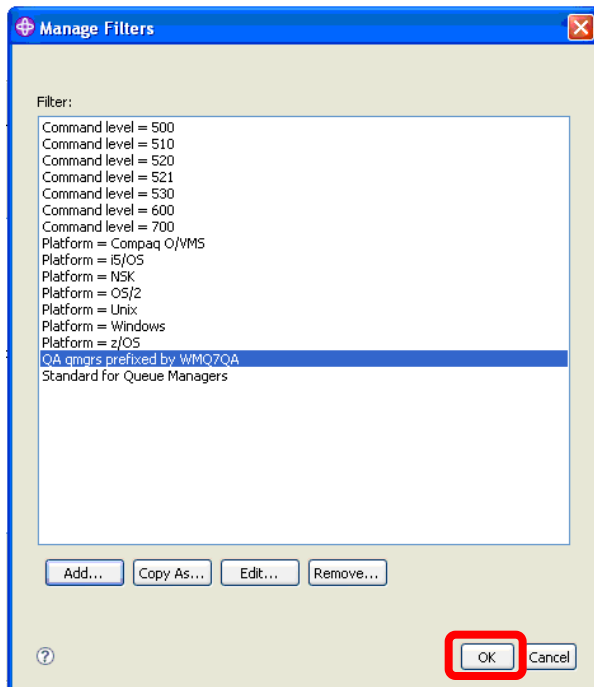
___18. Next, click the **Add** button to indicate that you want to add a new filter.



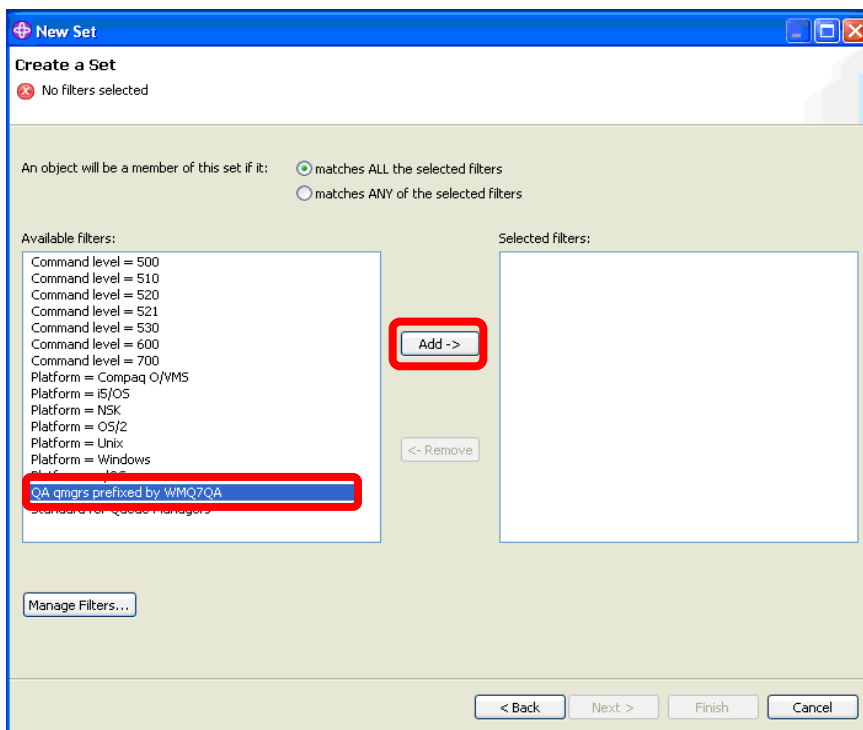
___19. Enter **QA qmgrs prefixed by WMQ7QA** for the **Filter Name**. Enter **WMQ7QA*** for the **Queue manager name like** field. The * is a wild card in this case....any queue manager name that begins with WMQ7QA will satisfy this filter. Click the **OK** button.



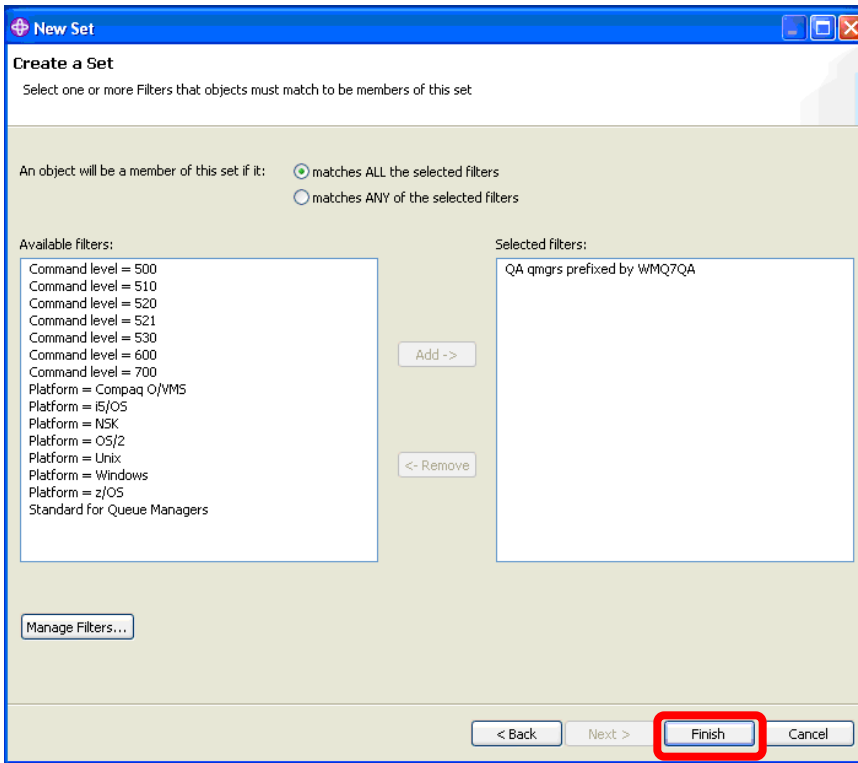
- __20. The new filter now appears in the filter list. Click the **OK** button.



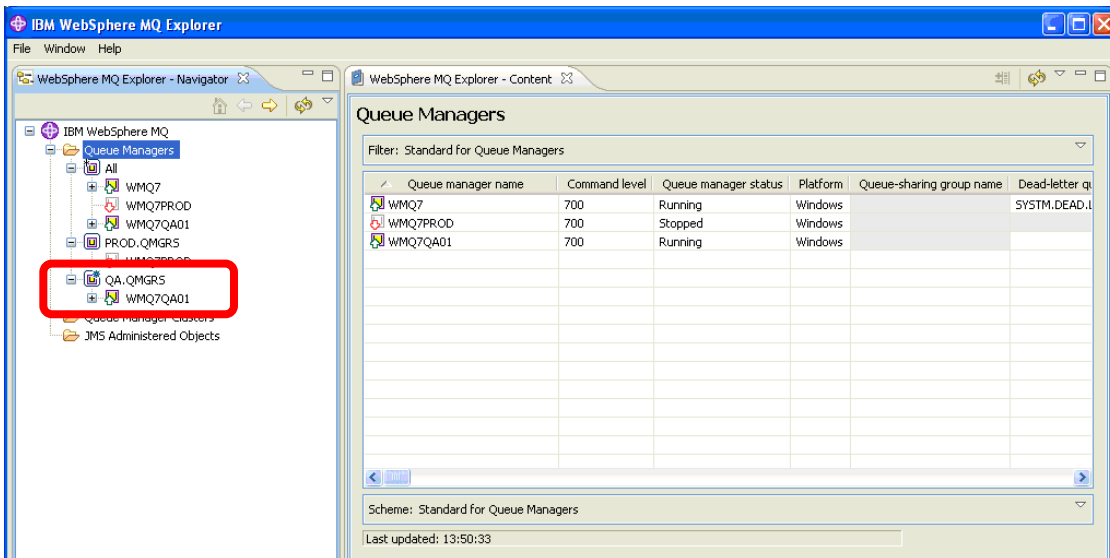
- __21. Click the **QA qmgrs prefixed by WMQ7QA** in the **Available filters** list to select it then click the **Add** button to move it to the **Selected filters** box.



22. Then click **Finish**.



23. In the main MQ Explorer menu you can see that the new Set has been created and queue manager **WMQ7QA01** has been automatically added to the **QA.QMRGRS** group as it satisfied the filter you specified.

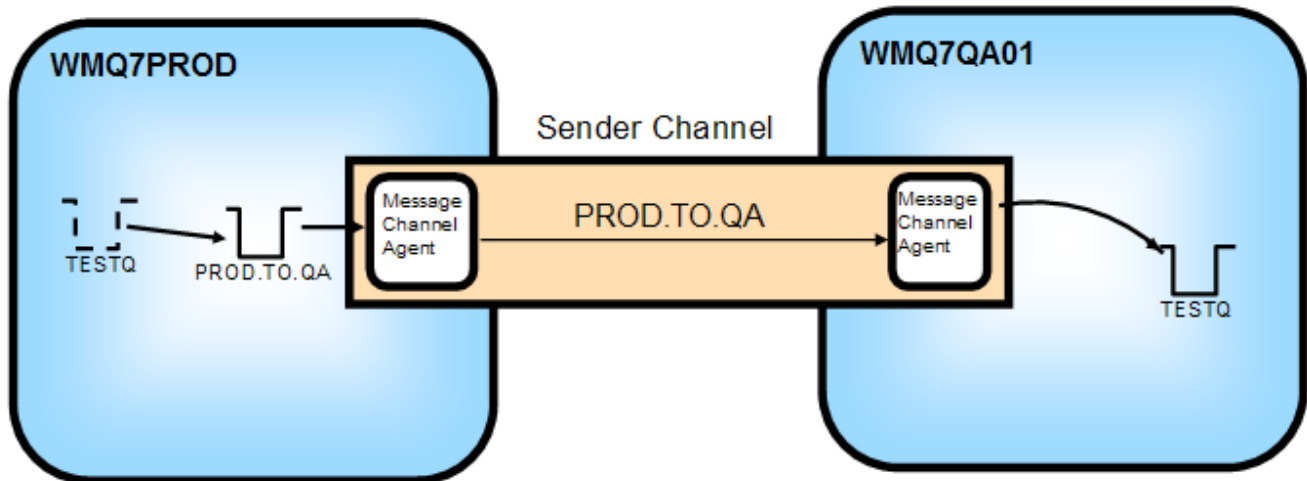


If you created a new queue manager whose name satisfied your filter it would also get automatically added to the QA.QMRGRS Set.

This is the end of this portion of Lab 1

1.4 Distributed queuing

Your next step will be to explore distributed queuing. Since the primary reason to have an enterprise messaging product like WebSphere MQ is to move messages reliably between systems, this lab will give you a simple example of the steps involved in connecting two queue managers and moving messages between them. Here is a diagram of the configuration you will create:



Here are the MQ objects you will be creating and working with:

WMQ7PROD:

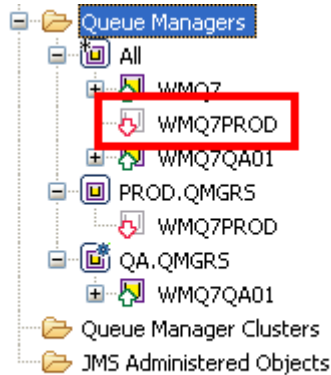
- A Remote queue definition called **TESTQ***
- A Transmission queue called **PROD.TO.QA***
- A Sender Channel called **PROD.TO.QA***

WMQ7QA01:

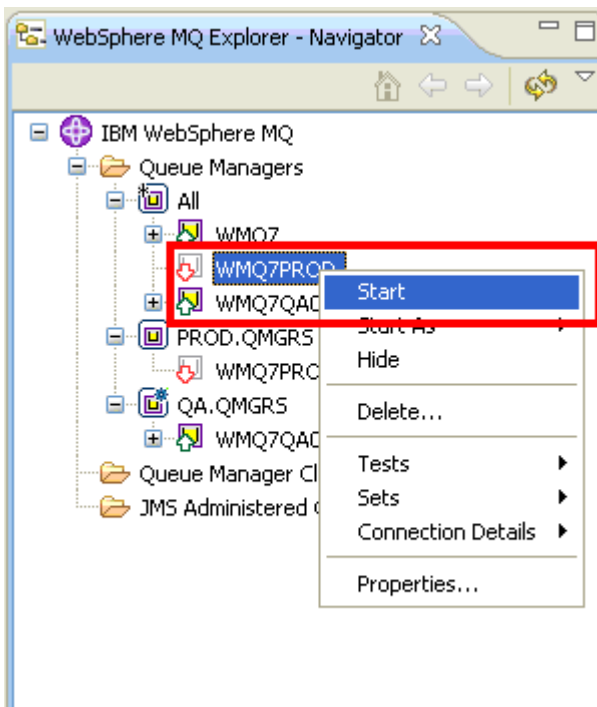
- A Local queue called **TESTQ***
- A Receiver Channel called **PROD.TO.QA***

You will then use one of the sample programs supplied with WebSphere MQ (called amqsput) to send messages from the WMQ7PROD queue manager, and another sample program called amqsget to read those messages off the destination queue on WMQ7QA01.

- ___1. In a previous step you stopped the WMQ7PROD queue manager. You want to restart it now. In the navigator pane, you should see the icon representing the WMQ7PROD queue manager with a **red** arrow pointing down, as in the picture below; this indicates that the queue manager is stopped (a **green** arrow pointing up indicates that the queue manager is running).



- ___2. Right-click on the WMQ7PROD queue manager, and select **Start**, as shown below.

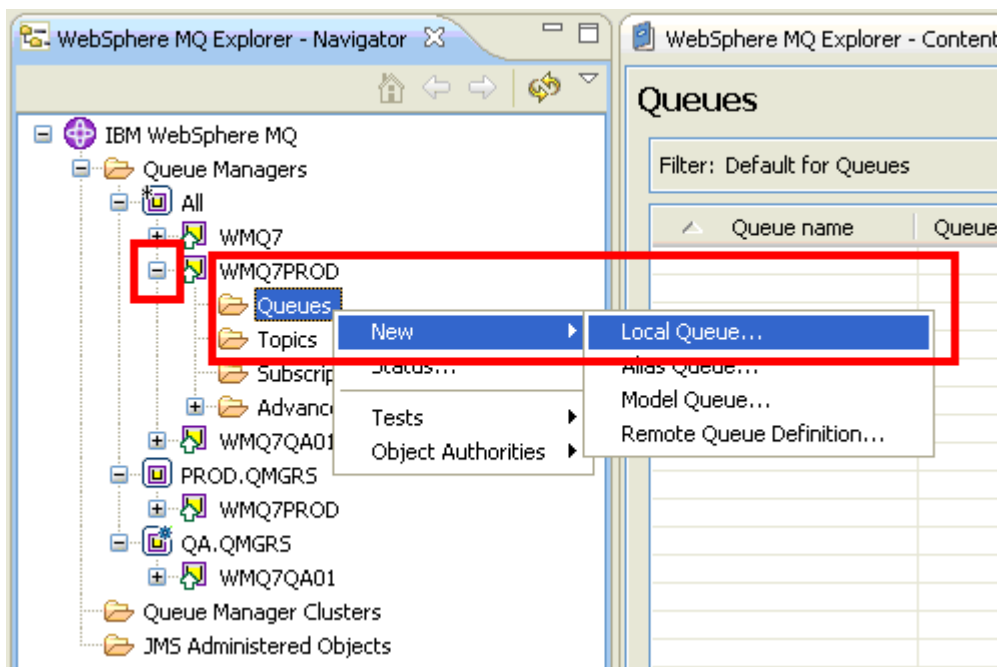


You will see a pop-up like the one below for several seconds while the queue manager starts.



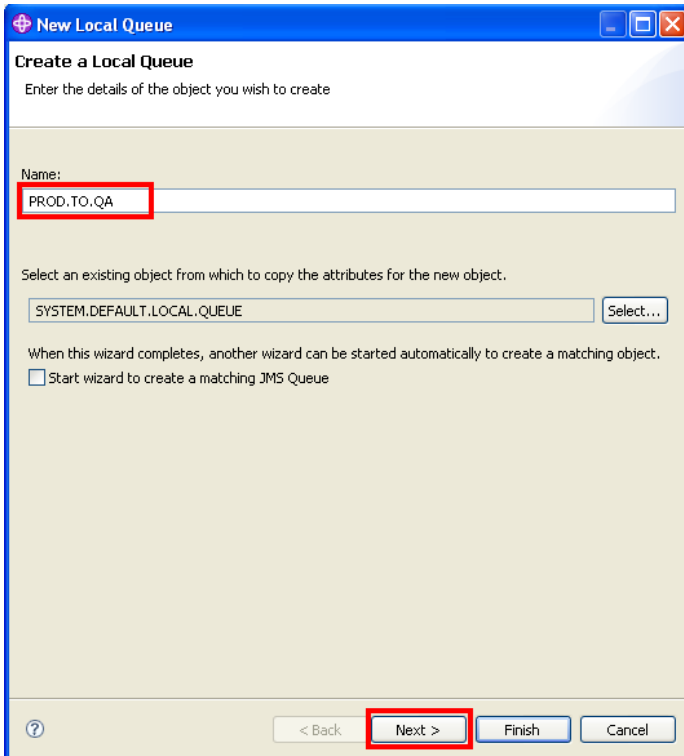
With WMQ7PROD running, you will now go through the steps needed to connect it to WMQ7QA01.

- __3. First you will create a *Transmission queue*. Click on the plus sign next to WMQ7PROD to expand the tree, and then right-click on the **Queues** folder, select **New** and then **Local Queue...**

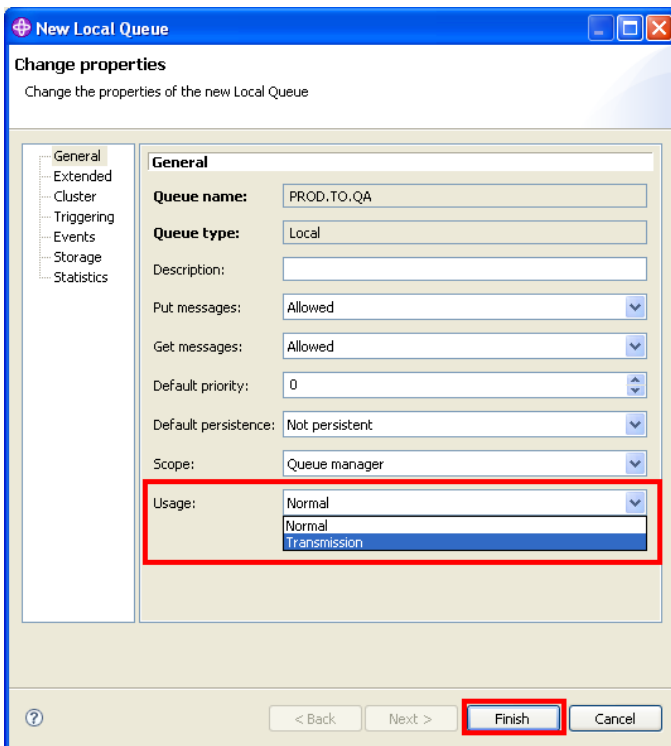


A Transmission queue is a special type of local queue, associated with a Sender Channel that holds messages safely until they are transferred to the queue manager on the other end of the channel. A typical nomenclature used for naming transmission queues and channels is `<sourceqmq>.TO.<targetqmq>`. You will use an abbreviated form for purposes of this lab, calling the transmission queue **PROD.TO.QA**.

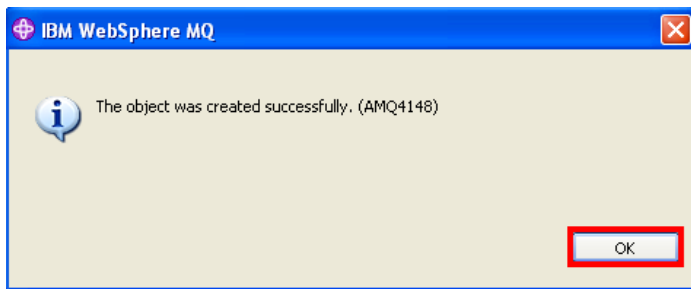
__4. Enter the name **PROD.TO.QA** as shown below, and click **Next**



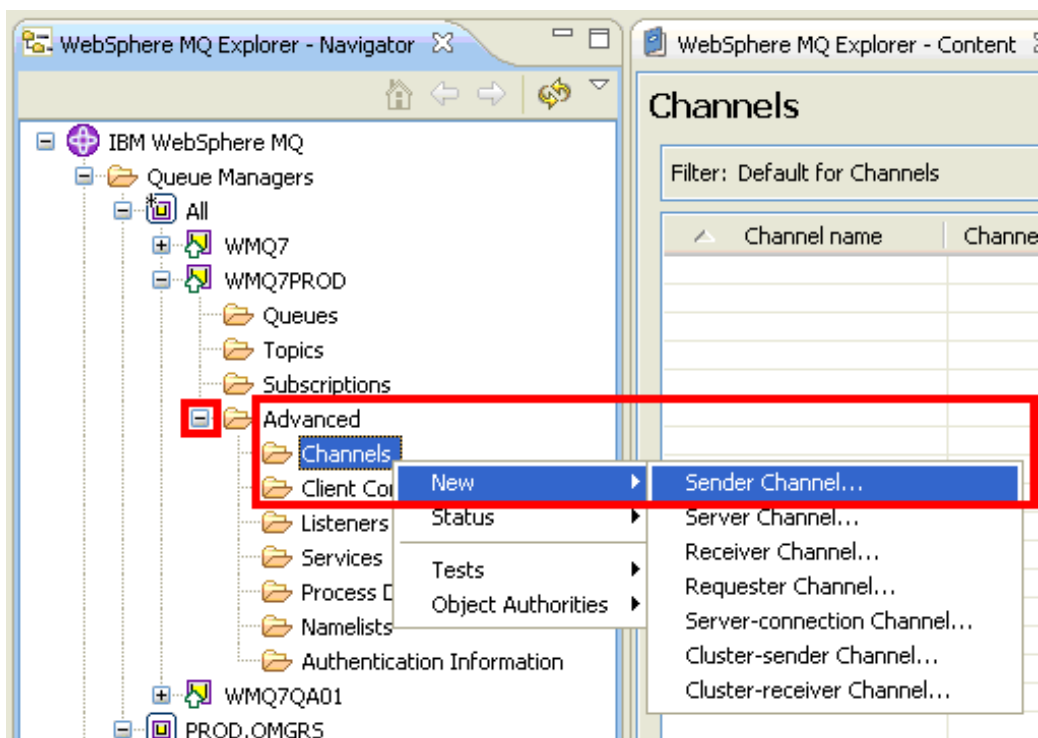
__5. Since this queue will be used as a transmission queue, indicate that by using the dropdown labeled **Usage**, and click **Finish**



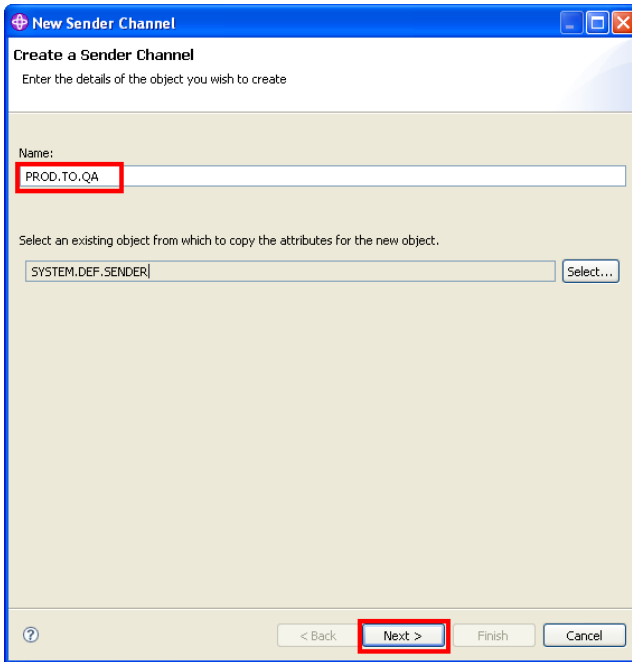
- __6. Click OK on the confirmation popup



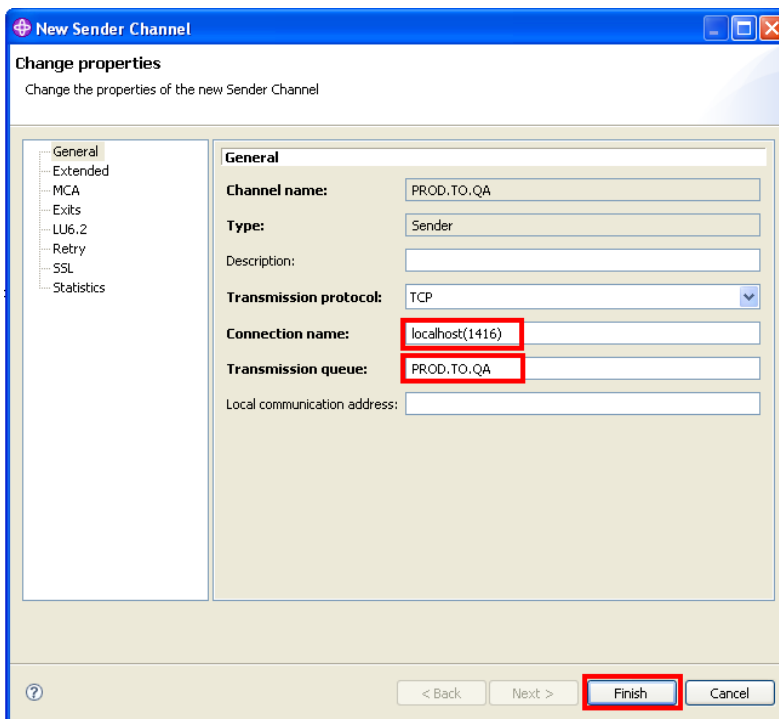
- __7. Next you will create the Sender Channel this transmission queue will be associated with. Click on the plus sign next to the **Advanced** folder under WMQ7PROD to expand the tree, and then right-click on the **Channels** folder, select **New** and then **Sender Channel**



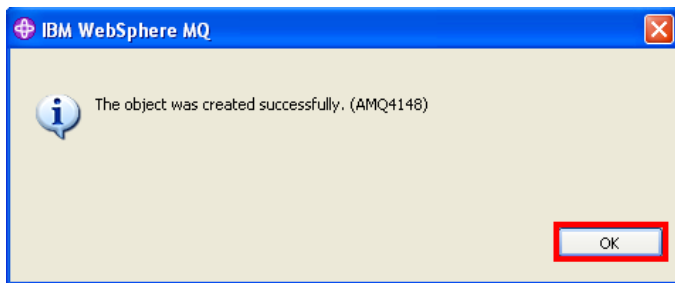
__8. Enter the name **PROD.TO.QA** as shown below, and click **Next**.



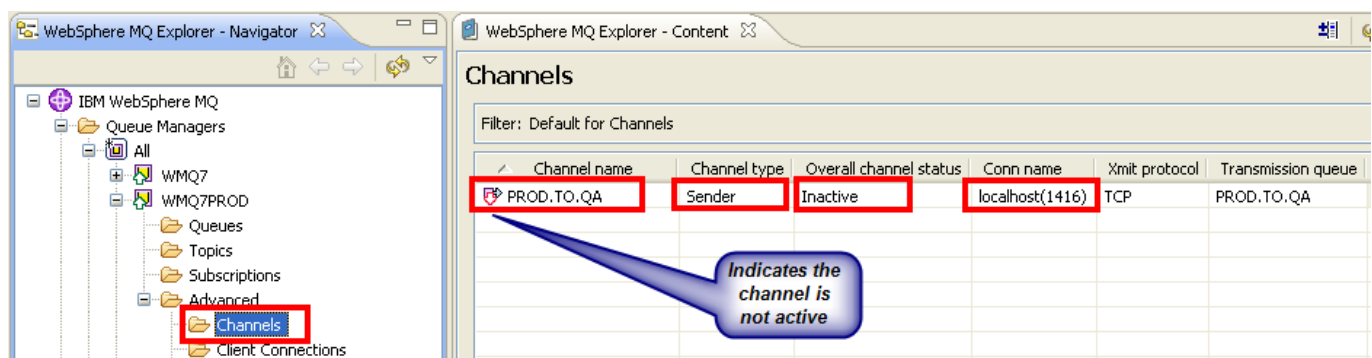
__9. You now need to identify the network location of the queue manager this channel is to connect to. Remember when you created the WMQ7QA01 queue manager, you assigned it a listener port of 1416. Specify that now, using **localhost** as the **Connection name**, as shown below. Also enter the name of the **Transmission queue** you created earlier, **PROD.TO.QA**. Then click **Finish**.



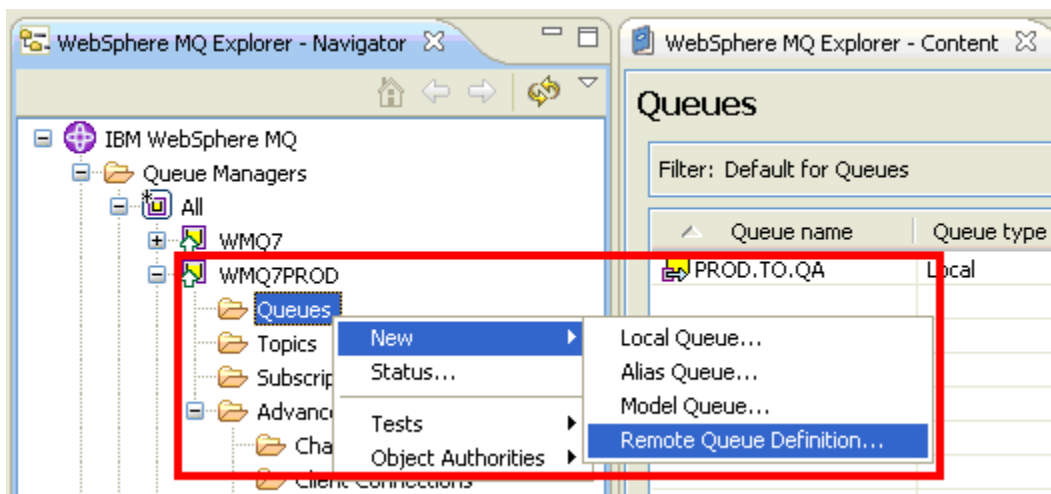
- ___10. Click OK on the confirmation popup. The channel has been created.



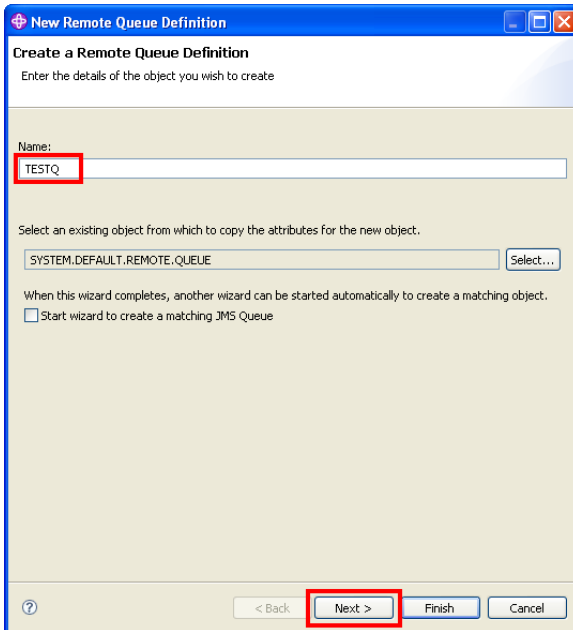
- ___11. In the Content pane on the right-hand side, you should see the details of the channel you just created. Notice that, in addition to the name, you see listed the **Channel type** of *Sender*, the **Overall channel status** as *Inactive*, and a **Conn name** of *localhost(1416)*



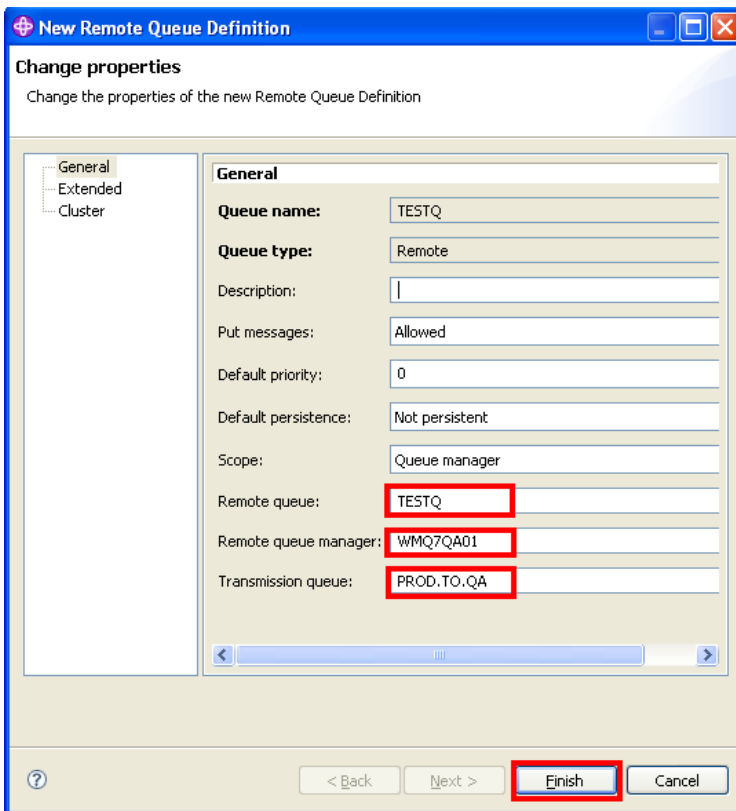
- ___12. The last task remaining on WMQ7PROD is to create a Remote queue definition. This definition will act as an alias of the real target queue that you will be creating shortly on WMQ7QA01. To create the remote queue definition, right-click on the **Queues** folder under WMQ7PROD, select **New** and then **Remote Queue Definition...**



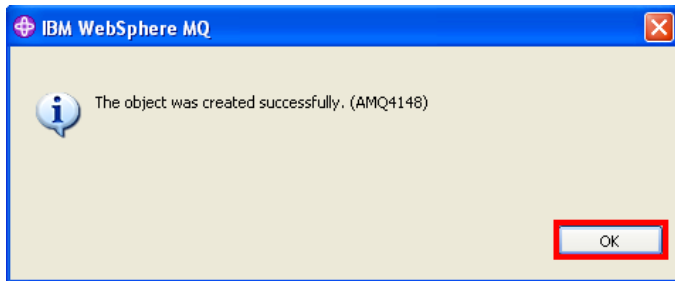
___13. Name the remote queue definition **TESTQ**, and click **Next**



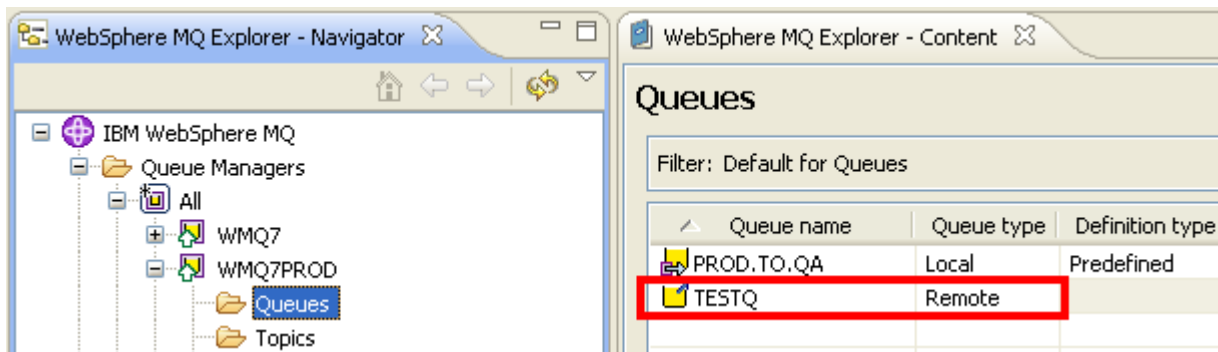
___14. Identify the name of the **Remote queue** (i.e. the name of the actual target queue) as **TESTQ**. Specify the **Queue manager name** where the target queue resides (**WMQ7QA01**), and indicate the **Transmission queue** to be used to reach that queue manager (**PROD.TO.QA**). When those have been entered, click **Finish**.



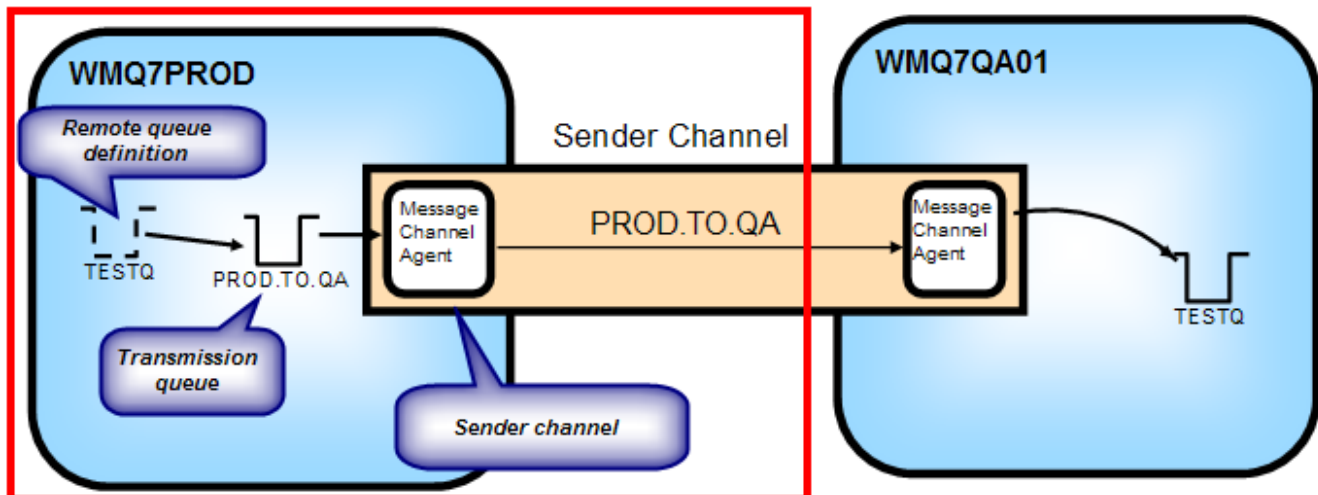
- ___15. Click OK on the confirmation popup. The remote queue definition has been created.



- ___16. You should now see your new queue definition under WMQ7PROD, with a **Queue type** of **Remote**



Let's review what you have configured so far. Look at the left-hand side of the diagram below:

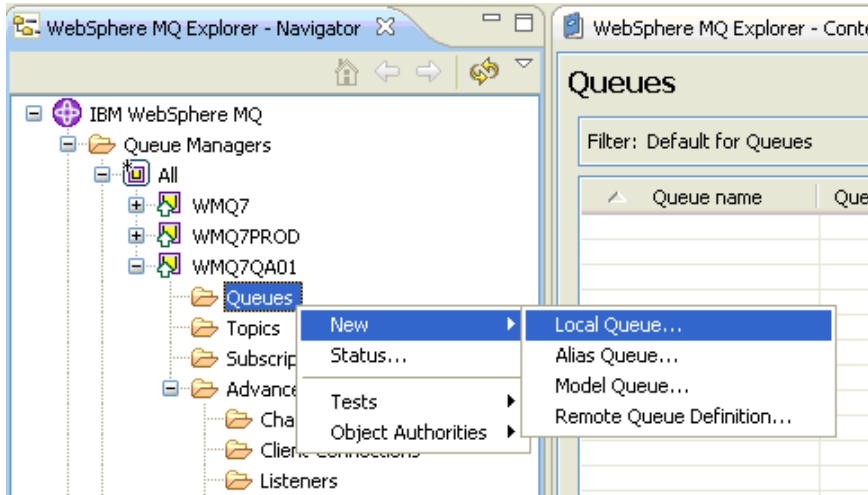


On WMQ7PROD you created a Remote queue definition of TESTQ, which points to a Transmission queue called PROD.TO.QA, which is associated with a Sender channel also called PROD.TO.QA that will connect to WMQ7QA01.

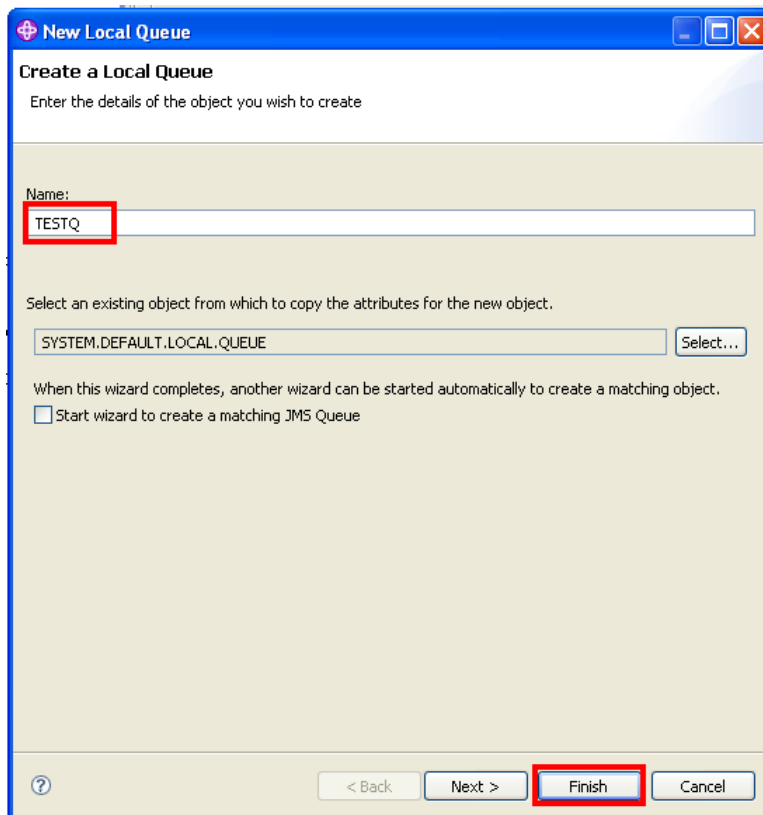
With the Sender side setup complete, you now need to create the corresponding definitions on WMQ7QA01.

___17. You previously created a Remote queue definition called TESTQ. As mentioned then, this definition is an alias for the actual local queue you are about to create.

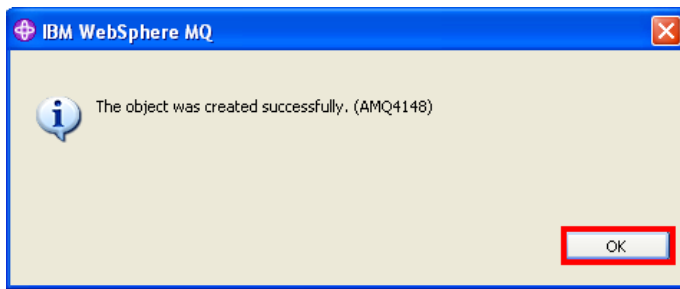
Expand the folder tree for queue manager WMQ7QA01, right-click on the **Queues** folder, click **New**, then **Local Queue...**



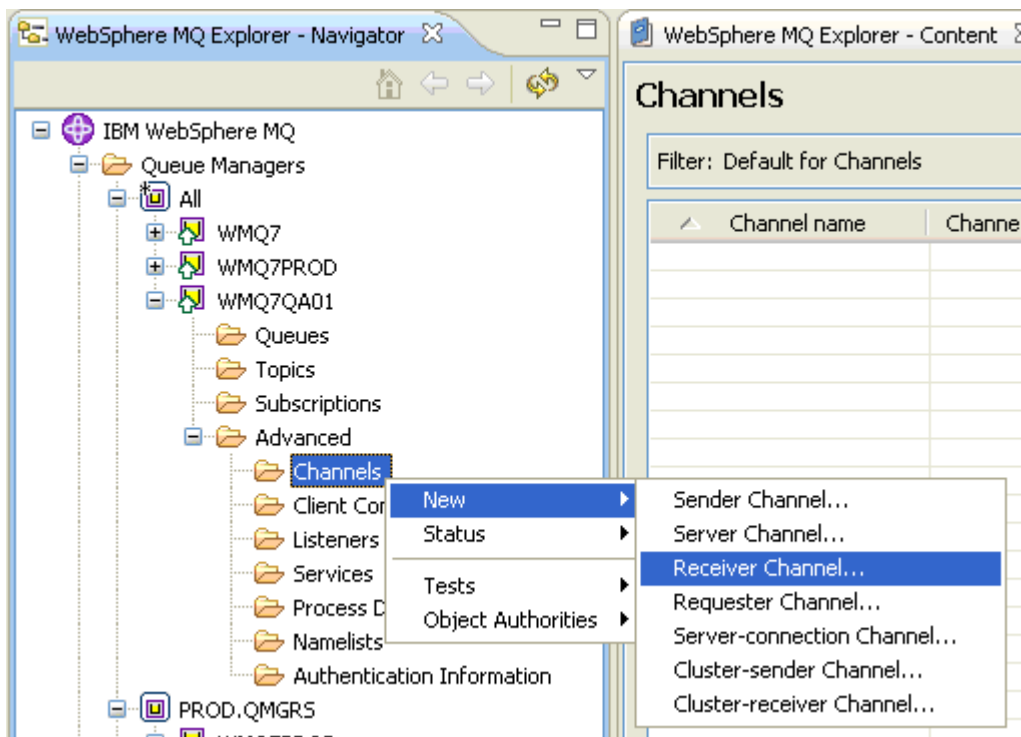
___18. Enter **TESTQ** as the Name. The default values are acceptable for this queue, so click **Finish**.



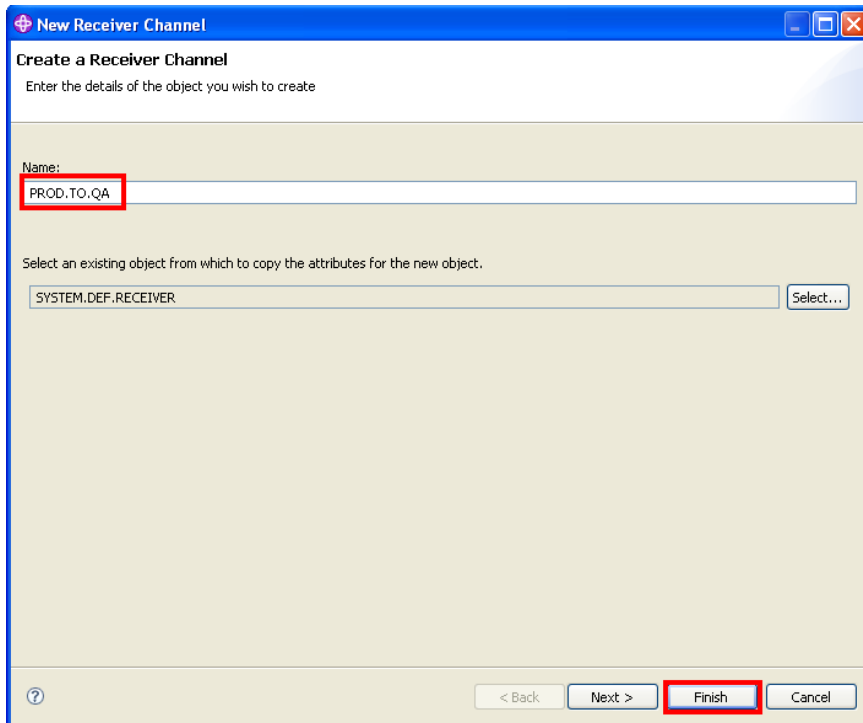
- __19. Click OK on the confirmation popup.



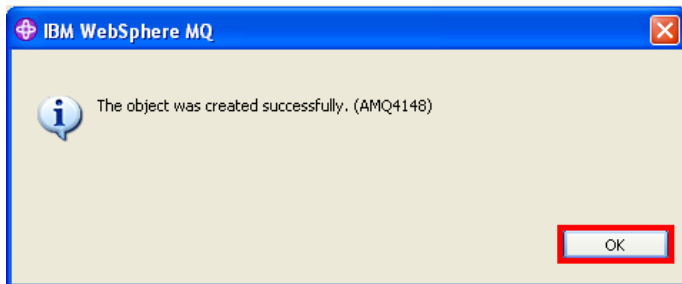
- __20. Next you will create the Receiver Channel. This channel must have the same name as the corresponding Sender channel you created earlier. Expand the **Advanced** folder tree under WMQ7QA01, right-click on the **Channels** folder, then select **New**, then **Receiver Channel...**



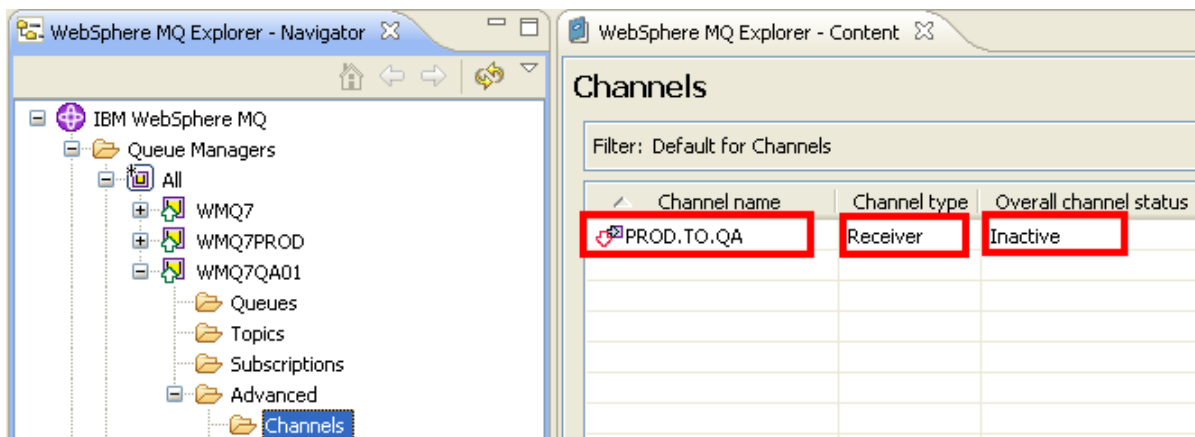
- __21. Enter the name **PROD.TO.QA** as shown below. No other properties are required for a Receiver channel, so click **Finish**.



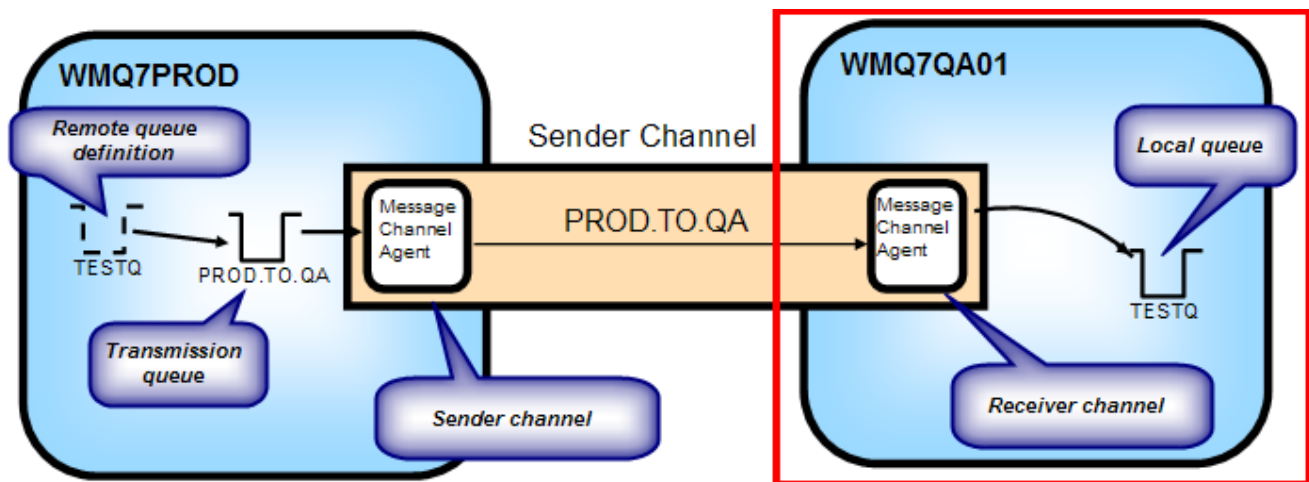
- __22. Click OK on the confirmation popup. The channel has been created.



23. In the Content pane on the right-hand side, you should see the details of the channel you just created. You should see the **Name**, *PROD.TO.QA*, with a **Channel type** of *Receiver* and an **Overall channel status** of *Inactive*.



Your configuration should now be complete. Let's review what you have configured, looking at the right-hand side of the diagram below:



On WMQ7QA01 you created a Local queue called **TESTQ**, which will be the target queue for your test. You also created a **Receiver channel** as a partner for the Sender channel you created in WMQ7PROD, called **PROD.TO.QA**.

With the Receiver side setup complete, you are now ready to test your configuration and see how WebSphere MQ moves messages.

__24. Launch a command window by clicking the icon found on the taskbar



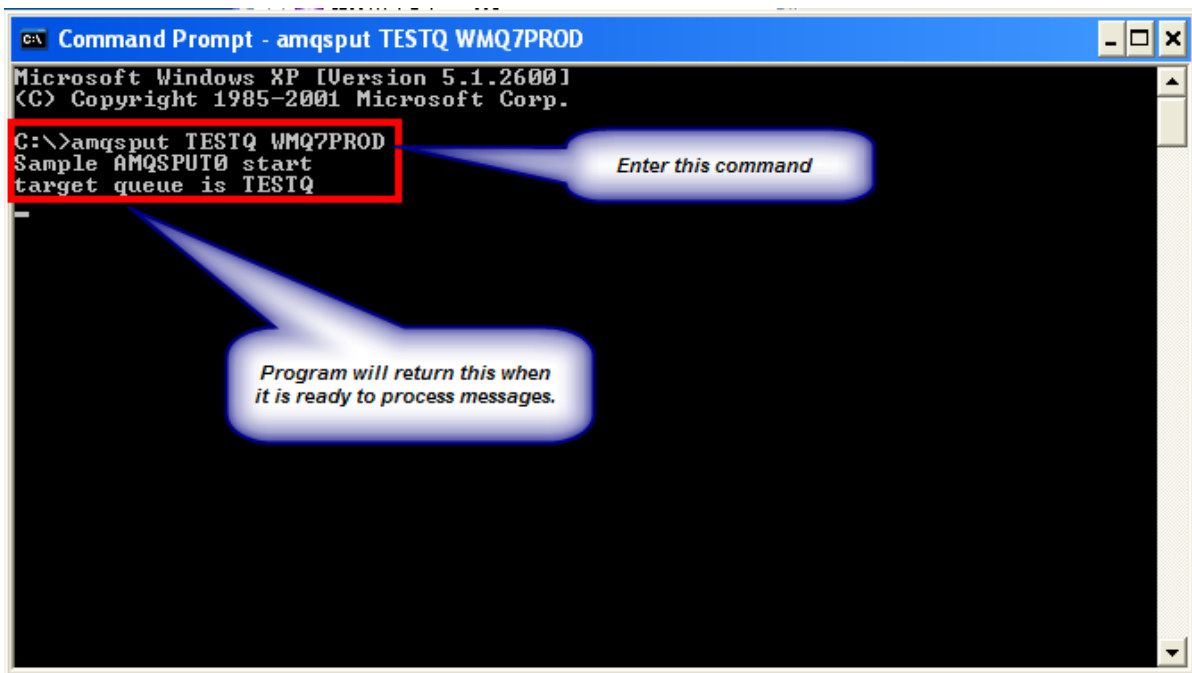
__25. For testing the putting (or sending) side, you will use a sample program supplied with WebSphere MQ called amqsput. The format of this command is:

amqsput <q_name> <qmgr_name>

For your test, you will be putting messages to the TESTQ remote queue definition on queue manager WMQ7PROD, so enter the following command:

amqsput TESTQ WMQ7PROD

You should see the following when you enter this in the command window:



The amqsput sample program will take each line of text that you enter, and put each line as a separate message to the named queue.

__26. Enter three lines of text, as shown below:

```

Microsoft Windows XP [Version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

C:\>amqspuT TESTQ WMQ7PROD
Sample AMQSPUT0 start
target queue is TESTQ
test msg1
test msg2
test msg3

```

Each line will be put on the target queue as a separate message.

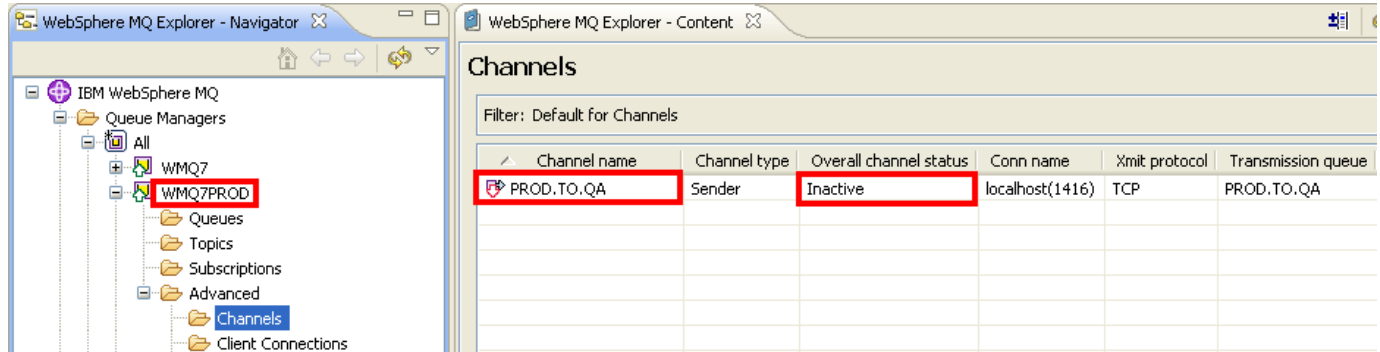
__27. Return to the MQ Explorer. In the Navigator pane, if necessary, expand the tree under WMQ7PROD by clicking the plus sign, and select the **Queues** folder. You should see the following:

Queue name	Queue type	Definition type	Open input count	Open output count	Current queue depth
PROD.TO.QA	Local	Predefined	0	1	3
TESTQ	Remote		0	0	

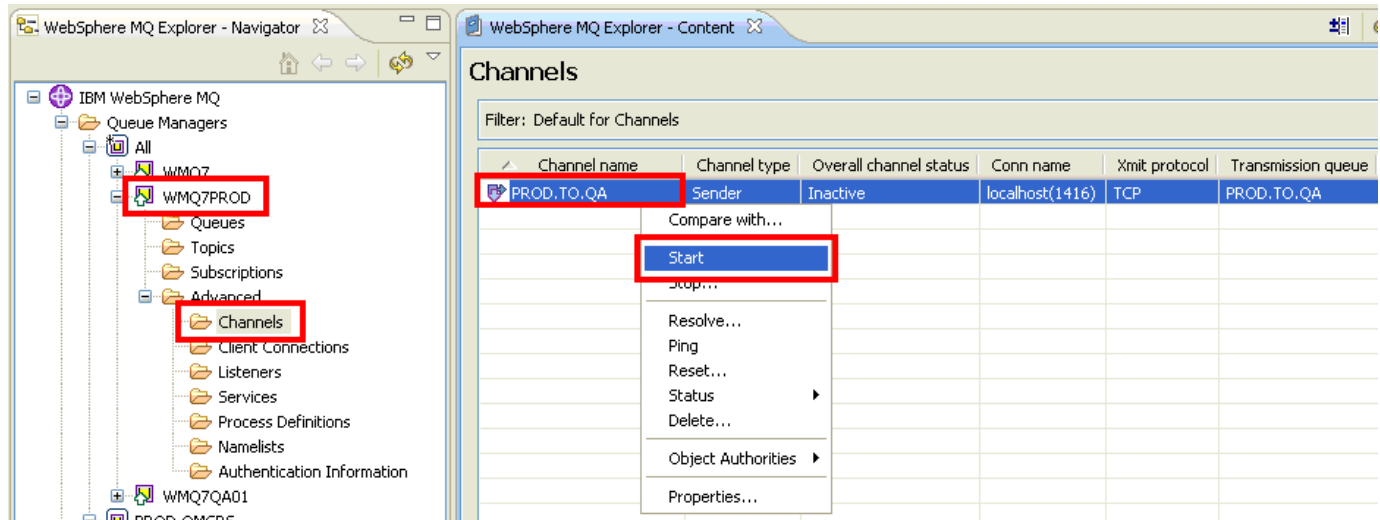
Look at the **Current queue depth** column and consider what you see here. Notice that TESTQ has no messages, and in fact this field is grayed out. Why???

Remember that a remote queue definition is like an [alias](#); it is not an actual queue, but is instead a *reference* to one. In this case, it is a reference to the *transmission queue* associated with the channel that points to the queue manager where the actual TESTQ instance resides. If you look at the current queue depth of transmission queue **PROD.TO.QA** you see that it contains three messages. But why are they sitting here???. Why haven't they been transferred to the WMQ7QA01 queue manager?

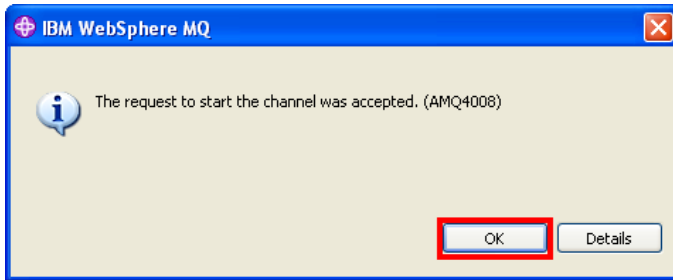
- __28. To find out, click on the **Channels** folder on queue manager WMQ7PROD, and look at the **Overall channel status** for Channel **PROD.TO.QA**. Notice that the channel is **Inactive**! In order to move messages, the channel must have a status of Running. It is possible to configure the channel such that it will start *automatically* when a message arrives on its transmission queue, but we did not do that for purposes of this lab. Instead, you will start the channel manually.



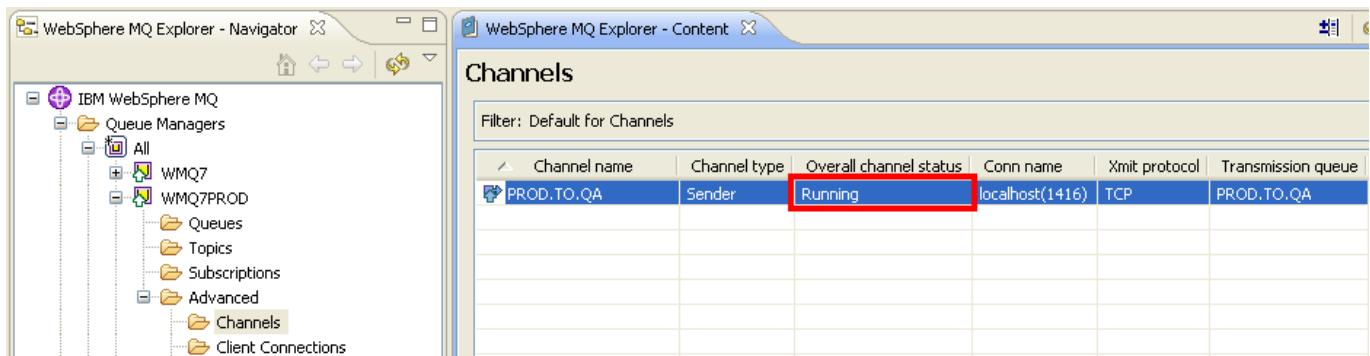
- __29. Right-click on the **PROD.TO.QA** Sender channel entry, and click **Start**



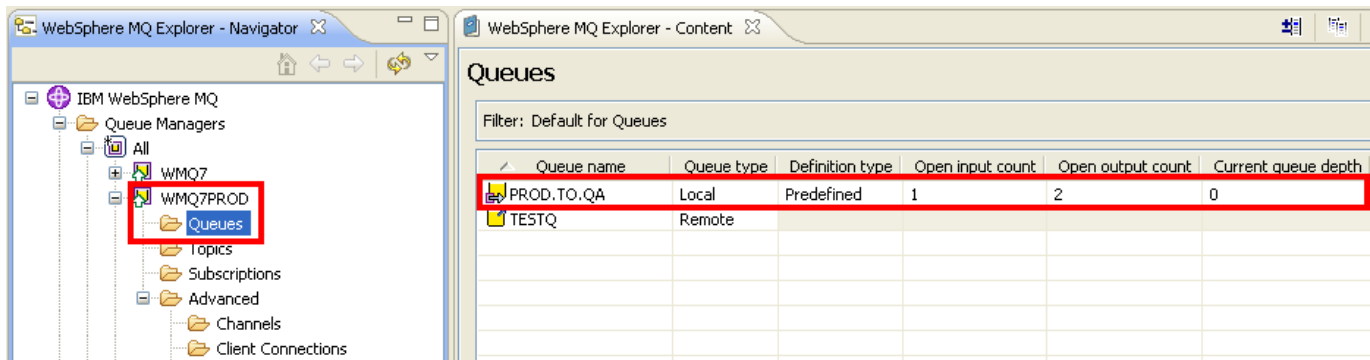
__30. Click **OK** on the confirmation popup.



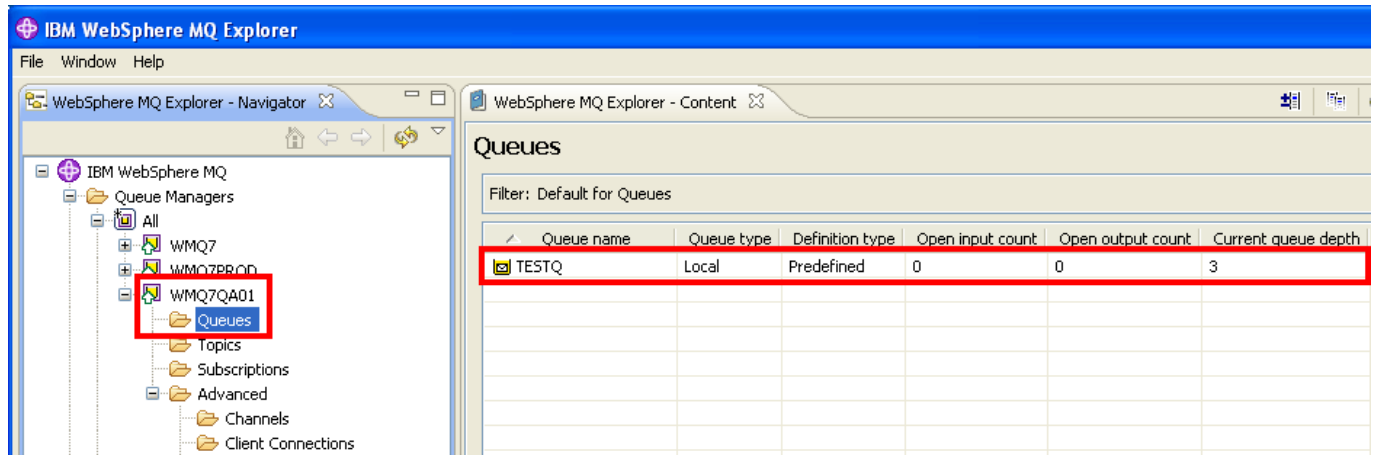
__31. Returning to the Overall channel status, you may see it go through several states (Initializing, Binding, etc) over a few seconds until it reaches **Running** status, as shown below



__32. With the channel now running, click on the **Queues** folder again for queue manager WMQ7PROD, and look at the **Current queue depth** for transmission queue **PROD.TO.QA**. Notice that now the queue is empty.



- ___33. In the MQ Explorer navigator pane, expand the tree below the **WMQ7QA01** queue manager, click on the **Queues** folder, and look at the **Current queue depth** for Queue **TESTQ**. Notice the queue depth of the target queue is three, indicating that your messages have been moved here from the WMQ7PROD queue manager.



- ___34. To confirm that these are actually the messages that you put using amqsput, you will use another sample program supplied with WebSphere MQ, called amqsget. Open another command window by clicking the icon found on the taskbar



- ___35. The amqsget sample program will open the specified queue and wait for messages. The format of this command is:

amqsget <q_name> <qmgr_name>

For your test, you will be getting the messages from the TESTQ local queue definition on queue manager WMQ7QA01, so enter the following command:

amqsget TESTQ WMQ7QA01

You should see the following when you enter this in the command window:



```

C:\>amqsget TESTQ WMQ7QA01
Sample AMQSGET0 start
message <test msg1>
message <test msg2>
message <test msg3>

```

Here are the three test messages

If you wish, you can go back to the amqsput window and enter some more lines of text. If you then return to the amqsget window, you will see that the messages are delivered immediately (or nearly so), because the channel is now running.

Note: The amqsget program is designed to end after 15 seconds if no messages arrive on the queue; consequently, it's possible that the program will end before you can send your messages. If this happened, just restart the program.

When your testing is complete, terminate amqsput by just hitting the Enter key with no input text. Let amqsget terminate by waiting for the 15 second timeout to expire. Then close the two command windows.

This is the end of this portion of Lab 1

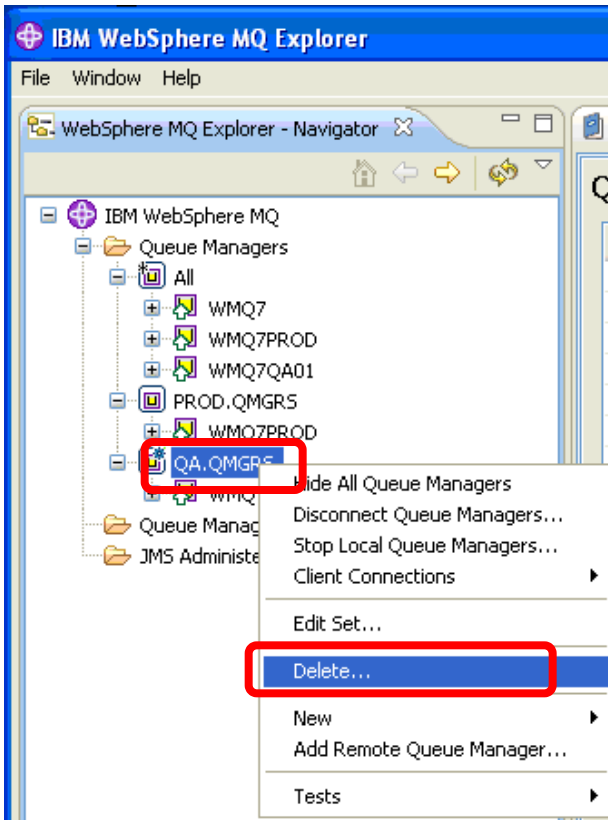
1.5 Lab Cleanup

Since you will make no further use in future labs of the last two queue managers you created, you are going to delete them.

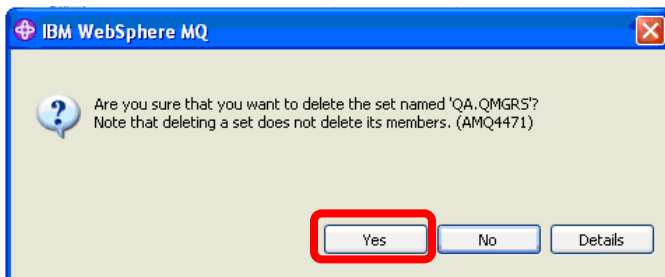
First you will remove the two Sets you created as they contain only these queue managers and would be empty after the queue managers are deleted.

Then you will stop the queue managers as you cannot delete one that is running. And then you will delete them.

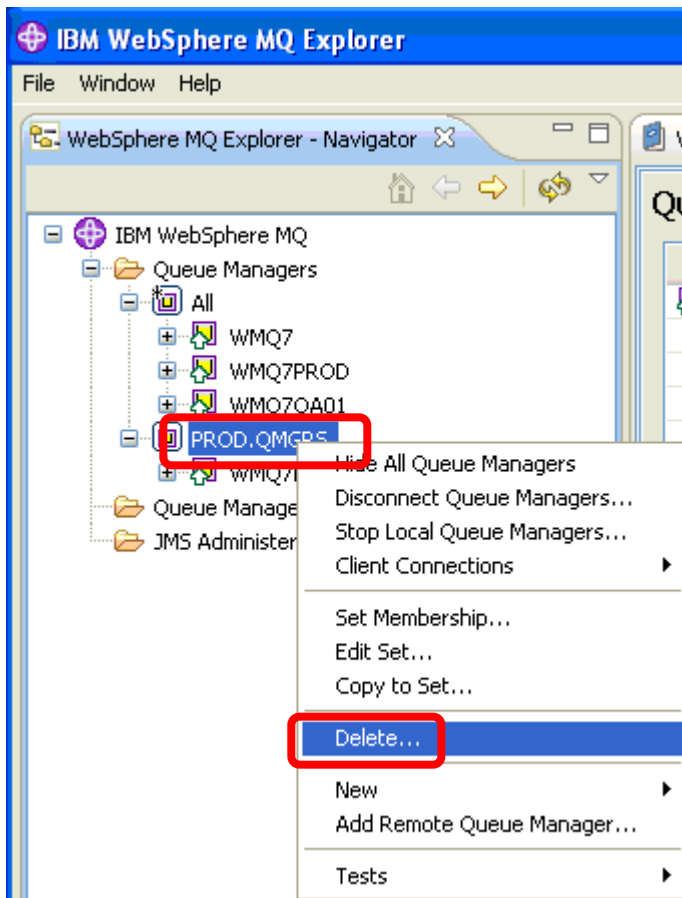
- ___1. Bring the MQ Explorer into view, **right-click** on the **QA.QMGRS** Set and select **Delete**.



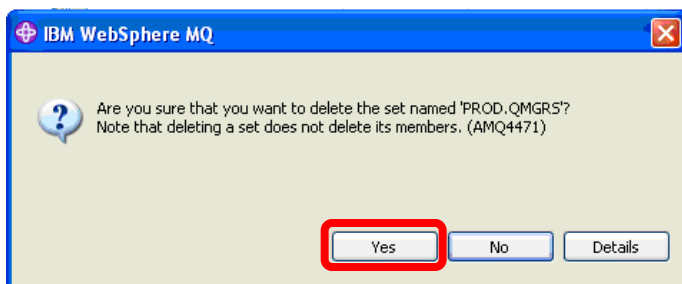
- ___2. Confirm the delete by clicking on the **Yes** button...



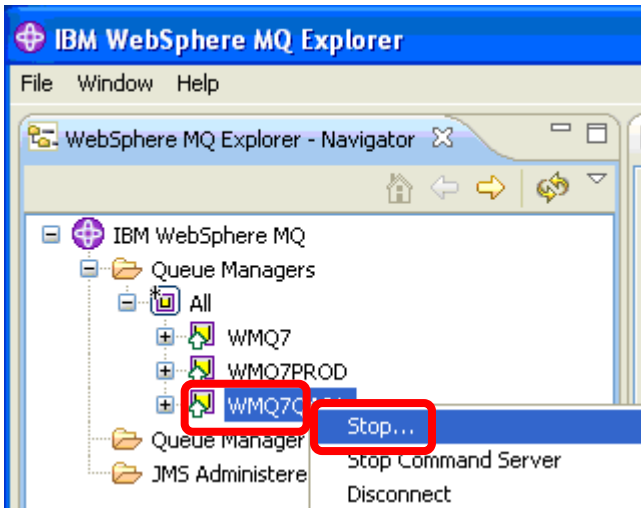
- ___3. Right-click on the **PRODS.QMGRS** Set and select **Delete**.



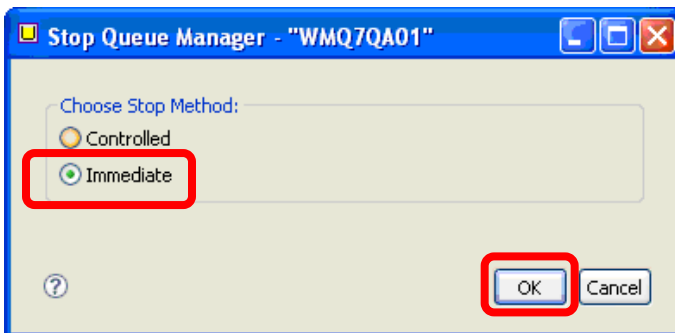
- ___4. Confirm the delete by clicking on the **Yes** button...



- __5. **Right-click** on the **WMQ7QA01** queue manager and select **Stop...**



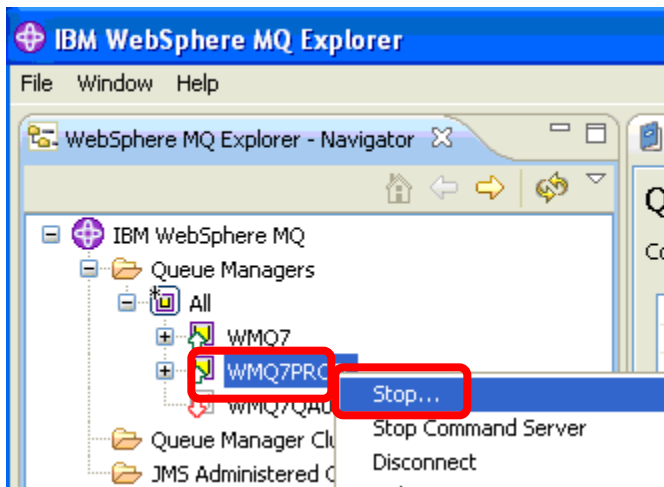
- __6. Select the **Immediate** radio button and click **OK**.



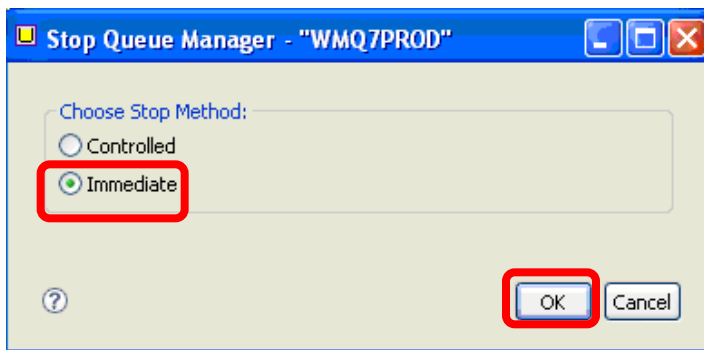
- __7. The following progress panel will appear...



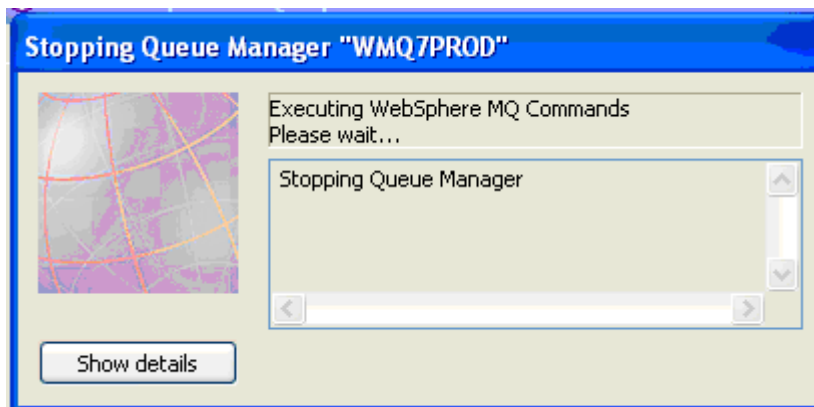
- __8. **Right-click** on the **WMQ7PROD** queue manager and select **Stop...**



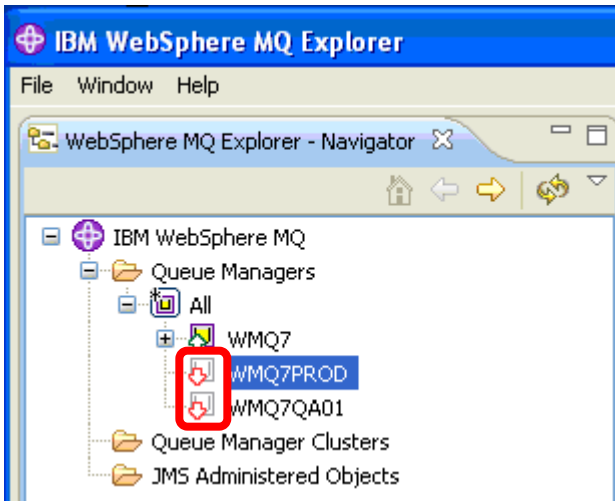
- __9. Select the **Immediate** radio button and click **OK**.



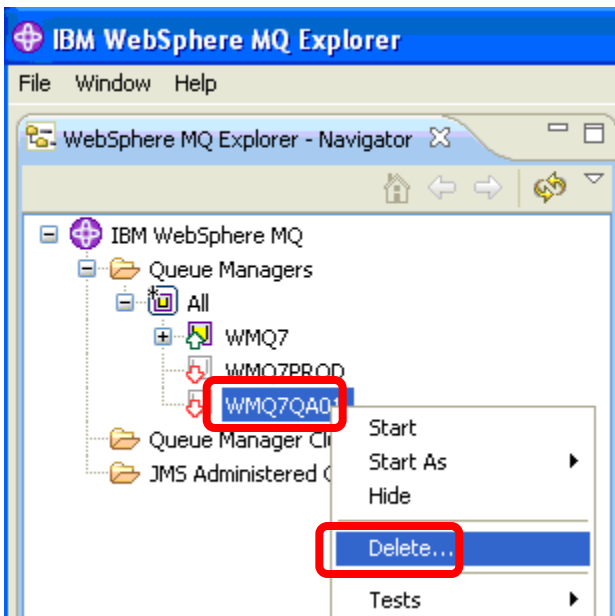
- __10. The following progress panel will appear...



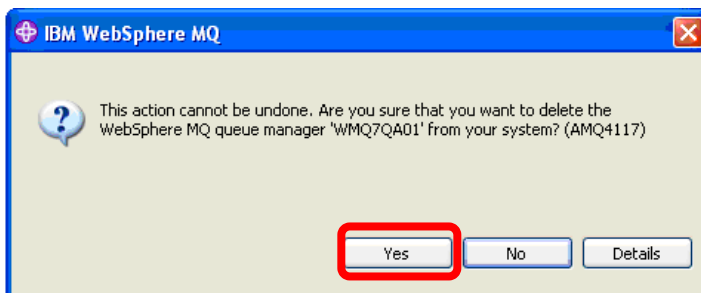
__11. Now both queue managers should be stopped.



__12. Right-click on the **WMQ7QA01** queue manager and select **Delete....**



__13. Confirm the delete by clicking the **Yes** button....



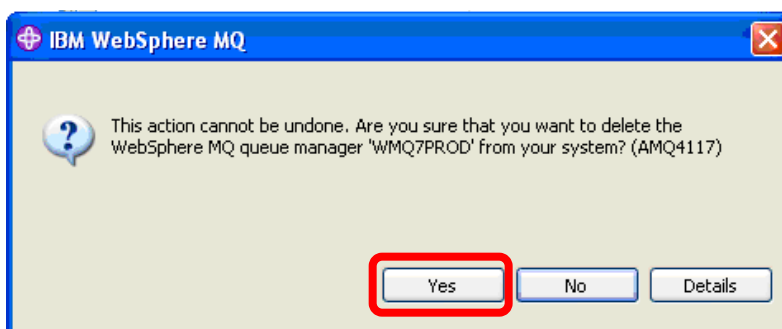
___14. The following progress panel will be displayed....



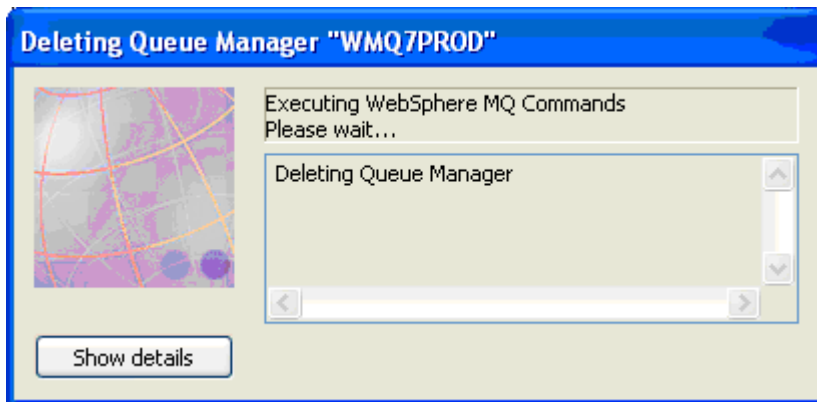
___15. Right-click on the **WMQ7PROD** queue manager and select **Delete**....



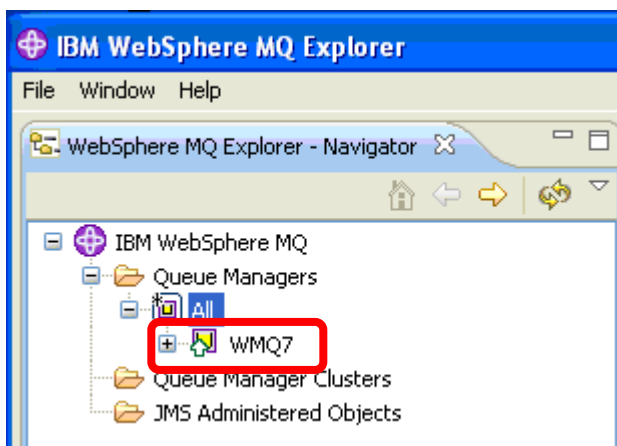
___16. Confirm the delete by clicking the **Yes** button....



___17. The following progress panel will be displayed....



___18. You should now have just a single queue manager, WMQ7



This concludes Lab 1.

Lab 2 Configuring the WebSphere MQ JMS Provider

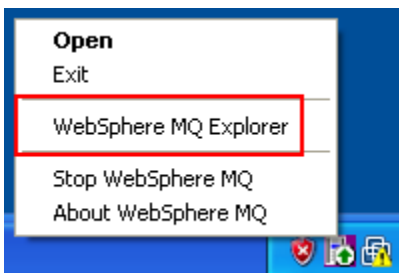
The purpose of this lab is to demonstrate some of the typical steps you will go through when configuring WebSphere MQ as a JMS provider. Some of the tasks that you will perform include:

- Configure the Administered objects that a JMS program typically requires. Administered objects are used to externalize **Connection Factories** and **Destinations** from the program. This allows JMS Applications to be portable between Messaging Providers by shielding the applications from provider-specific details.
- Define a **Java Naming and Directory Interface (JNDI) directory**. In this lab you will perform the following:
 - Create a JNDI directory
 - Populate it with Connection Factory and Destination definitions
 - Use the MQ Explorer wizard to create corresponding MQ definitions
 - Run one of the Java™ Message Service (JMS) sample programs from the command line to use those definitions to connect to MQ as a JMS provider and produce JMS messages

- ___1. If the MQ Explorer is not already running, you can launch it from the icon in the lower right hand corner of the screen that represents WebSphere MQ on this system.



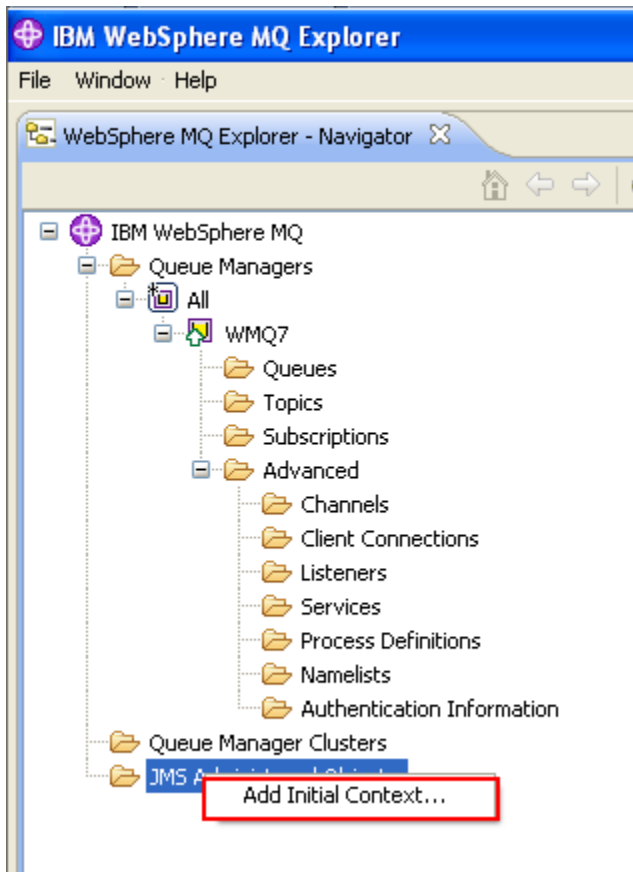
- ___2. Start the MQ Explorer by right-clicking on the icon and selecting **WebSphere MQ Explorer**



2.1 Create Administered Objects using MQ Explorer

- ___1. A directory has been created on your image called C:\JMS. This directory will hold the JNDI Namespace. When you create a connection to the namespace a file named .bindings will be created in this directory.

- __2. Right-click on **JMS Administered Objects** in the Navigator pane, and select **Add Initial Context...**



- ___3. On the **Connection details** screen, click on **File System**. Then click on **Browse** to navigate to the directory called **C:\JMS**. Click Next.

Add Initial Context

Connection details
Enter the location details of the JNDI namespace.

JMS administered objects are stored in Java Naming and Directory Interface (JNDI) namespaces. An Initial context defines the root of a JNDI namespace and is used to access the JMS objects that are stored in the namespace.

Where is the JNDI namespace located?

LDAP server

File system

Other

JNDI Service Provider

Factory class:

JNDI Namespace Location

Bindings directory:

Provider URL:

< Back **Next >** Finish Cancel

- ___4. On the **'User preferences'** screen, enter the Context nickname **Context1**. Note this can be a name of your choice but for this lab you will use **Context1**. This name will not be used elsewhere. Check both the **Connect immediately on finish** and **Automatically reconnect to context on startup** checkboxes, and then click **Finish**.

Add Initial Context

User preferences
Configure user preferences for accessing the Initial Context.

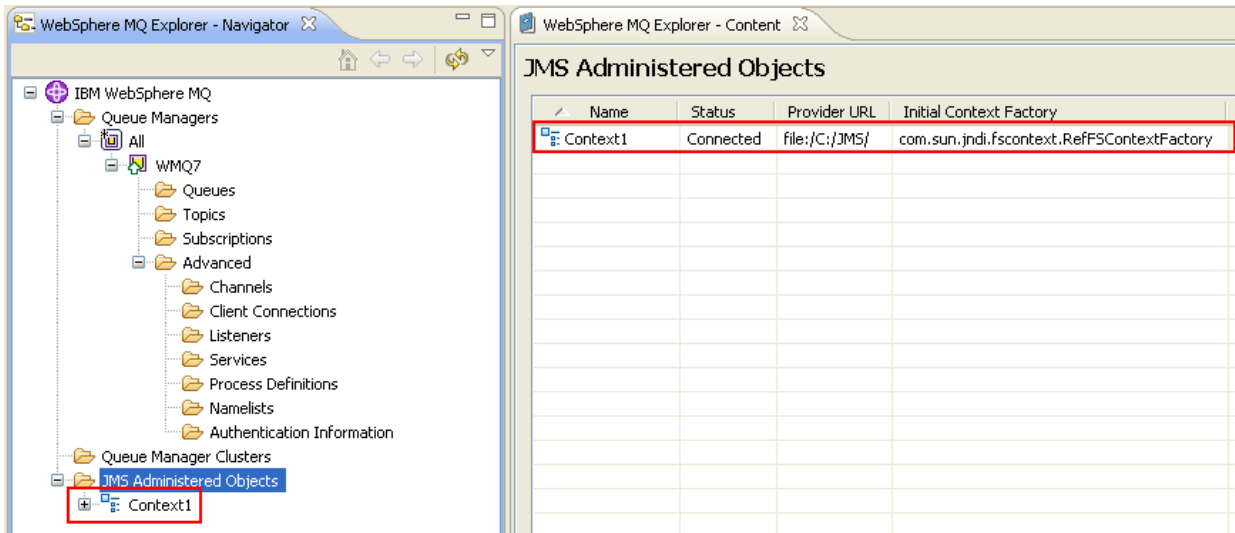
Context nickname:

Connect immediately on finish

Automatically reconnect to context on startup

< Back Next > **Finish** Cancel

__5. The newly created initial context is displayed in the list.

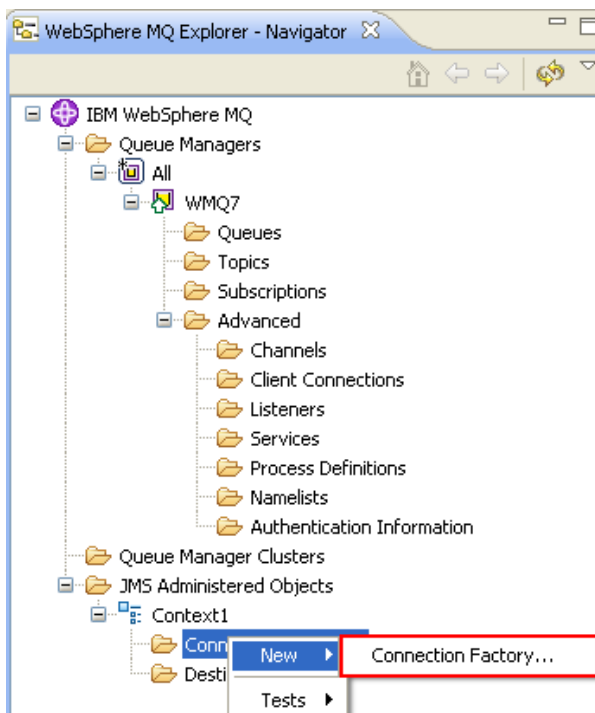


This concludes this portion of Lab 2.

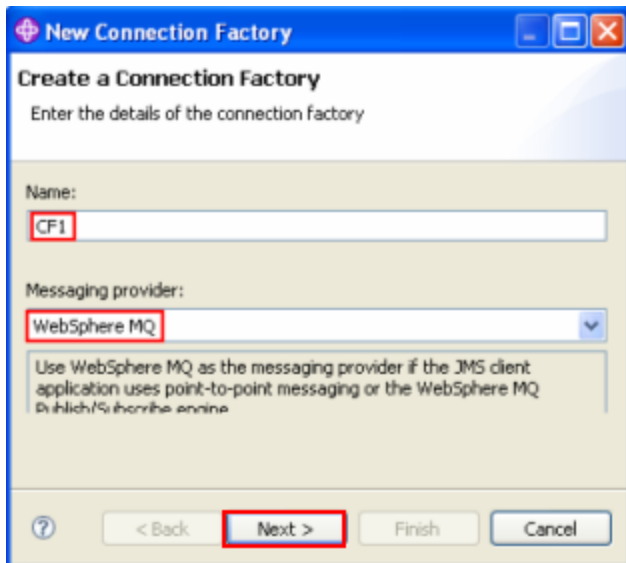
2.2 Create a connection factory for WebSphere MQ

A connection factory is the mechanism used by JMS to manage connections between your JMS application and the JMS Provider. You will now define a connection factory in the JNDI namespace.

__1. Expand the context name you just created. Right-click **Connection Factories**. Select **New-> Connection Factory**.



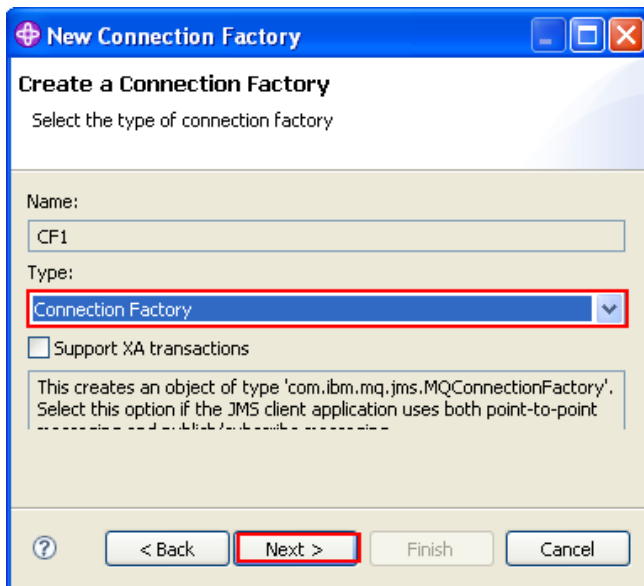
- __2. Enter the name **CF1**. Note this can be any name of your choice but for this lab you will use CF1. This will be required when running the program so you might want to make note of it. Accept **WebSphere MQ** as the messaging provider and then click **Next >**



The screenshot shows the 'New Connection Factory' dialog box with the following details:

- Title: New Connection Factory
- Section: Create a Connection Factory
- Instruction: Enter the details of the connection factory
- Name: CF1
- Messaging provider: WebSphere MQ
- Text: Use WebSphere MQ as the messaging provider if the JMS client application uses point-to-point messaging or the WebSphere MQ Publish/Subscribe service.
- Buttons: < Back, Next > (highlighted), Finish, Cancel

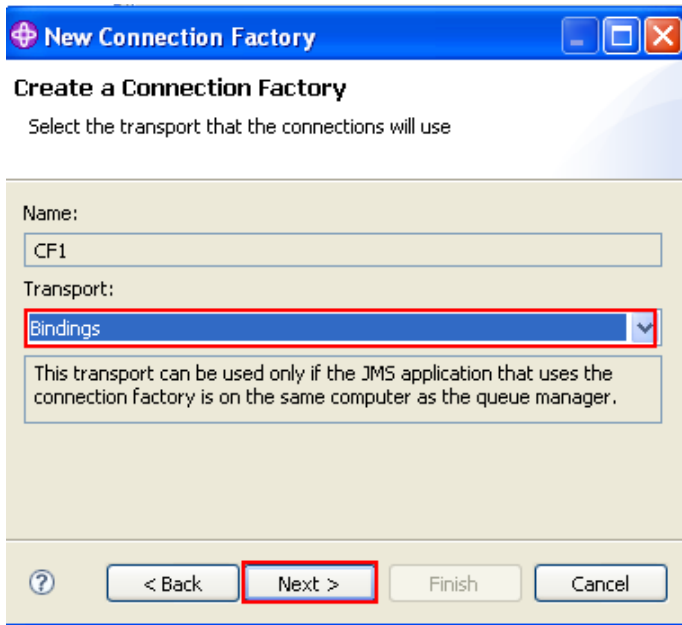
- __3. On the next screen, accept the Type as **Connection Factory** and leave **Support XA transactions** unchecked. Click on **Next >**.



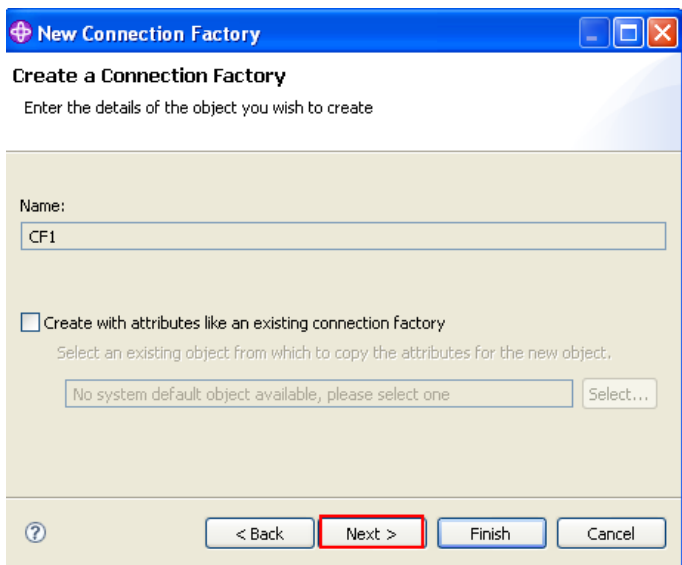
The screenshot shows the 'New Connection Factory' dialog box with the following details:

- Title: New Connection Factory
- Section: Create a Connection Factory
- Instruction: Select the type of connection factory
- Name: CF1
- Type: Connection Factory
- Support XA transactions:
- Text: This creates an object of type 'com.ibm.mq.jms.MQConnectionFactory'. Select this option if the JMS client application uses both point-to-point and publish/subscribe messaging.
- Buttons: < Back, Next > (highlighted), Finish, Cancel

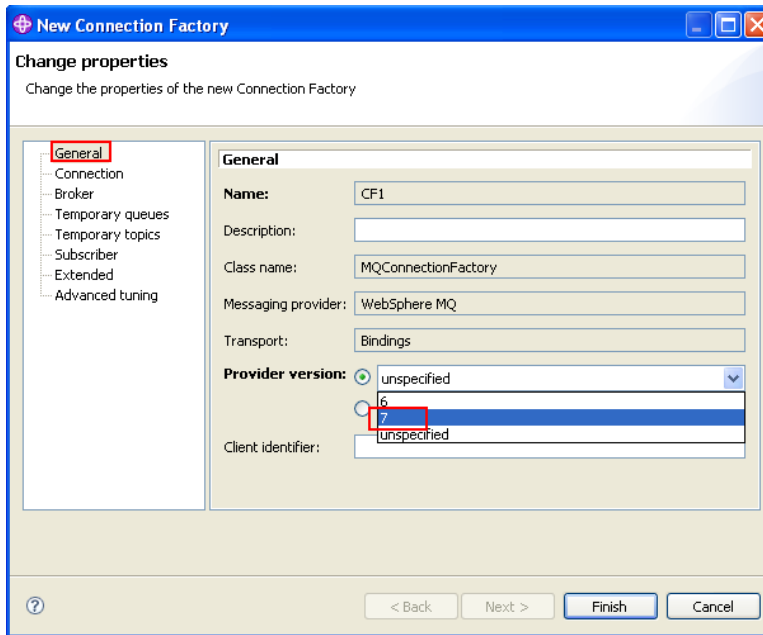
- ___4. On the next screen, accept **Bindings** as the Transport given we will run the JMS Application on the same machine as the Queue Manager. Click **Next >**.



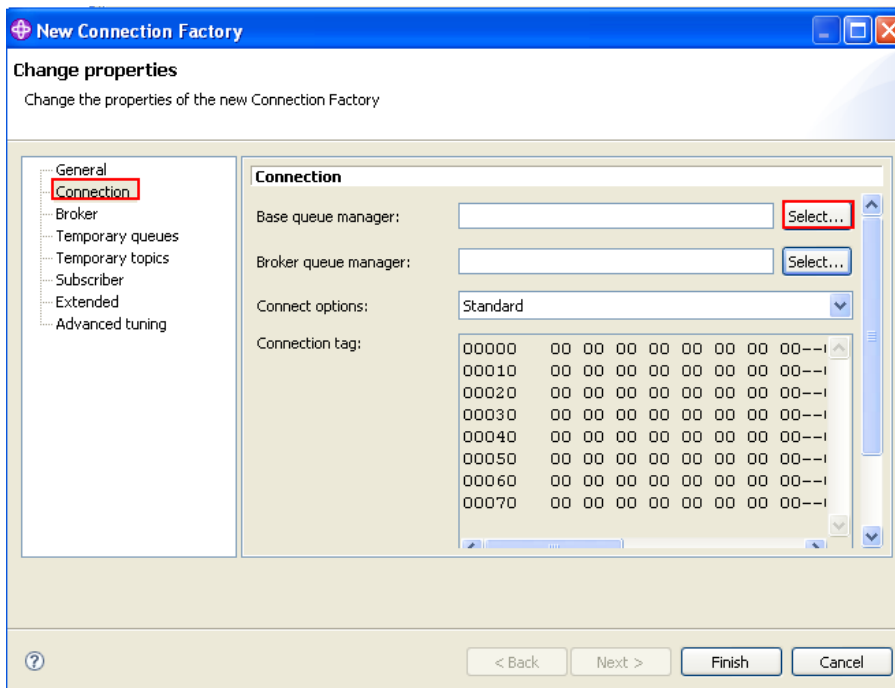
- ___5. On the next screen, leave 'Create with attributes like an existing connection factory' unchecked, and click **Next >**.



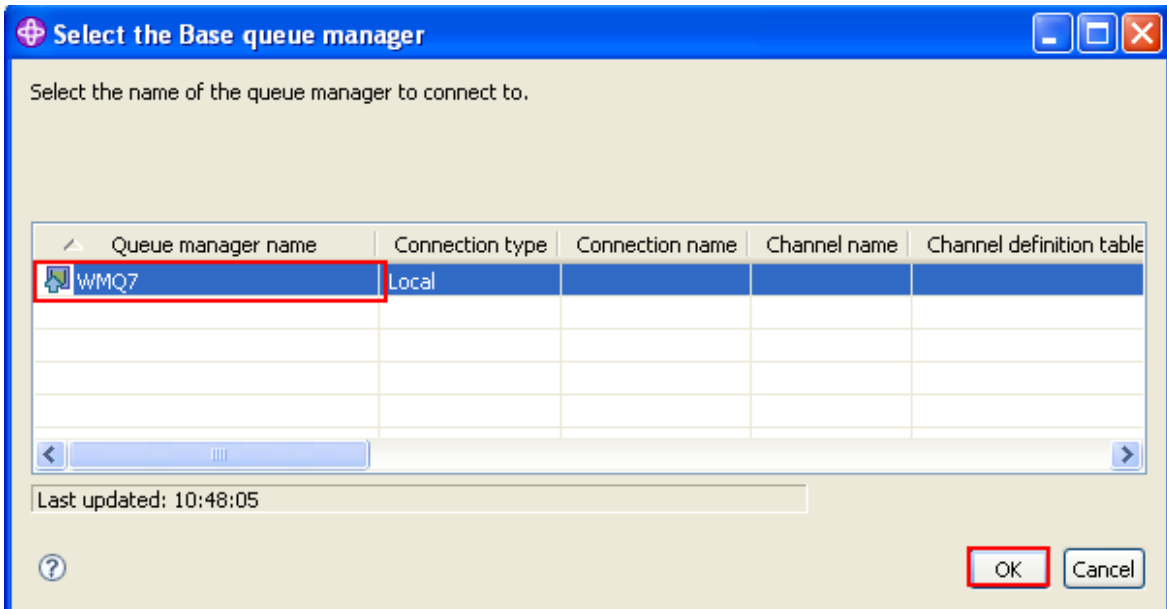
- ___6. This will open the **New Connection Factory** property sheet. Select **General** on the left hand menu. Use the Provider version '7'. This represents WebSphere MQ V7. Selecting this will enable JMS programs using this connection factory to utilize the new features of WebSphere MQ V7.



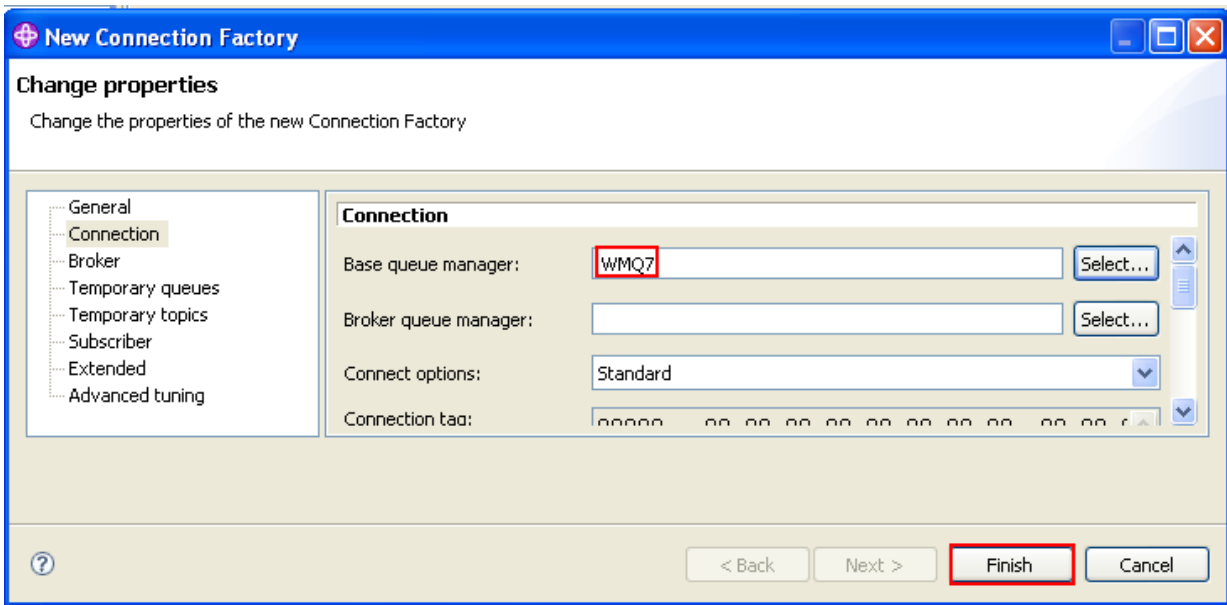
- ___7. Select **Connection** on the left hand menu. Click on the "Select" button for 'Base queue manager'



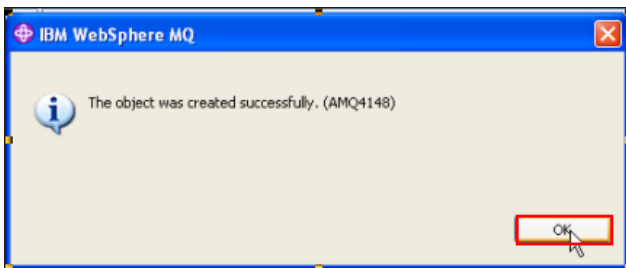
__8. Select the previously-defined Queue Manager **WMQ7** and then click “**OK**”.



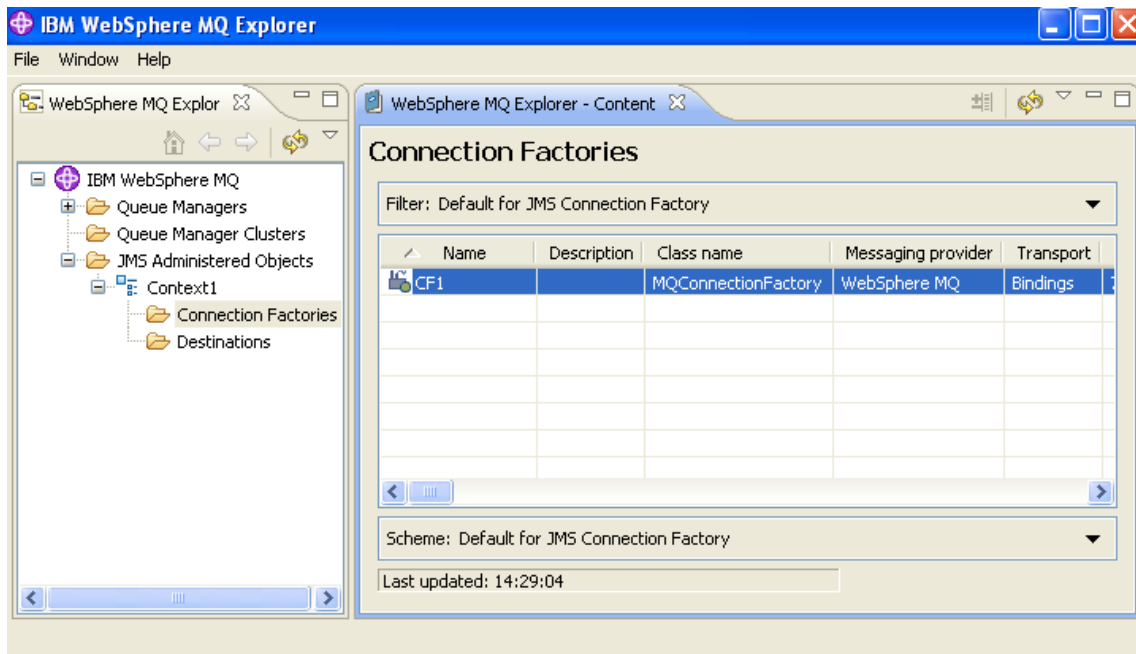
__9. Click **Finish** to create the connection factory.



__10. Click **OK** to dismiss the confirmation box



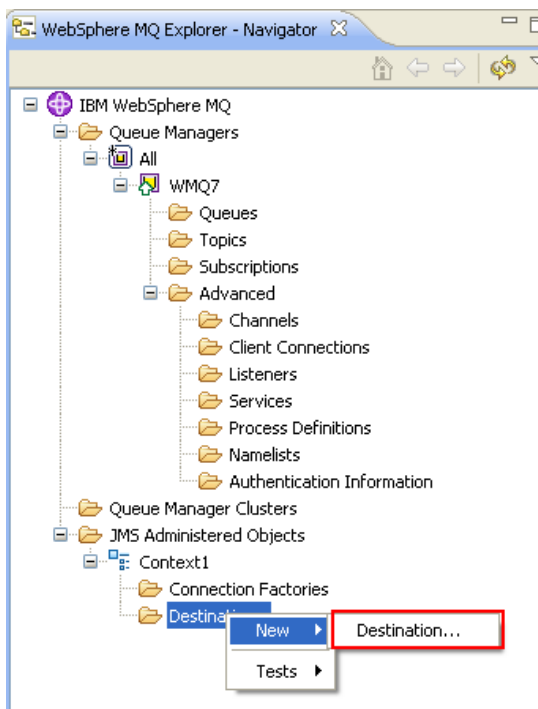
__11. Observe that Connection Factory **CF1** now appears in the Content list.



This concludes this portion of Lab 2.

2.3 Create a Destination for the JMS Application to put a message to.

___1. Right-click 'Destinations' and click New → Destination...



- ___2. Enter the name **JMS1**. Note this can be any name of your choice but for this lab you will use **JMS1**. Leave the Type as **Queue** and ensure **'Start wizard to create a matching MQ Queue'** is checked. This will create a corresponding WebSphere MQ Queue. The Queue that you create will be used to verify that messages are successfully written using the sample program. Click **Next >**.

New Destination

Create a Destination
Enter the details of the object you wish to create

Name:

Messaging provider:
WebSphere MQ and Real-time

A destination that is created in WebSphere MQ Explorer can be used with both WebSphere MQ and Real-time messaging providers.

Type:
Queue

Select this option if the JMS application uses point-to-point messaging. The destination will represent a queue.

When this wizard completes, another wizard can be started automatically to create a matching object.

Start wizard to create a matching MQ Queue

< Back **Next >** Finish Cancel

- ___3. Accept the defaults on this next screen and click **Next >**.

New Destination

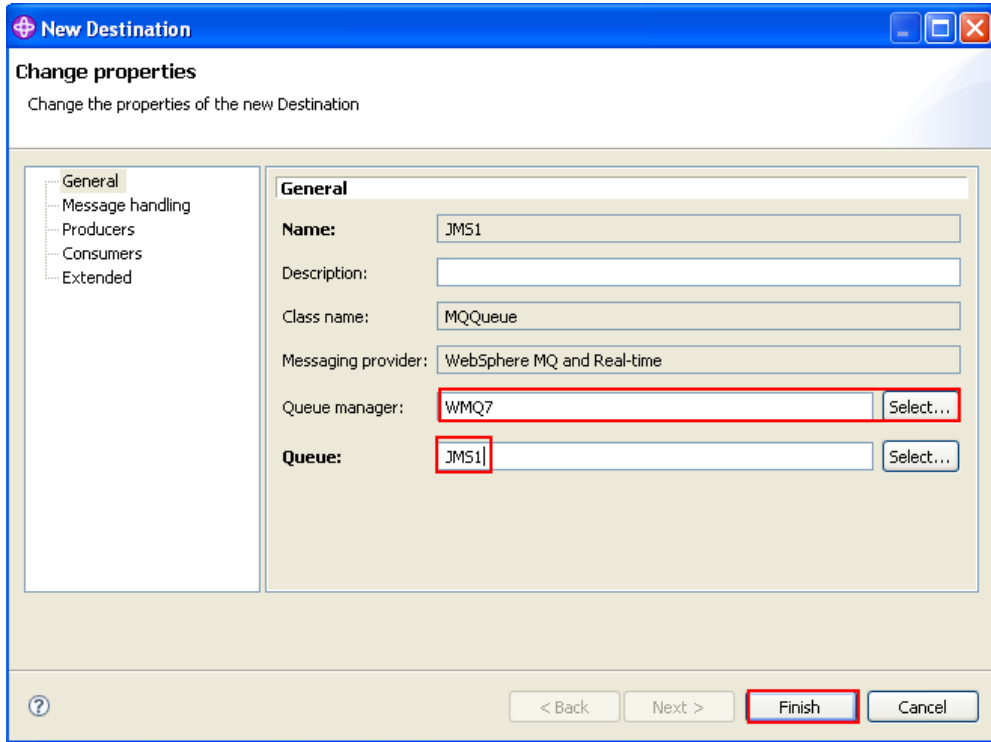
Create a Destination
Enter the details of the object you wish to create

Name:

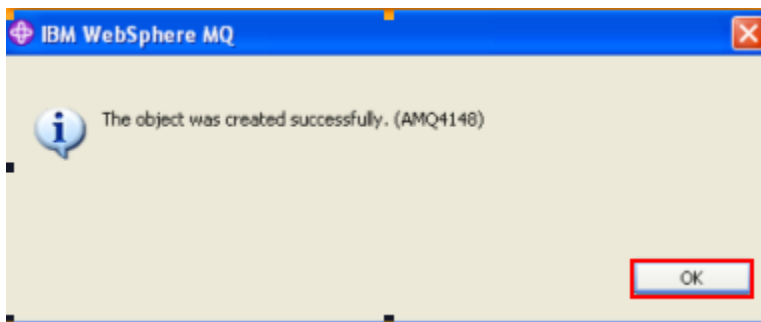
Create with attributes like an existing destination
Select an existing object from which to copy the attributes for the new object.

< Back **Next >** Finish Cancel

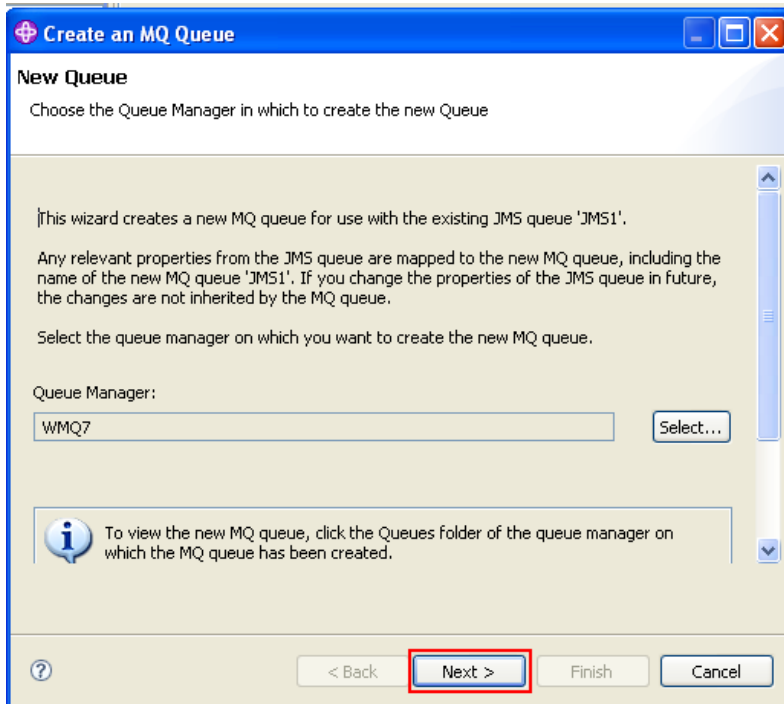
- ___4. On the change properties screen you can set some MQ-specific properties of the destination. Select **WMQ7** as the Queue Manager and define a Queue with the same name as the JMS Destination. Note the Queue name can be any name of your choice but for this lab you will use **JMS1**. This will be created as part of the upcoming wizard. Click **Finish** to launch the MQ configuration wizard.



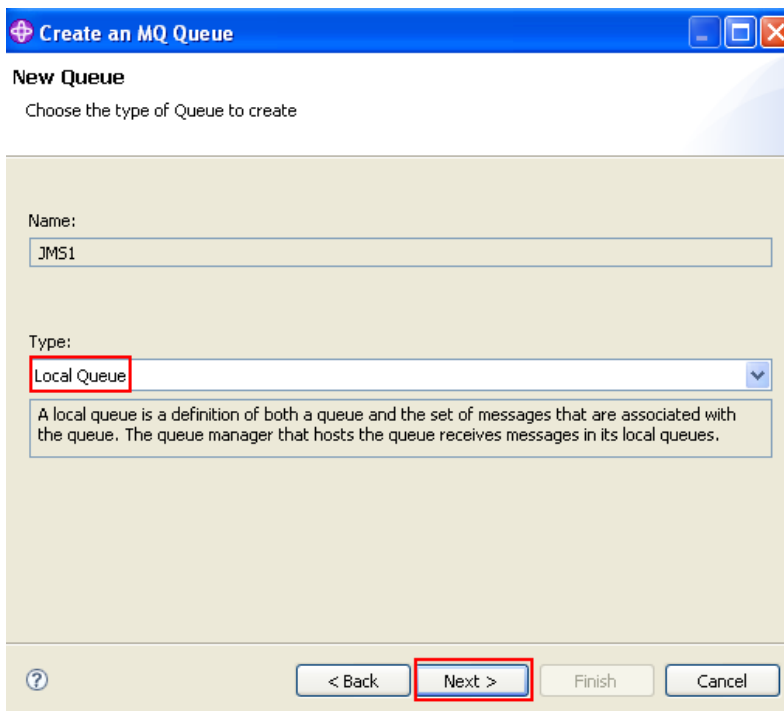
- ___5. Click **OK** to dismiss the confirmation screen.



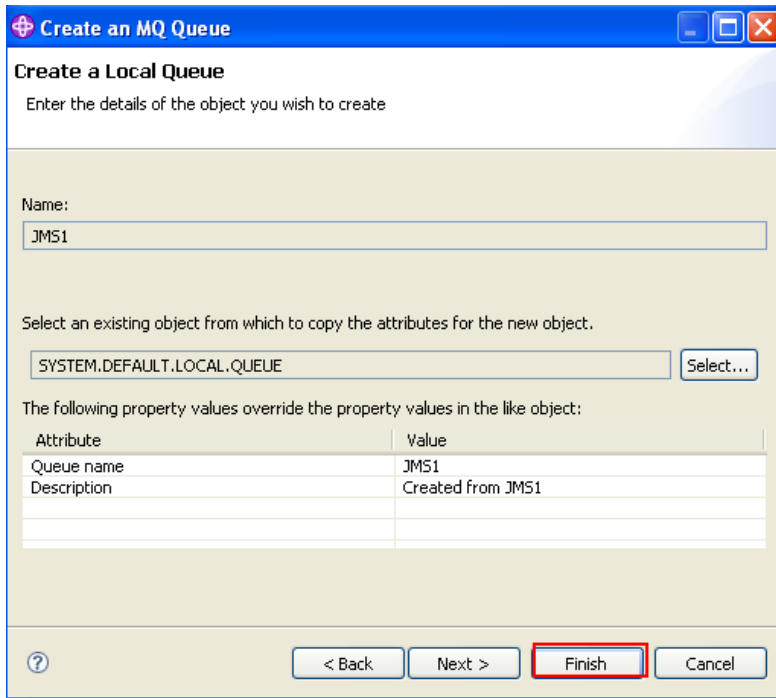
- __6. The **Create an MQ Queue** wizard will start automatically. Click **Next >**.



- __7. Accept the default **'Local Queue'** for Type and then click **Next >**



__8. Click **Finish** to create the MQ queue.



Create an MQ Queue

Create a Local Queue
Enter the details of the object you wish to create

Name:
JMS1

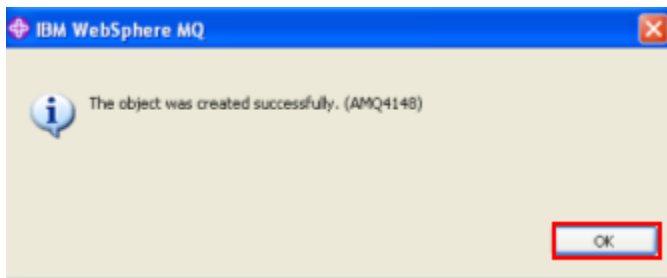
Select an existing object from which to copy the attributes for the new object.
SYSTEM.DEFAULT.LOCAL.QUEUE Select...

The following property values override the property values in the like object:

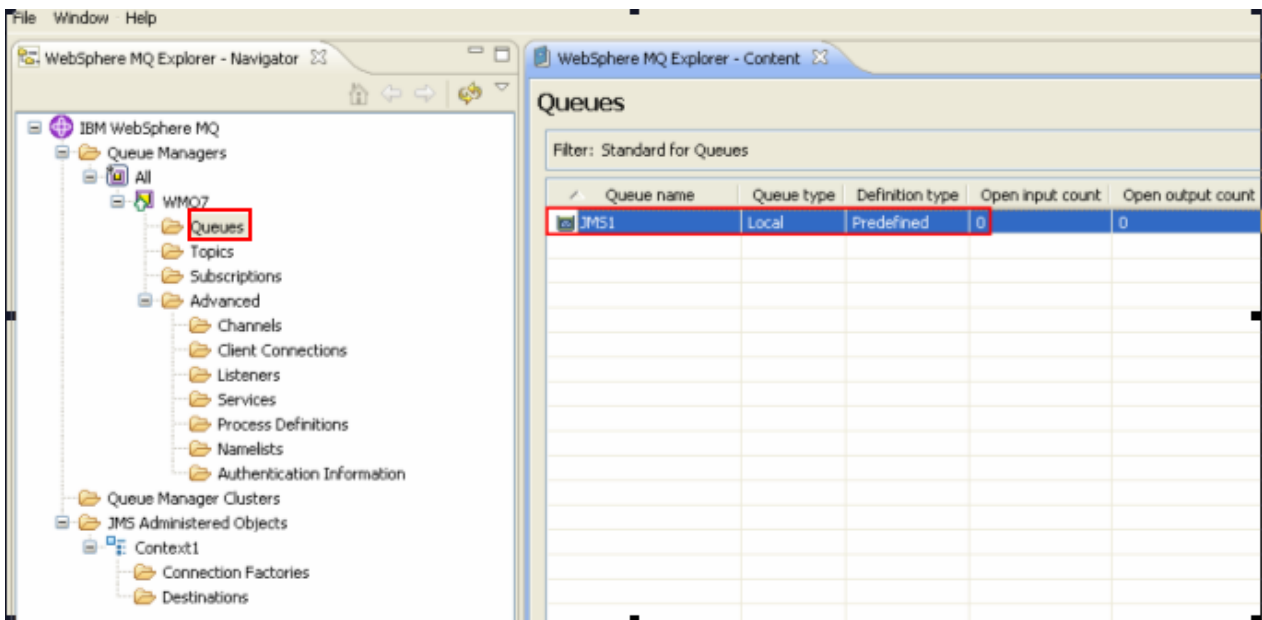
Attribute	Value
Queue name	JMS1
Description	Created from JMS1

< Back Next > **Finish** Cancel

__9. Click **OK** to close the confirmation prompt.



- __10. Click on the Queues folder under WMQ7 in the MQ Explorer Navigator pane, and look at the Content pane to confirm that an MQ local queue called **JMS1** has been created.



This concludes this portion of Lab 2.

2.4 Writing a JMS message using a Java sample program

Some very good code samples ship with WebSphere MQ V7 and you will use one of them to create a JMS message on an MQ JMS queue.

- __1. Open a command prompt. A shortcut for this is in the start menu.



__2. Type in the following command:

```
java JmsJndiProducer -i file:/c:/jms/ -c CF1 -d JMS1
```

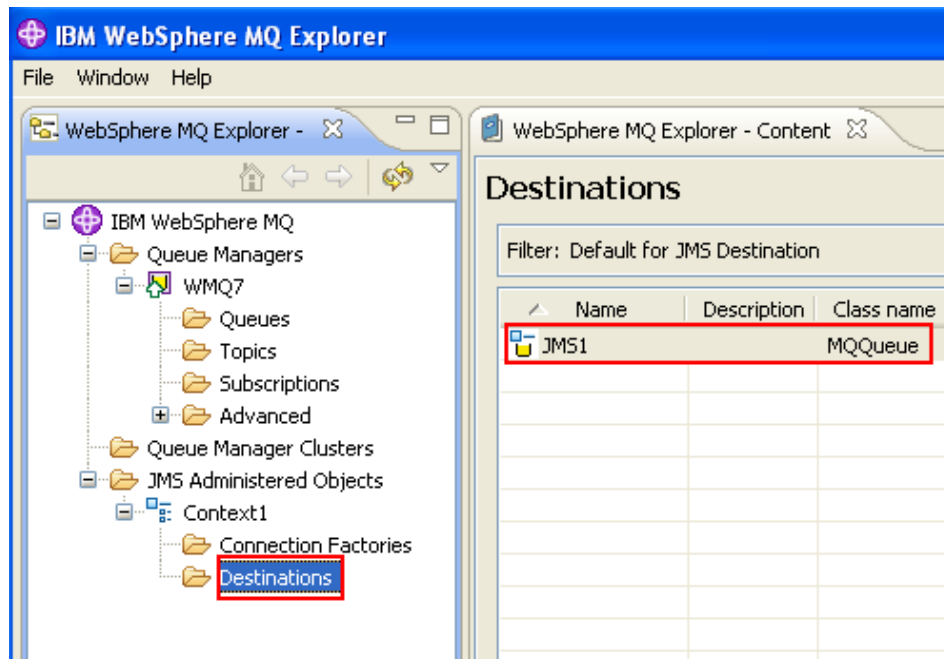
This will run the Java JMS sample program JmsJndiProducer. The “-i” argument points the program to the location of your JNDI directory. The “-c CF1” identifies the connection factory for your test queue manager, and the “-d JMS1” identifies the JMS destination queue.

```
C:\>java JmsJndiProducer -i file:/c:/jms/ -c CF1 -d JMS1
Initial context found!
Sent message:

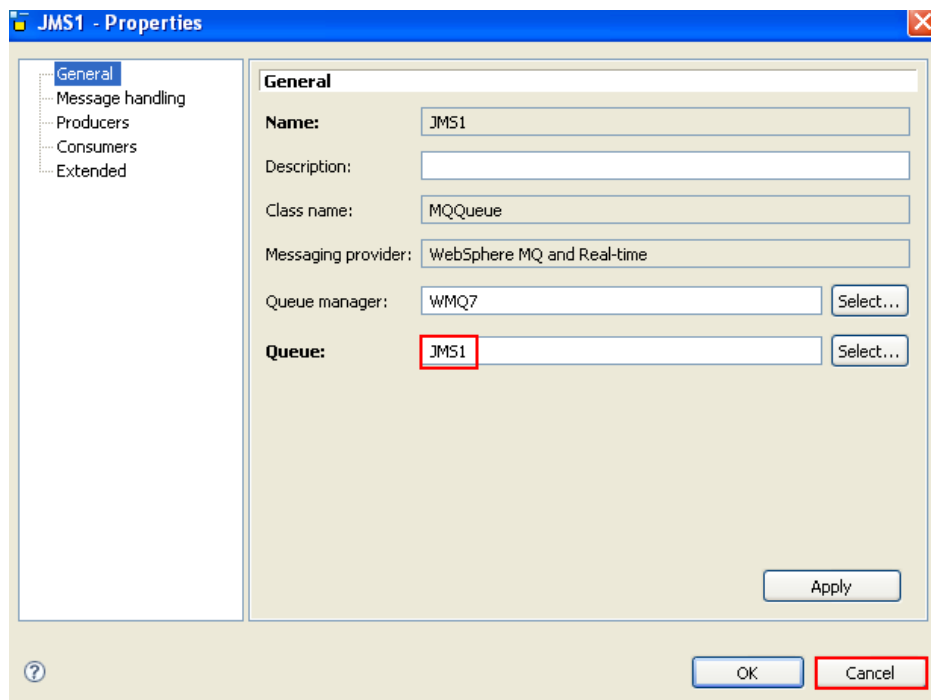
JMSMessage class: jms_text
JMSType: null
JMSDeliveryMode: 2
JMSExpiration: 0
JMSPriority: 4
JMSMessageID: ID:414d5120574d5137202020202020202020202020203540a44820020d02
JMSTimestamp: 1218739184686
JMSCorrelationID: null
JMSDestination: queue://WMQ7/JMS1
JMSReplyTo: null
JMSRedelivered: false
JMSXAppID: ere MQ\java\jre\bin\java.exe
JMSXDeliveryCount: 0
JMSXUserID: student
JMS_IBM_PutApplType: 11
JMS_IBM_PutDate: 20080814
JMS_IBM_PutTime: 18394470
msJndiProducer: Your lucky number today is 655
SUCCESS
C:\>
```

Having run this program, you should now have a message in the **JMS1** queue. To see whether you do, you can look for the message that was produced by this program using the WebSphere MQ Explorer. Switch back to the MQ Explorer window

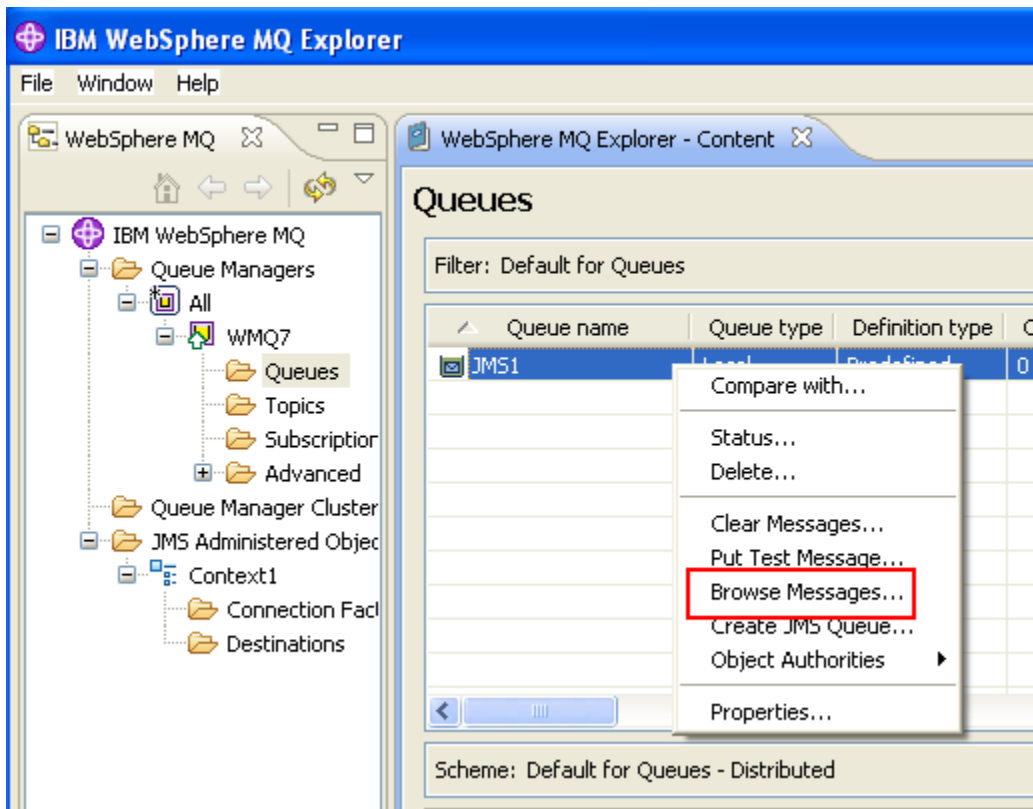
- ___3. In the MQ Explorer Navigator pane, locate the folder called **JMS Administered Objects**. Click on the **Destinations** folder beneath it, and in the Content pane you should see your JMS Destination **JMS1**. Double-click on the JMS1 destination to find the MQ Queue that is associated with it.



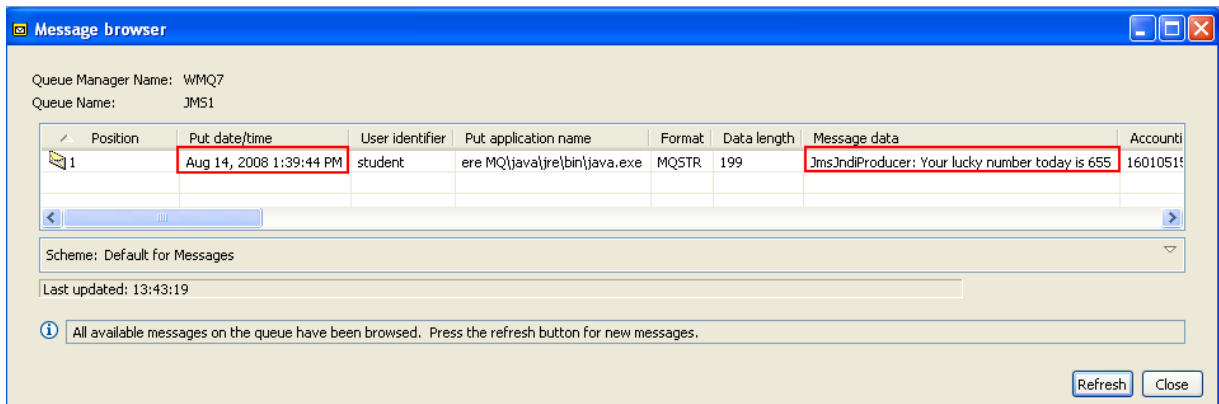
- ___4. Recall that MQ queue **JMS1** is associated with the JMS destination **JMS1**. Click on **Cancel**.



- __5. Right-click on the JMS1 MQ queue object and then select **Browse Messages...**



- __6. The message data column in the display should match the message text that you wrote with the JmsJndiProducer program.



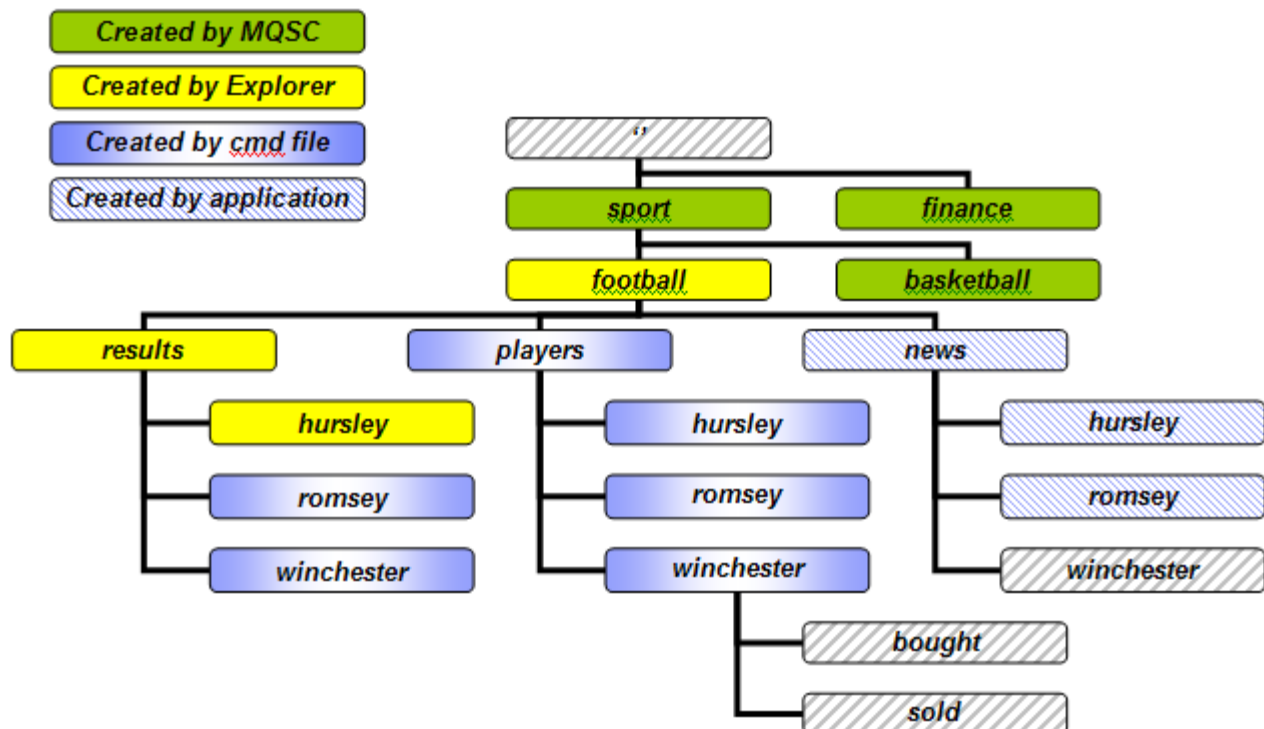
This concludes Lab 2.

Lab 3 Publish / Subscribe Administration Lab

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.

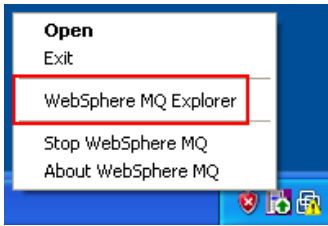


3.2 Using MQ Explorer to create and display information

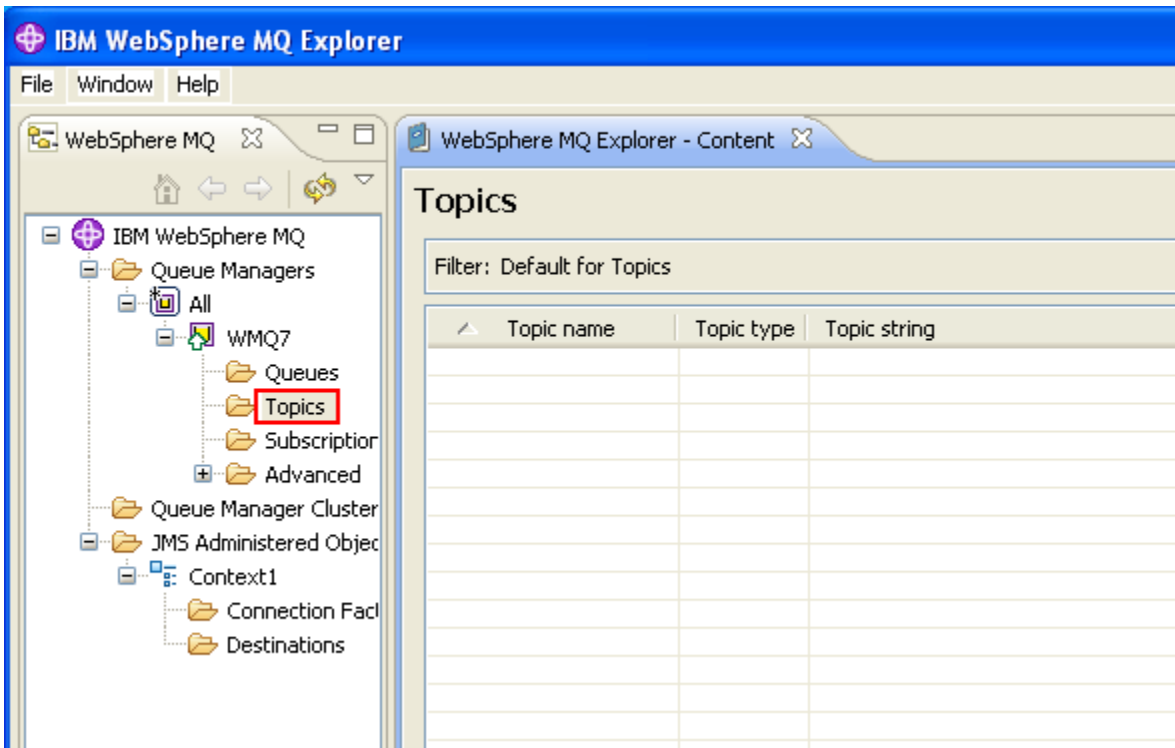
- ___1. If the MQ Explorer is not already running, you can launch it from the icon in the lower right hand corner of the screen that represents WebSphere MQ on this system.



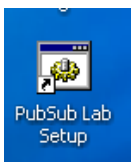
- __2. Start the MQ Explorer by right-clicking on the icon and selecting **WebSphere MQ Explorer**



- __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.



- __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.

```

: SUB<ASPARENT>
AMQ8690: WebSphere MQ topic created.
:
? : DEFINE TOPIC<SPORT.FOOTBALL.NEWS> +
: TOPICSTR('sport/football/news') +
: REPLACE +
: PUB<ENABLED> +
: SUB<ENABLED>
AMQ8690: WebSphere MQ topic created.
? MQSC commands read.
No commands have a syntax error.
All valid MQSC commands were processed.

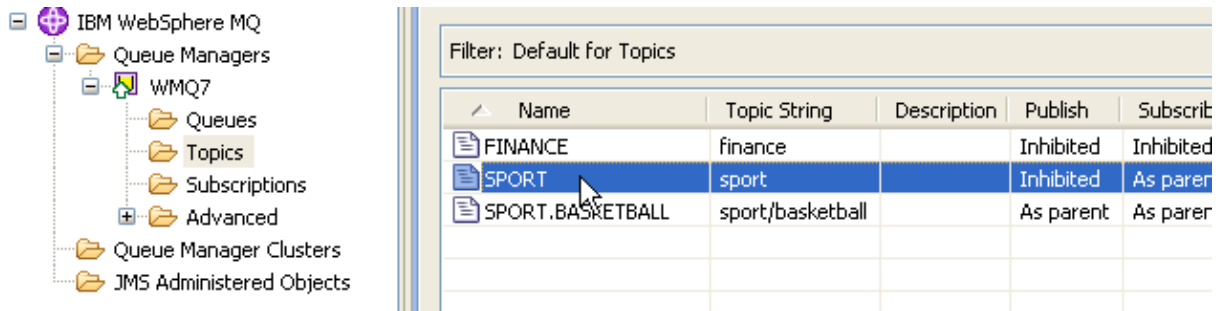
C:\Student\Lab_PubSubAdmin\resources>pause
Press any key to continue . . .

```

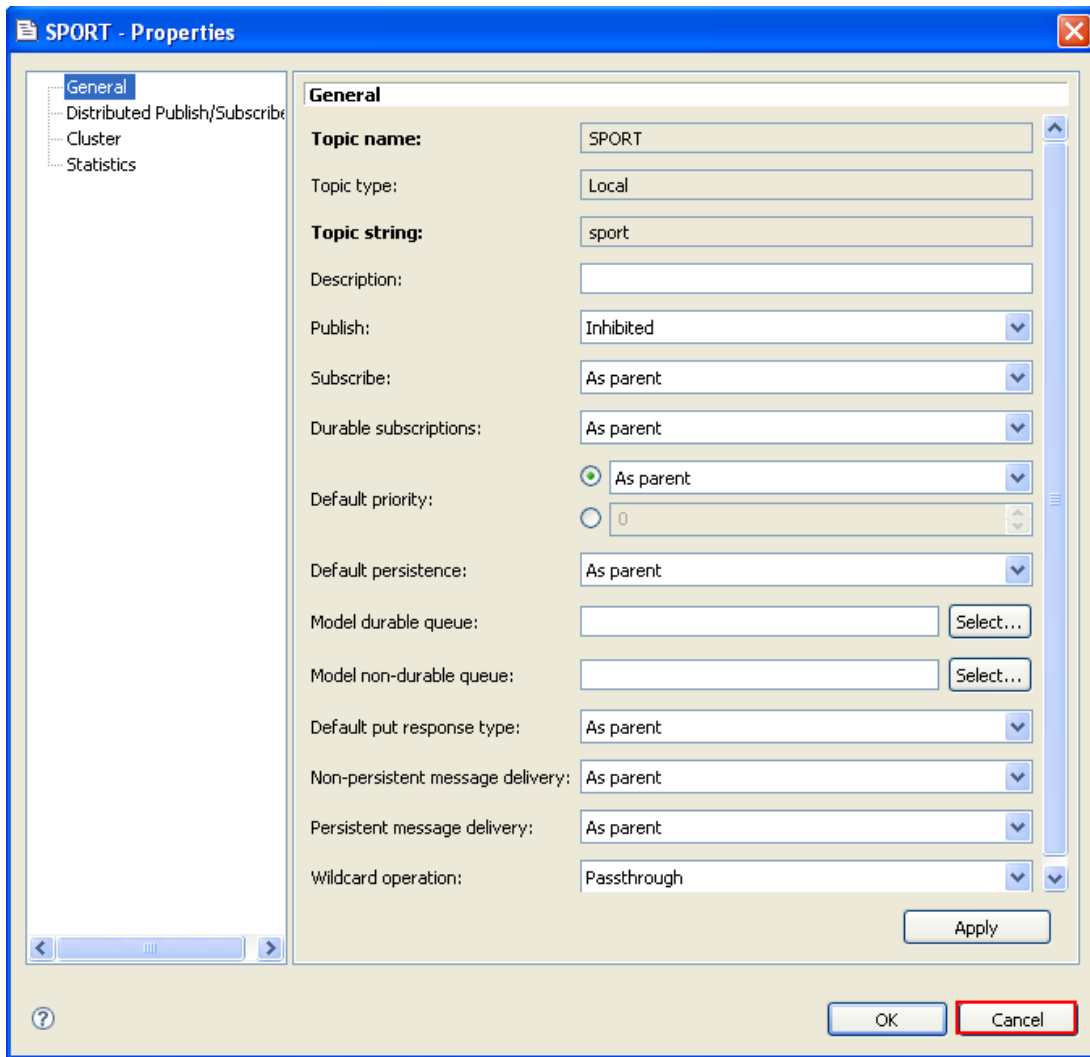
- __6. You should now see the following WebSphere MQ Topic objects displayed in the Content pane in the MQ Explorer. Observe the Topic objects (under the Topic name column) and their corresponding Topic string values. Also observe the Publish and Subscribe enablement status on the right.

Topic name	Topic type	Topic string	De...	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

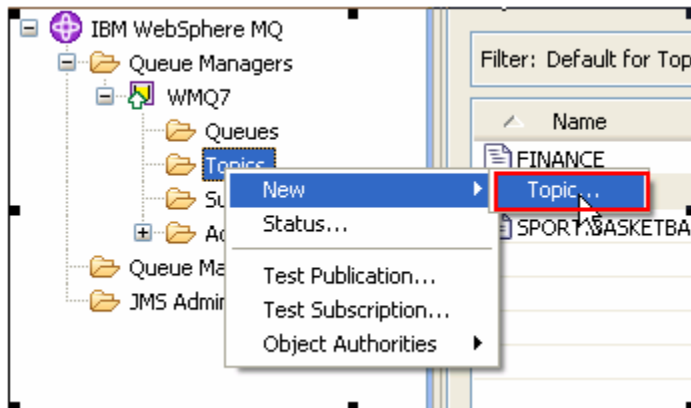
__7. Double-Click on the **SPORT** Topic object.



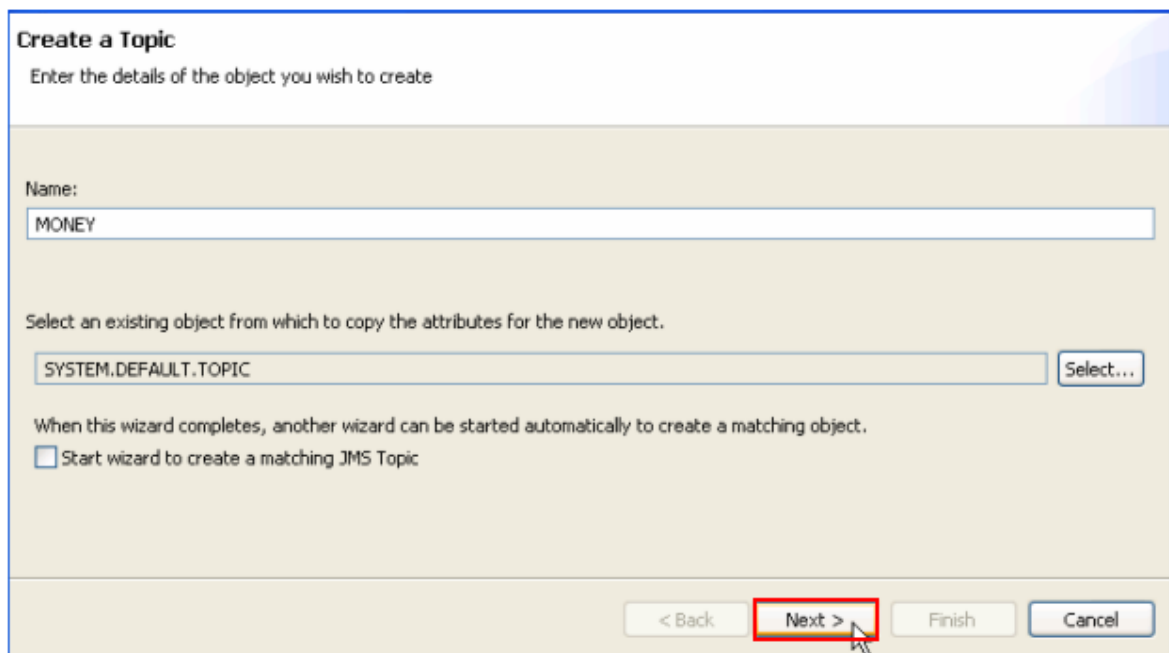
__8. Observe the various properties of the Topic object. Explore the various values available on the pull-downs if you wish. Close the window by clicking “**Cancel**”.



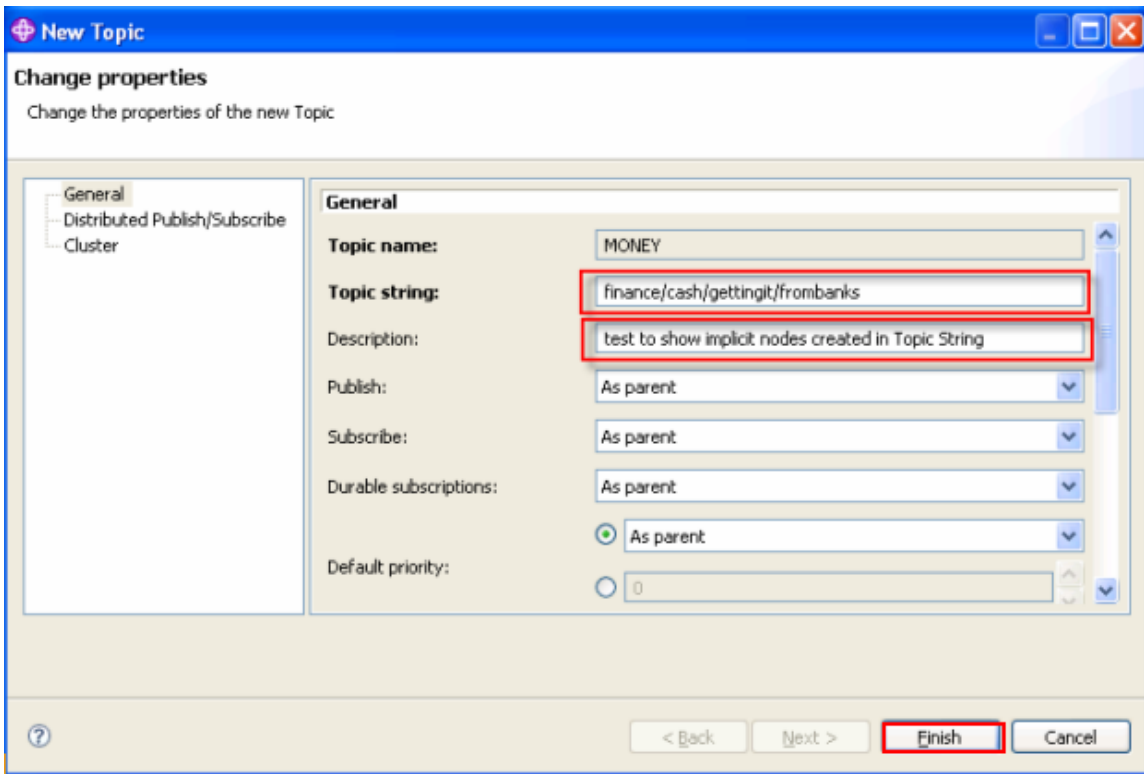
- ___9. From the Navigation pane select “**Topics**” under queue manager WMQ7. Right-click on **Topics** and then select **New**→**Topic** from the context menu.



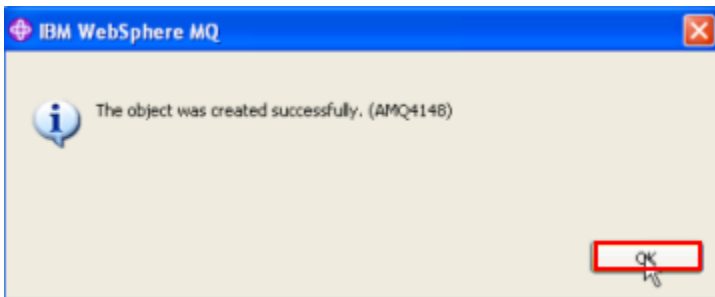
- ___10. Enter **MONEY** in the Name field; allow the other fields to default. Then click “**Next**”.

A screenshot of the 'Create a Topic' wizard dialog box. The title bar reads 'Create a Topic'. Below the title, it says 'Enter the details of the object you wish to create'. There are three main sections: 1. 'Name:' with a text input field containing 'MONEY'. 2. 'Select an existing object from which to copy the attributes for the new object.' with a dropdown menu showing 'SYSTEM.DEFAULT.TOPIC' and a 'Select...' button. 3. 'When this wizard completes, another wizard can be started automatically to create a matching object.' with a checkbox labeled 'Start wizard to create a matching JMS Topic' which is currently unchecked. At the bottom right, there are four buttons: '< Back', 'Next >', 'Finish', and 'Cancel'. The 'Next >' button is highlighted with a red rectangular box.

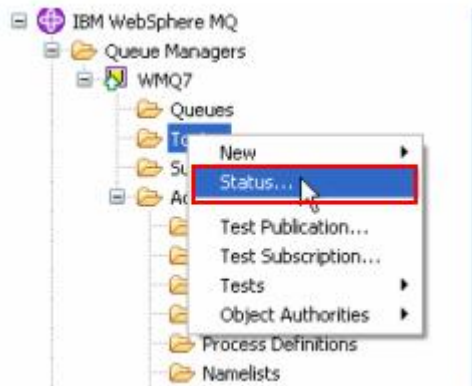
- ___11. Enter **“finance/cash/gettingit/frombanks”** (without the quote marks) in the **Topic string** field. Enter a description in the description field and then click **“Finish”**



- ___12. Close the confirmation box by clicking **“OK”**.



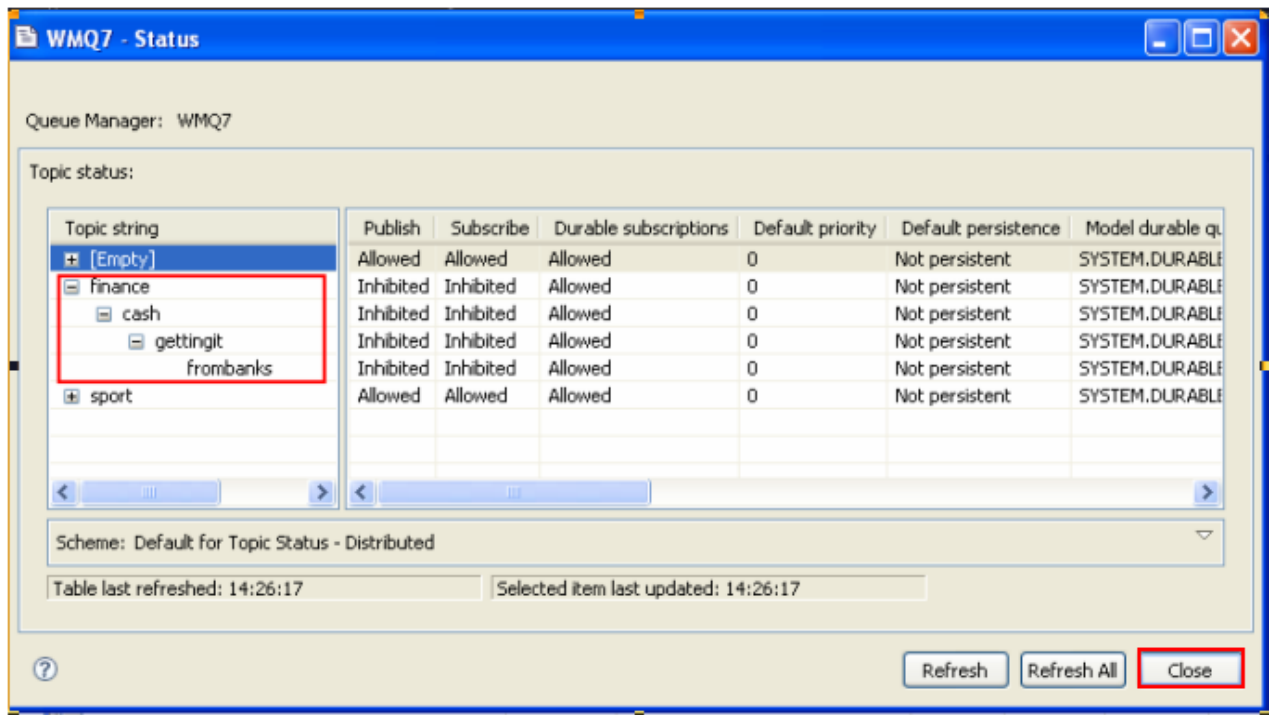
- __13. Back in the MQ Explorer, right click on **“Topics”** (under WMQ7) and then select **“Status”**.



- __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 MQ Explorer.

- __15. Close the Topic Status view by clicking on **“Close”**.

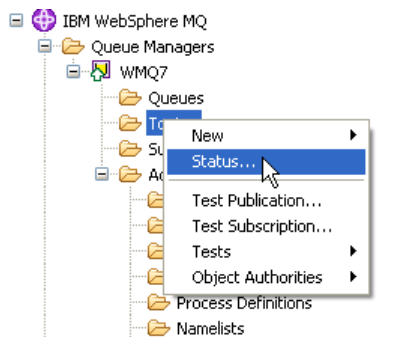


This concludes this portion of Lab 3.

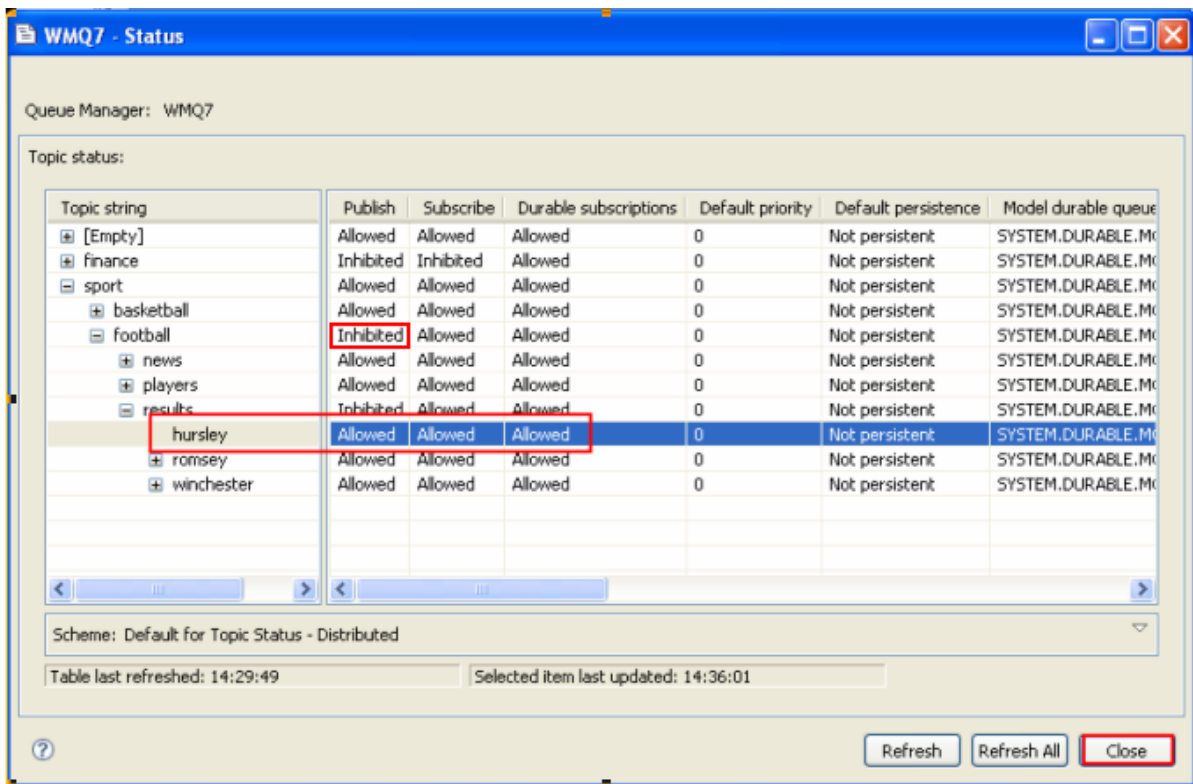
3.3 A First look at the MQ Explorer Pub / Sub Test tools

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.

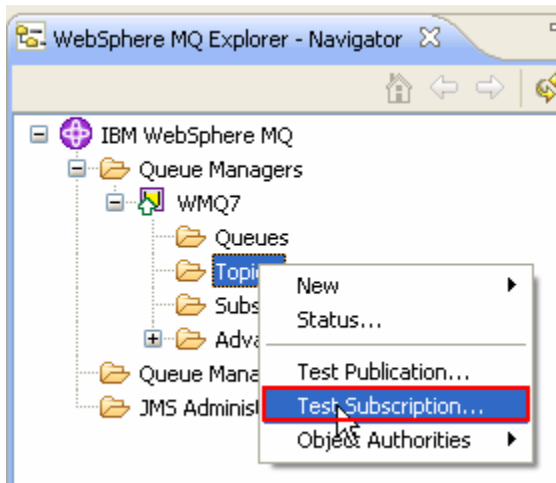
1. You will now be working with some pre-defined topics. From the WebSphere MQ Explorer, display topic status by right-clicking on the **Topics** folder and selecting **Status...**



2. 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.

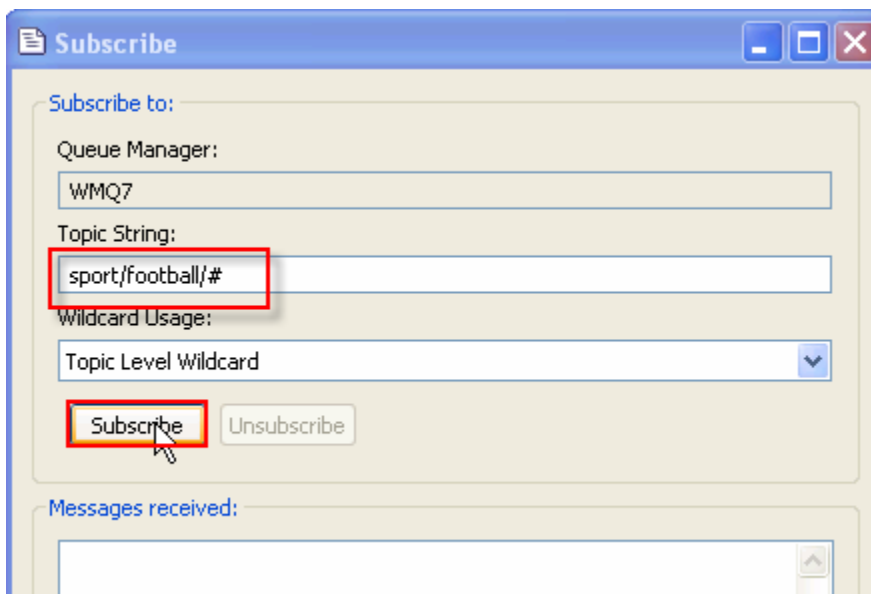


- __3. Start a test subscription window by right-clicking on Topics and selecting “**Test Subscription**”.



- __4. 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.



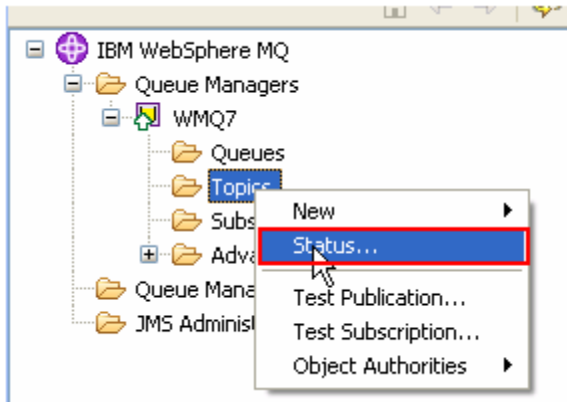
- __5. Minimize the Subscribe window by clicking on the minimize button.



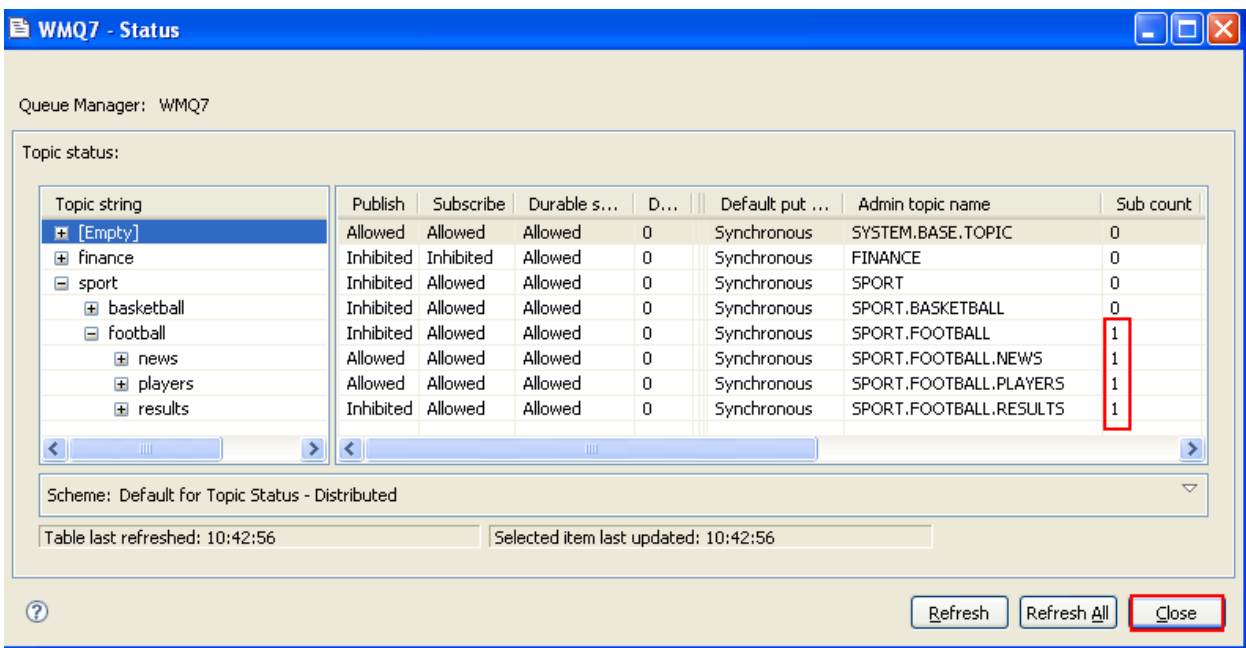
- __6. The Subscribe window will “park” itself at the bottom left part of the Windows desktop. You will restore this window in a later step of this lab.



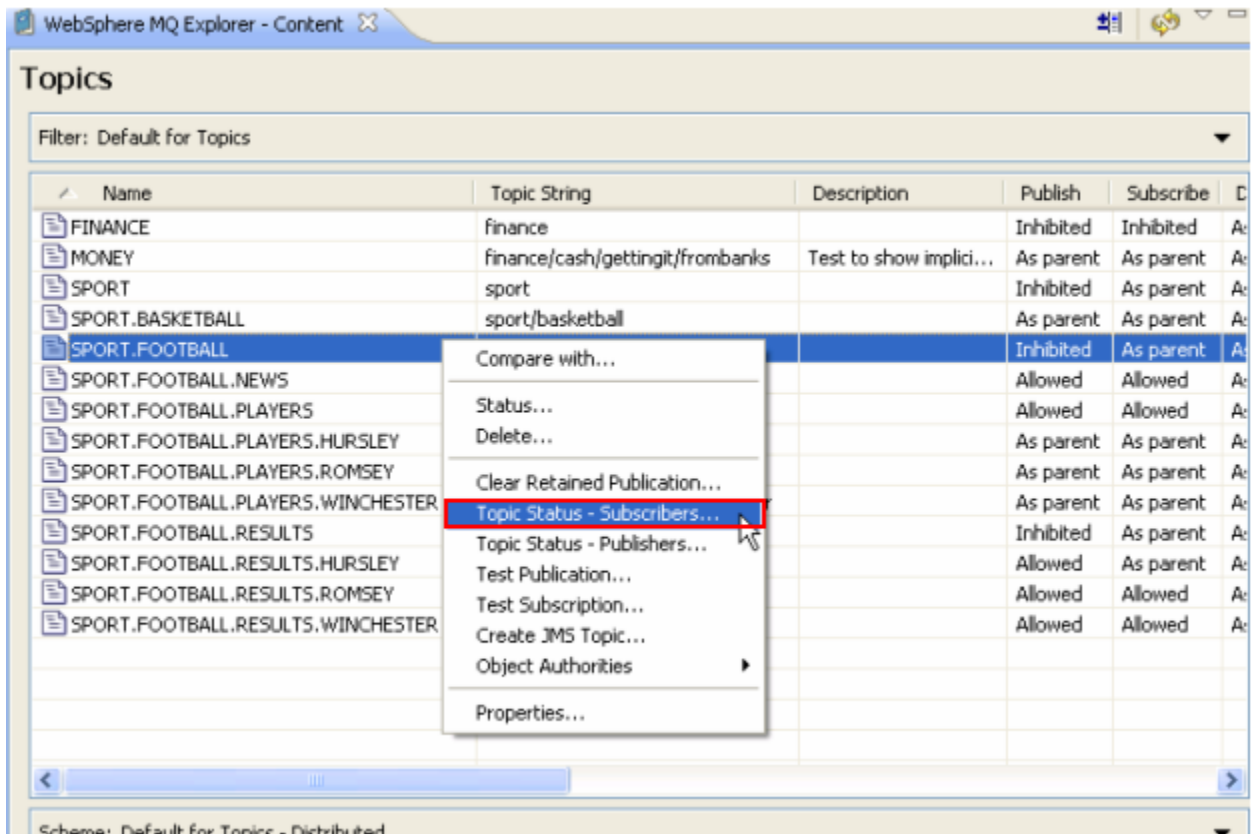
- __7. Right click on the Topics folder then select **Status...**



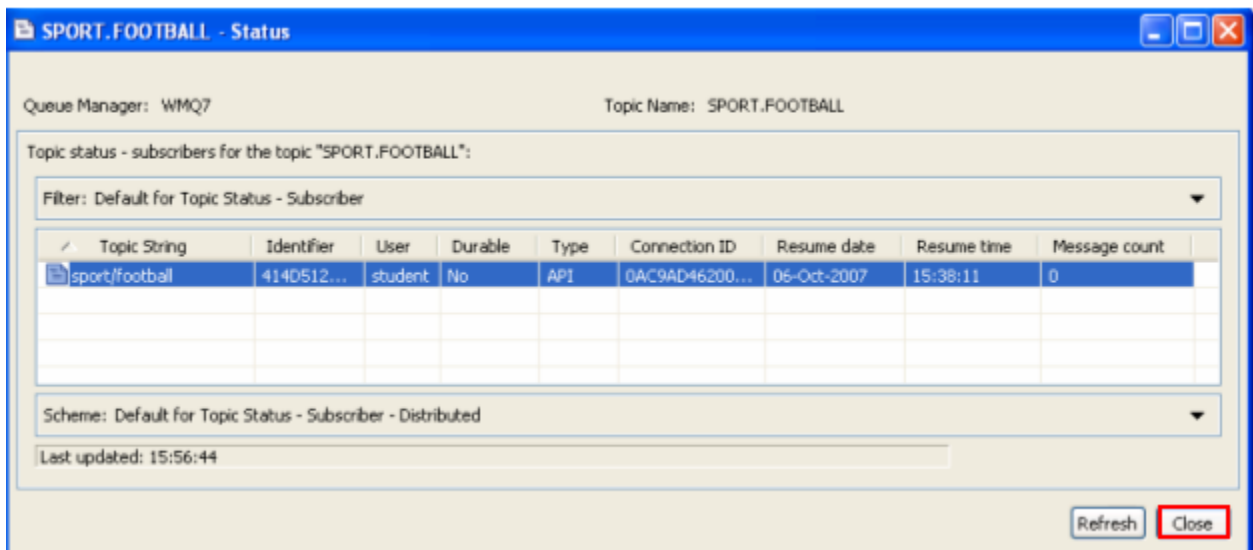
- __8. Expand the **sport** tree. Observe the Subscription counts; sport/football and its children have a positive subscription count. Close the topic status window.



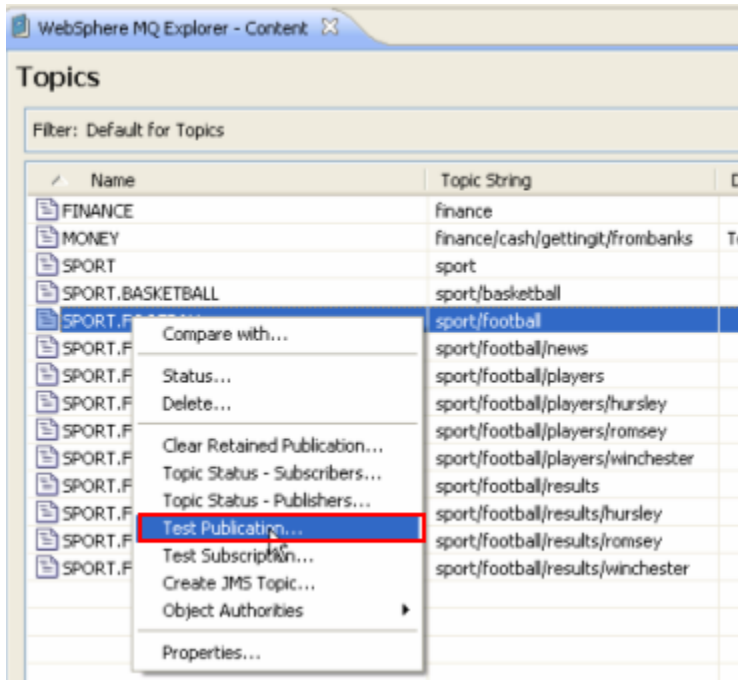
- ___9. From the Topic display, select the “**SPORT.FOOTBALL**” row, right-click and select “**Topic Status – Subscribers**”. This gives detailed information about subscribers to this Topic object.



- ___10. Observe the detailed display and then close the status screen by clicking on the “**Close**” button.



- ___11. Now you will publish a message. Returning to the Topic list, select the **SPORT.FOOTBALL** row, right-click and then select “**Test Publication**”.



- ___12. This dialog will publish a message to the topic string “**sport/football**”.

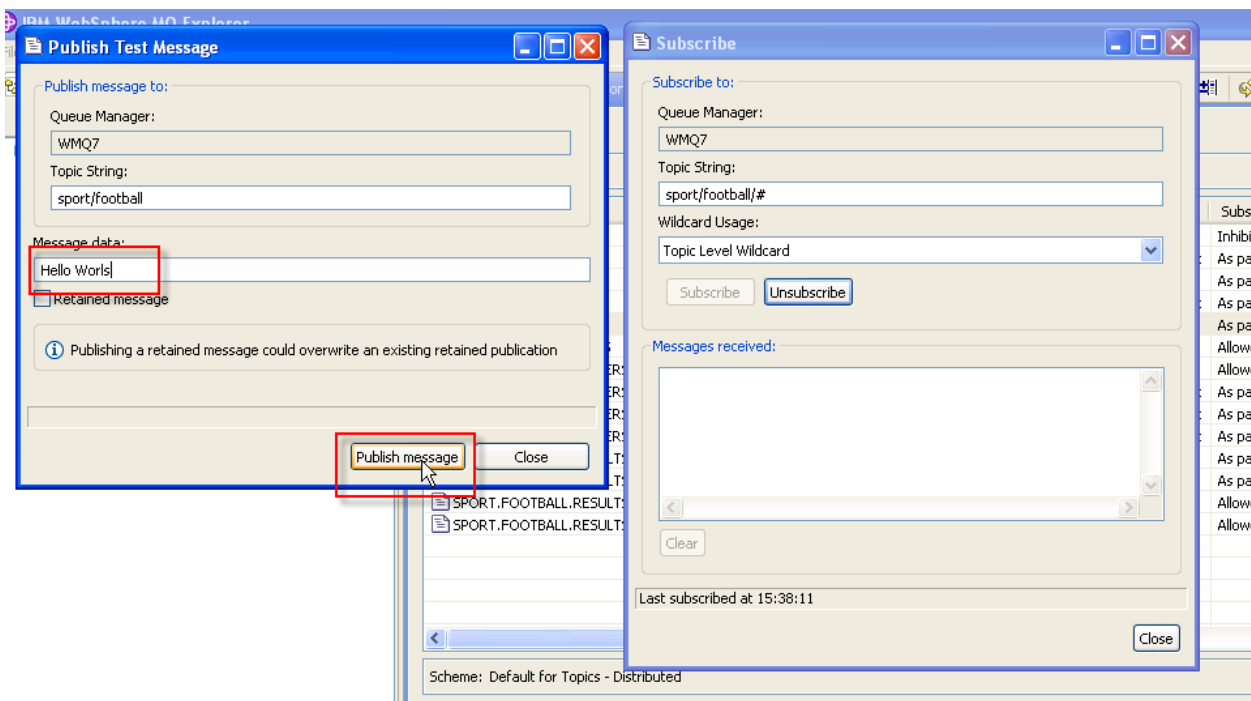


- ___13. Before entering a message and sending it you will arrange the windows on the screen.

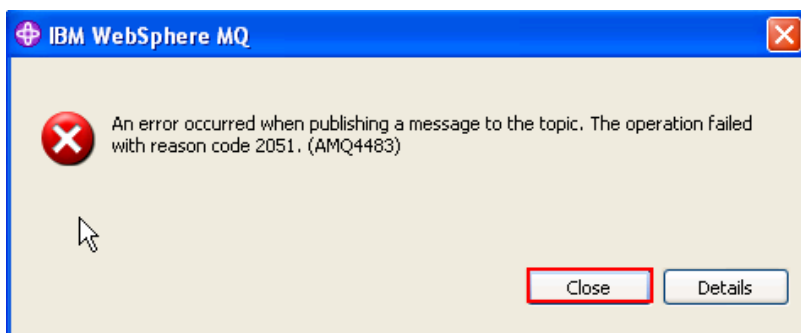
- __14. Locate the “**Subscribe**” Test Tool window where you previously subscribed to “**sport/football/#**”. Click on the Restore window button to restore the window. Now position the **Publish Test Message** and the restored **Subscribe** test tool windows so they both are visible. Then return focus to the **Publish** window.



- __15. Type a message such as “**Hello World**” and then press “**Publish Message**”.
- __16. 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.

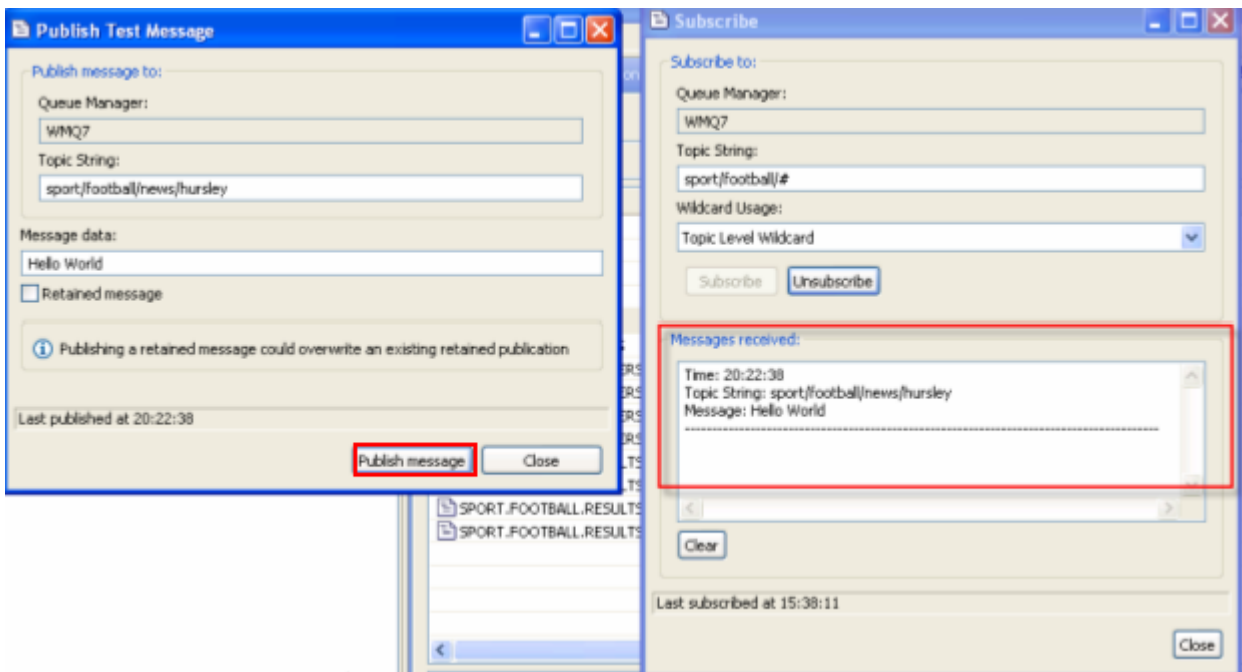
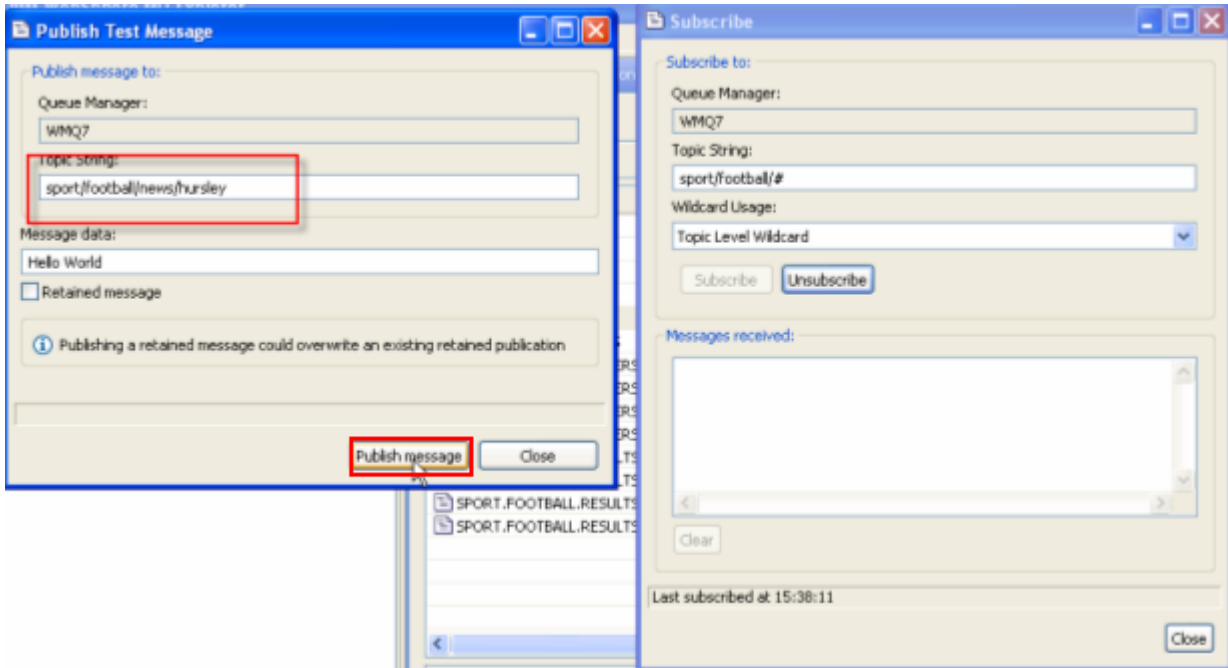


- __17. Click on Close to dismiss the error popup.

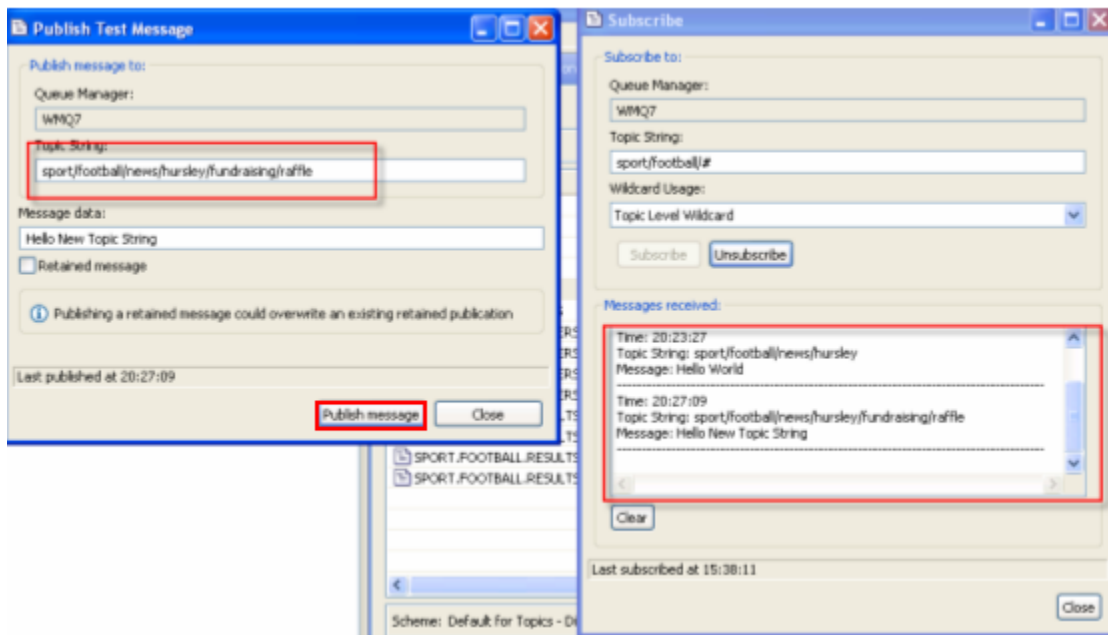


___18. In the Publish Test Message window, overtype the topic string to “**sport/football/news/hursley**” and click on **Publish Message**. You have published and subscribed your first message!

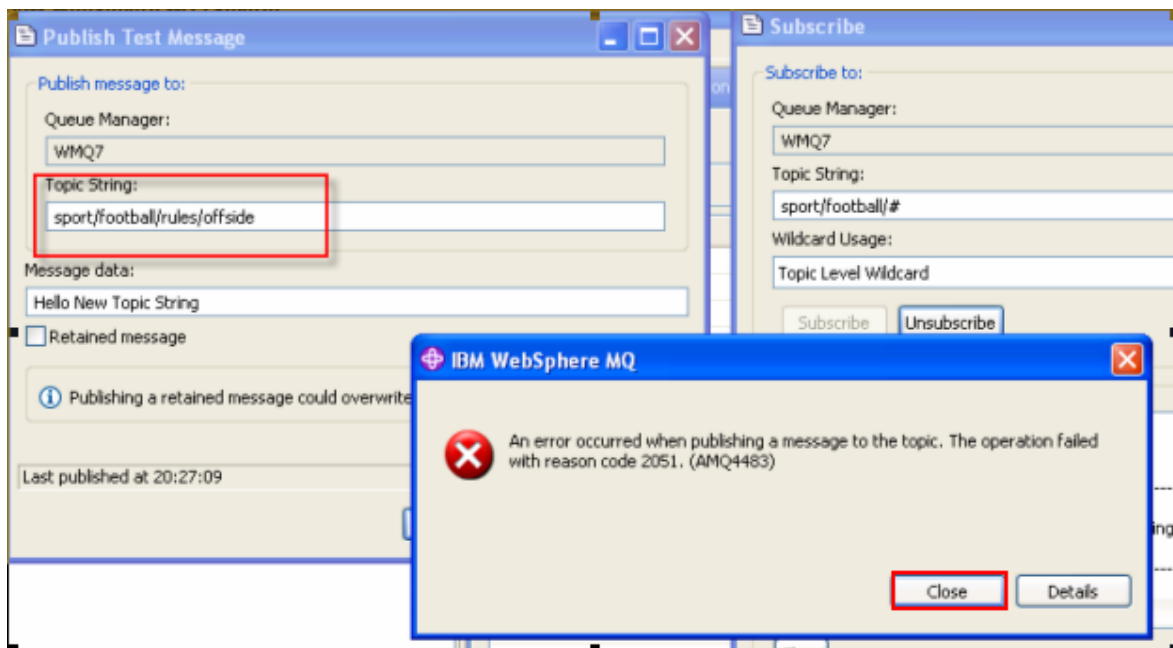
Note: these dynamically created Topic objects are temporary and only exist for a limited amount of time before the queue manager removes them; for example if you restart the Queue Manager, they will no longer exist



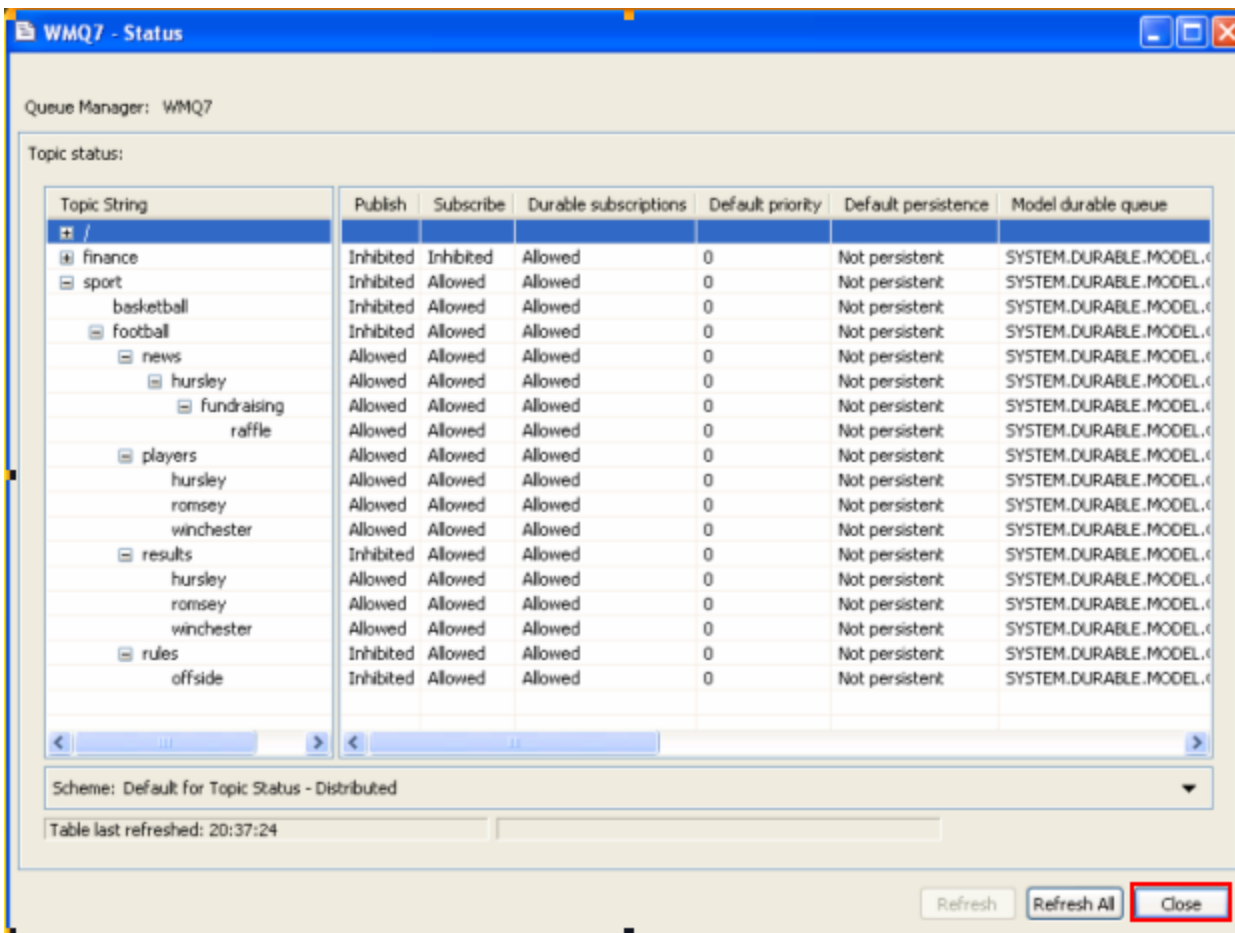
- __19. Now try publishing to “sport/football/news/hursley/fundraising/raffle”. The message is sent to the subscriber. New levels of the hierarchy have been created automatically.



- __20. 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.



- 21. 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.

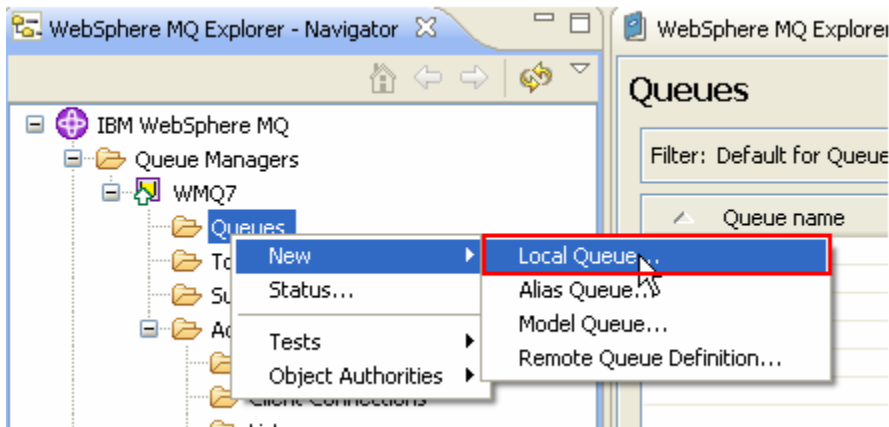


This concludes this portion of Lab 3.

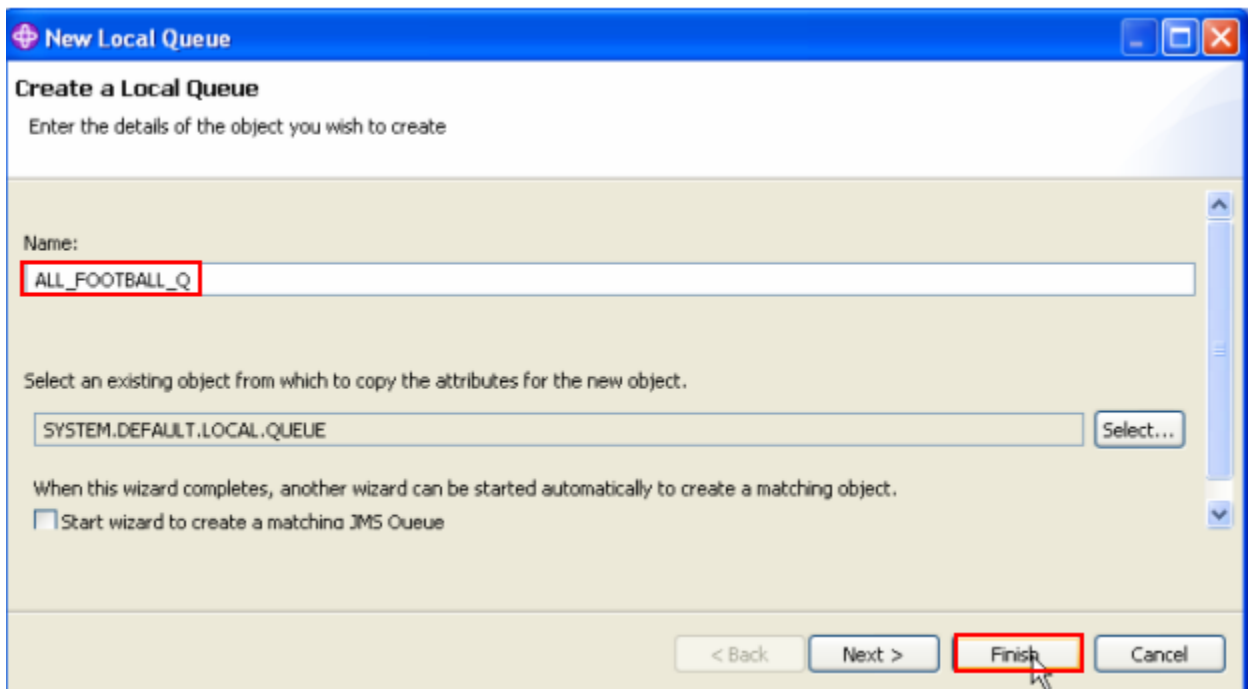
3.4 Administered Subscriptions

While it is typical for subscribers to register their own subscriptions, it is possible to administratively register a subscription using 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.

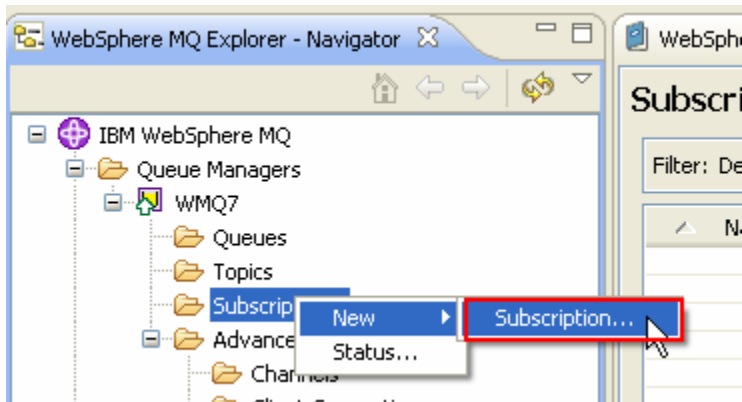
- __1. In the MQ Explorer, select “Queues”. Right-click and select “New→Local Queue”.



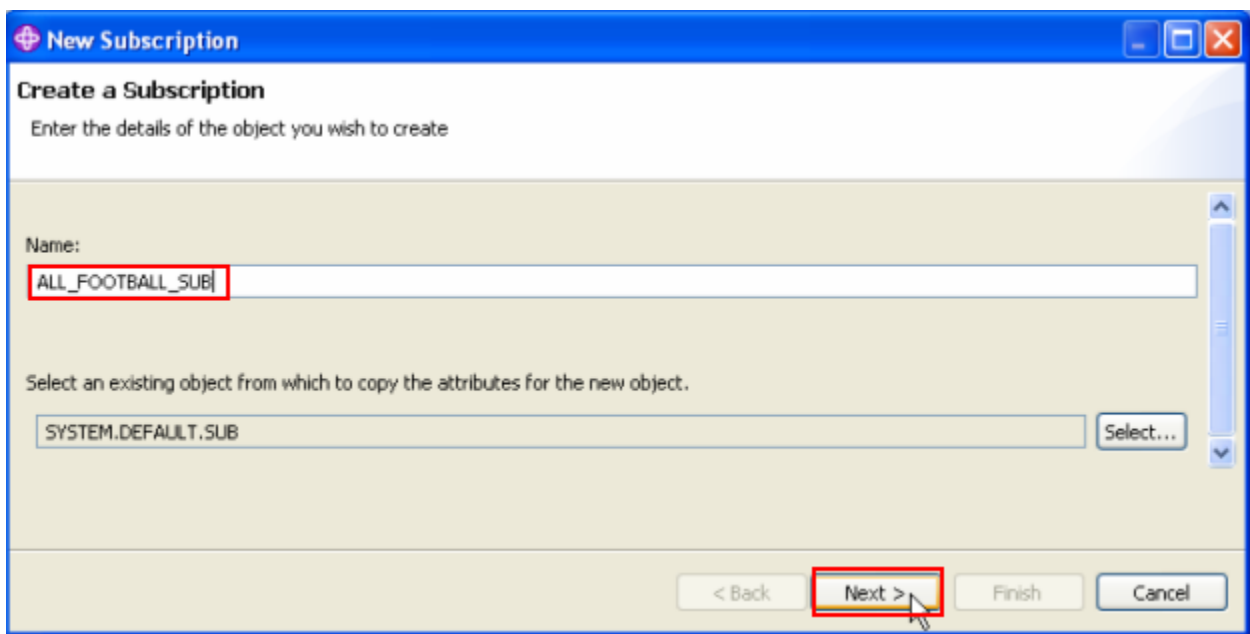
- __2. Name the queue “ALL_FOOTBALL_Q” and press “Finish”.



__3. Select “**Subscriptions**”, right-click and select “**New→Subscription**”.



__4. Type **ALL_FOOTBALL_SUB** as the subscription name and then click “**Next**”.



- __5. 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**

New Subscription

Change properties
Change the properties of the new Subscription

General
Extended

General

Subscription name: ALL_FOOTBALL_SUB

Topic

Topic name: Select...

Topic string:

Wildcard usage:

Scope:

Destination

Destination class:

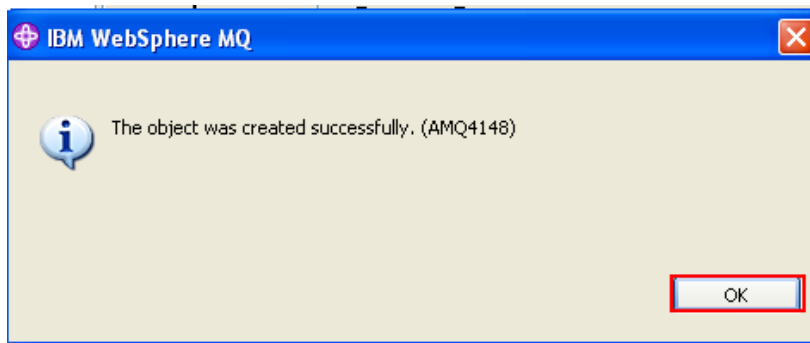
Destination queue manager:

Destination name:

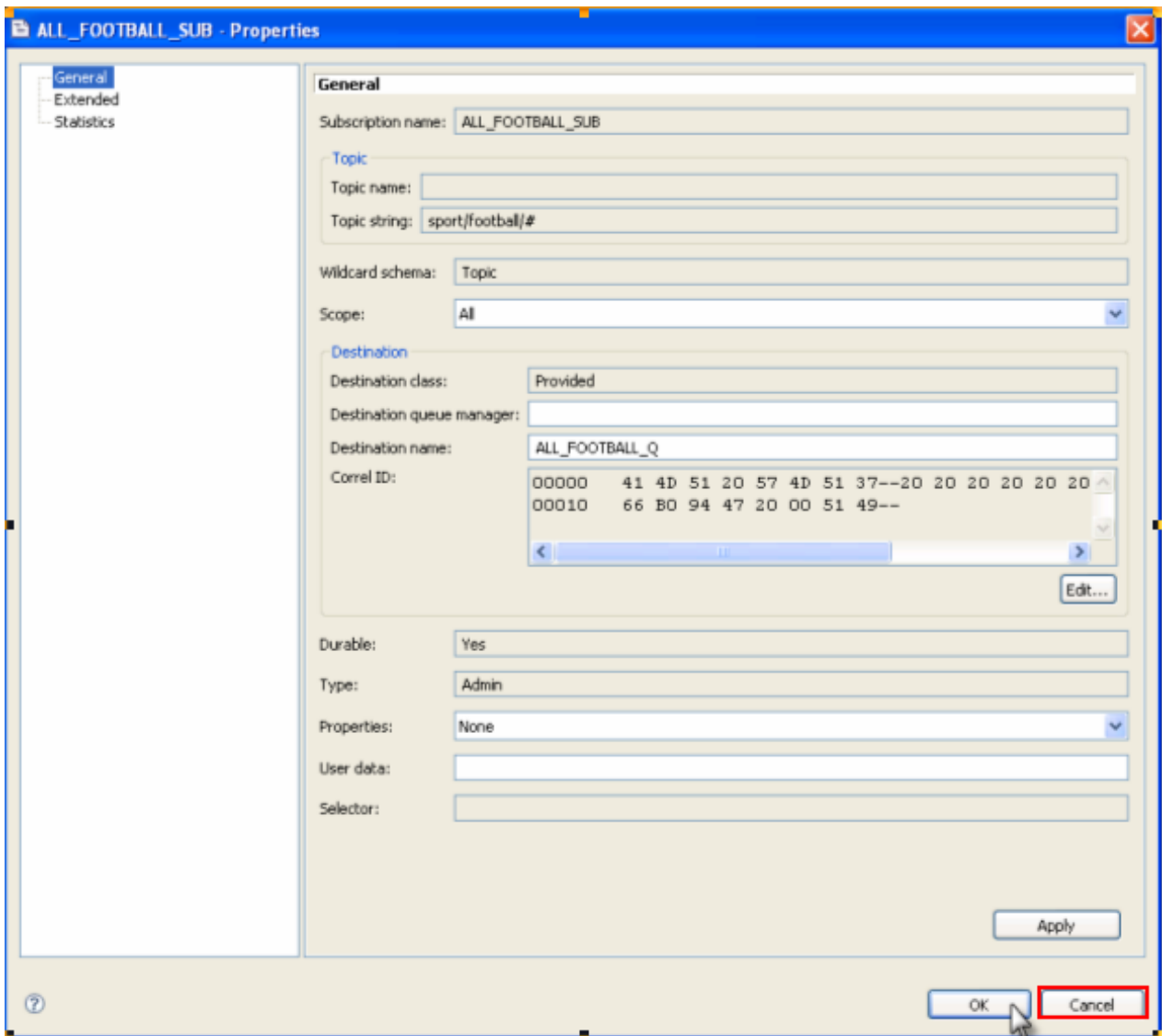
Correlation identifier:
00000 00 00 00 00 00 00 00 00--00 00 00
00010 00 00 00 00 00 00 00 00--

< Back Next > **Finish** Cancel

- __6. Click **OK** to close the confirmation window.



__8. The attributes of the new subscription are displayed.



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

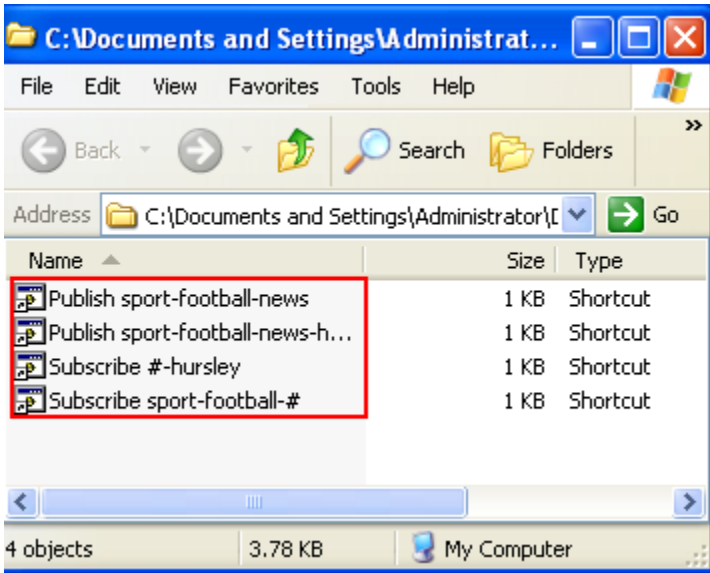
__9. Close the properties window by clicking on **Cancel**.

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.

1. 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.



- __2. 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.

The screenshot shows four terminal windows arranged in a 2x2 grid. The top-left window is titled 'Publish sport-football-news-hursley' and shows the command 'AMQSPUB sport/football/news/hursley' being executed, with output 'Sample AMQSPUBA start' and 'target topic is sport/football/news/hursley'. The top-right window is titled 'Publish sport-football-news' and shows the command 'AMQSPUB sport/football/news' being executed, with output 'Sample AMQSPUBA start' and 'target topic is sport/football/news'. The bottom-left window is titled 'Subscribe sport-football-#' and shows the command 'AMQSSUBW sport/football/#' being executed, with output 'Sample AMQSSUBA start' and 'Calling MQGET : Waiting for published messages'. The bottom-right window is titled 'Subscribe #-hursley' and shows the command 'AMQSSUBW #/hursley' being executed, with output 'Sample AMQSSUBA start' and 'Calling MQGET : Waiting for published messages'.

```
C:\Student\Labs\scripts>AMQSPUB sport/football/news/hursley
Sample AMQSPUBA start
target topic is sport/football/news/hursley
```

```
C:\Student\Labs\scripts>AMQSPUB sport/football/news
Sample AMQSPUBA start
target topic is sport/football/news
```

- __3. 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.

```
C:\ Subscribe sport-football-#
C:\Student\Labs\scripts>AMQSSUBW sport/football/#
Sample AMQSSUBA start
Calling MQGET : Waiting for published messages
message <test message 1>
Calling MQGET : Waiting for published messages
```

```
C:\ Subscribe #-hursley
C:\Student\Labs\scripts>AMQSSUBW #/hursley
Sample AMQSSUBA start
Calling MQGET : Waiting for published messages
message <test message 1>
Calling MQGET : Waiting for published messages
```

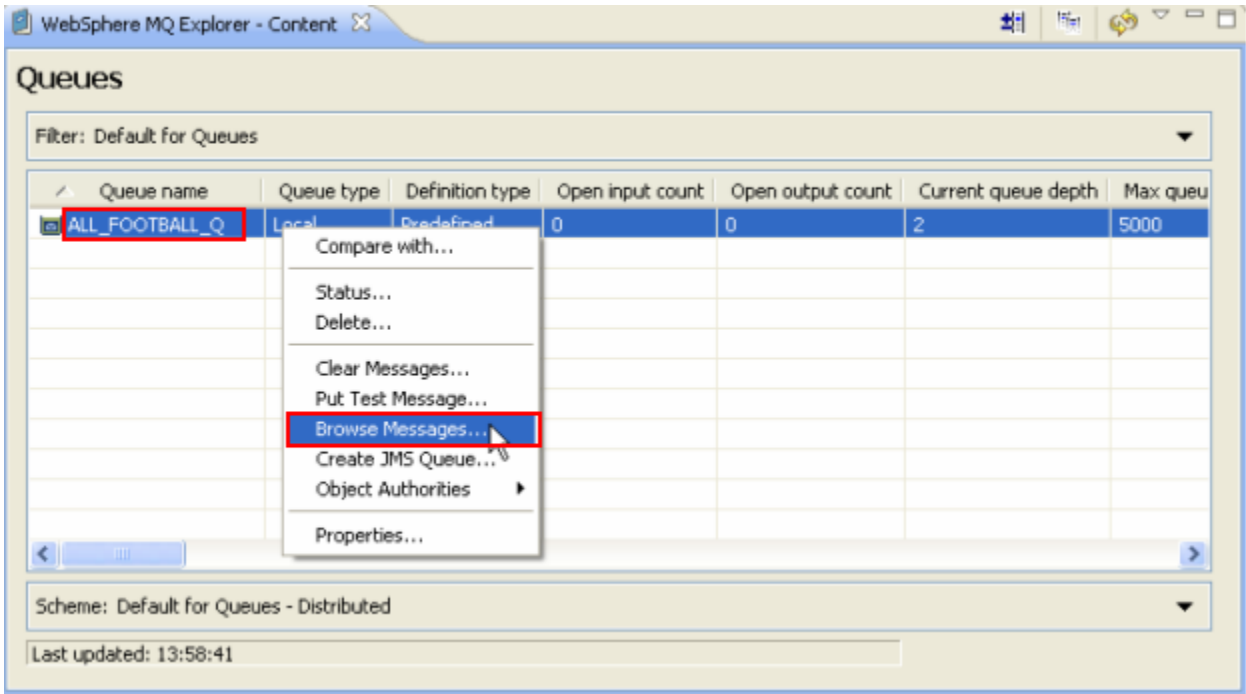
- __4. Again in the top left window type the text “**Hursley News**” and then press “Enter”.
- __5. In the top right window type the text “**Football News**” and press “Enter”. Notice that the **sport/football/#** subscription gets both publications. This is because when you subscribed you used a *multi-level 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.

```
C:\ Subscribe sport-football-#
C:\Student\Labs\scripts>AMQSSUBW sport/football/#
Sample AMQSSUBA start
Calling MQGET : Waiting for published messages
message <test message 1>
Calling MQGET : Waiting for published messages
message <Hursley News>
Calling MQGET : Waiting for published messages
message <Football News>
Calling MQGET : Waiting for published messages
```

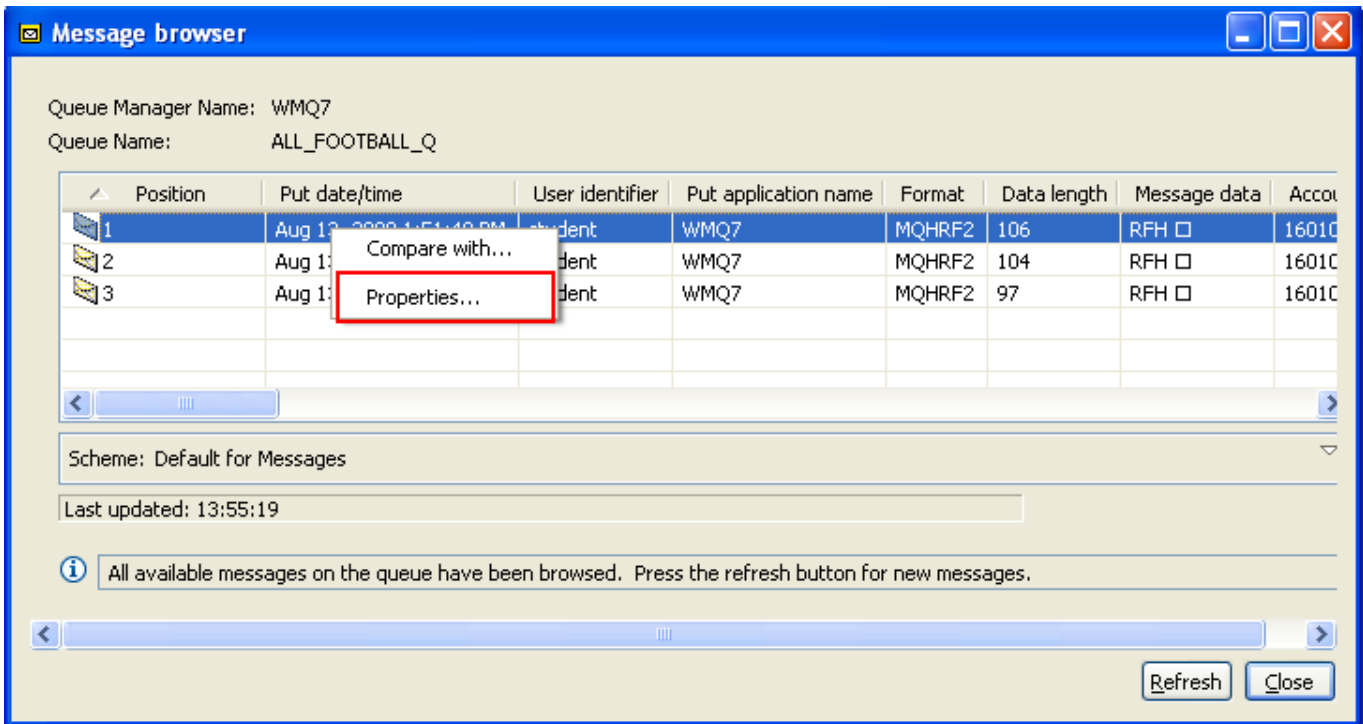
- __6. On the other hand, the **#/hursley** subscription gets only one.

```
C:\ Subscribe #-hursley
C:\Student\Labs\scripts>AMQSSUBW #/hursley
Sample AMQSSUBA start
Calling MQGET : Waiting for published messages
message <test message 1>
Calling MQGET : Waiting for published messages
message <Hursley News>
Calling MQGET : Waiting for published messages
```

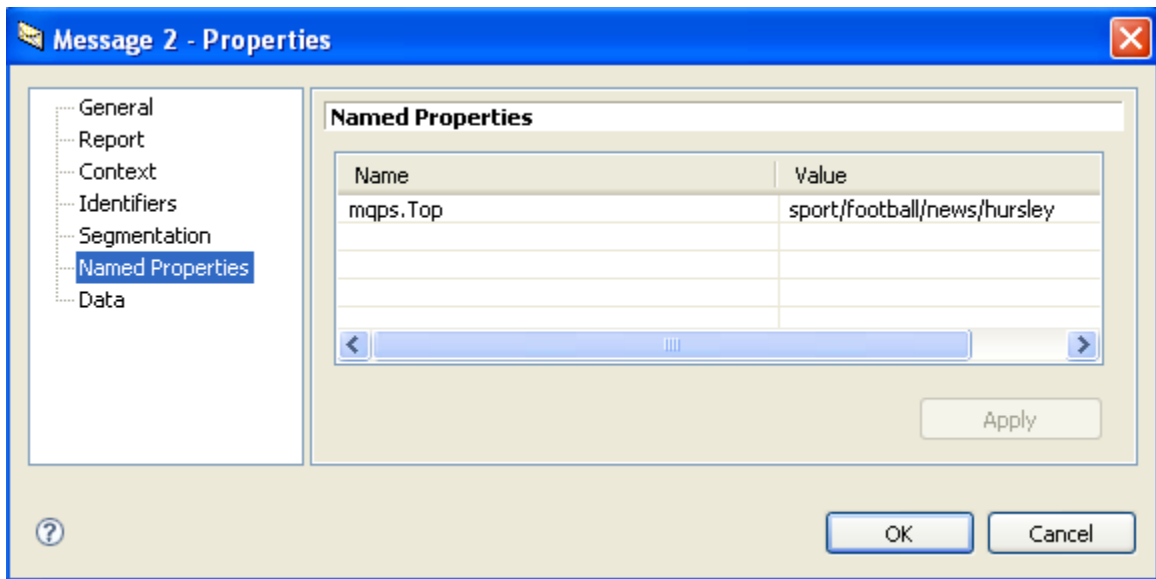

__9. Right-click on the MQ queue **ALL_FOOTBALL_Q** and select **Browse Messages**



__10. You should see three messages on the queue (or as many as you put in the amqsub test). Select one of the messages, right-click and choose **Properties**.



__11. Click on the “**Named Properties**” tab. From this display you can see the originating topic string.



__12. Close the four or five open command windows as you will no longer need them.

This concludes Lab 3.

Lab 4 WebSphere MQ Security lab

4.1 WebSphere MQ Security Lab Overview

This lab will demonstrate WebSphere MQ security administration for point-to-point as well as publish/subscribe MQ objects using the WebSphere MQ Explorer.

4.2 Review the security on the system

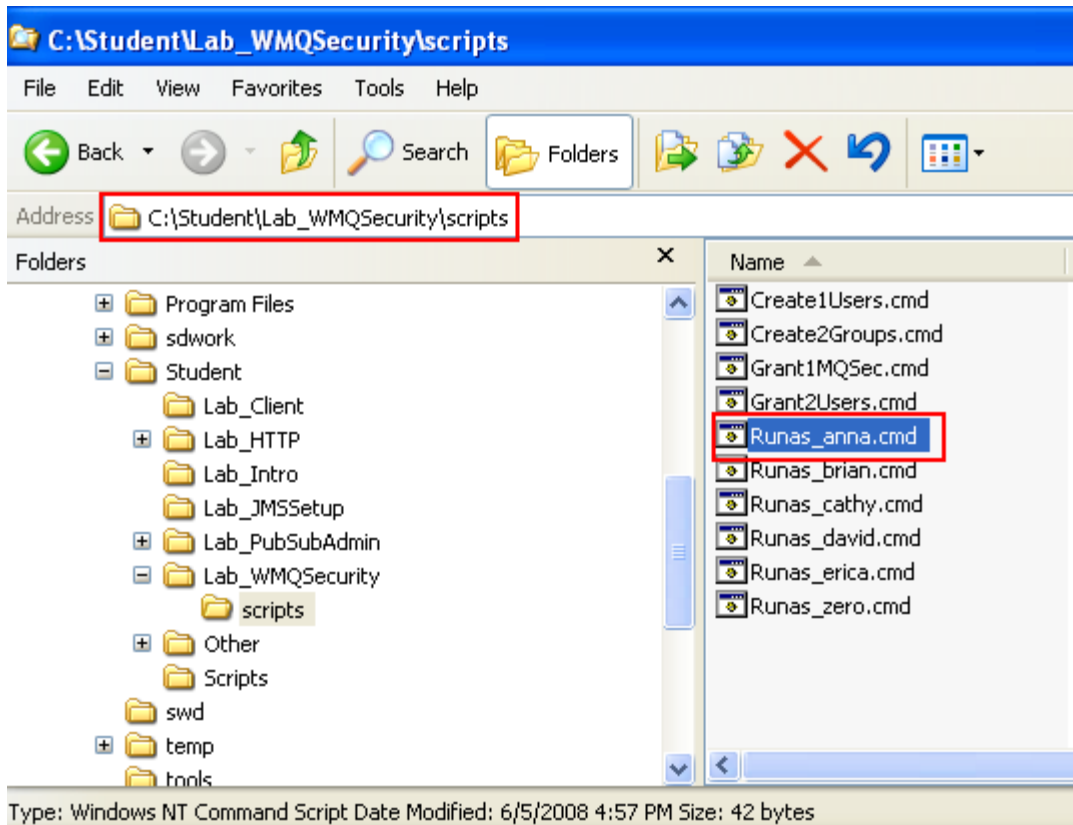
- __1. There are several user ids that are pre-defined on your system. You will be using the user called “**Anna**” and also “**Brian**” to demonstrate some of the security capabilities of WebSphere MQ V7.

Username	Role
Anna	Superuser, member of the mqm group
Brian	Subscriber
Cathy	Result publisher
David	Journalist, can publish to news
Erica	Manager of Hursley football club
Zero	A user without access to the MQ system

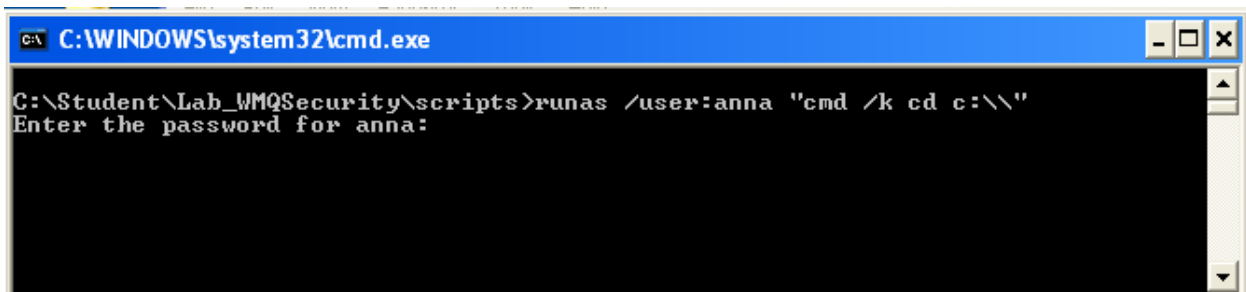
4.3 Using RUNAS to change runtime authority.

- __1. Using the Windows explorer, navigate to the directory **C:\Student\Lab_WMQSecurity\scripts**

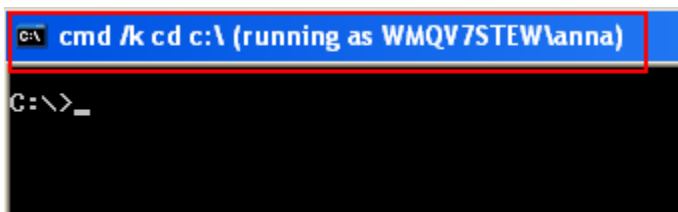
- __2. To run as a different user you will use a program called “runas”. For example, to “runas” user Anna, double-click the script “Runas_anna.cmd”. This will prompt for your password (which is **passw0rd** with a zero (“0”) for the “o”) and open a command window running with that user’s authority.



- __3. Type in the password for Anna which is **passw0rd** (with a zero (“0”) for the “o”)

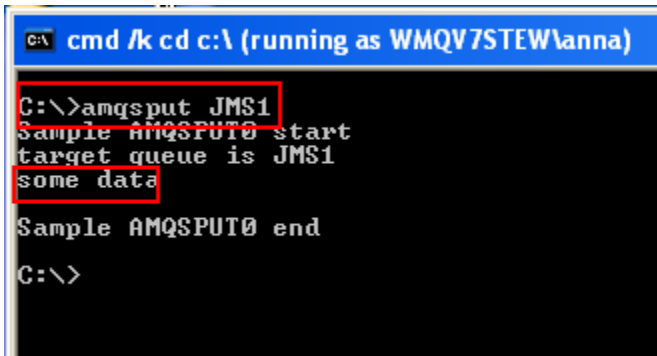


- __4. You are now running as user Anna.
This concludes this portion of Lab 4.



4.4 Point to Point (Queue) Security

- __1. Anna is a “**super user**” because her user id is a member of the **mqm** group. She is able to basically do anything without security stopping her.
- __2. In the runas window of Anna, type in the following command: **amqspout JMS1**. Type in some text, press Enter, and then press the Enter key a second time to end amqspout. Everything should work fine as expected.

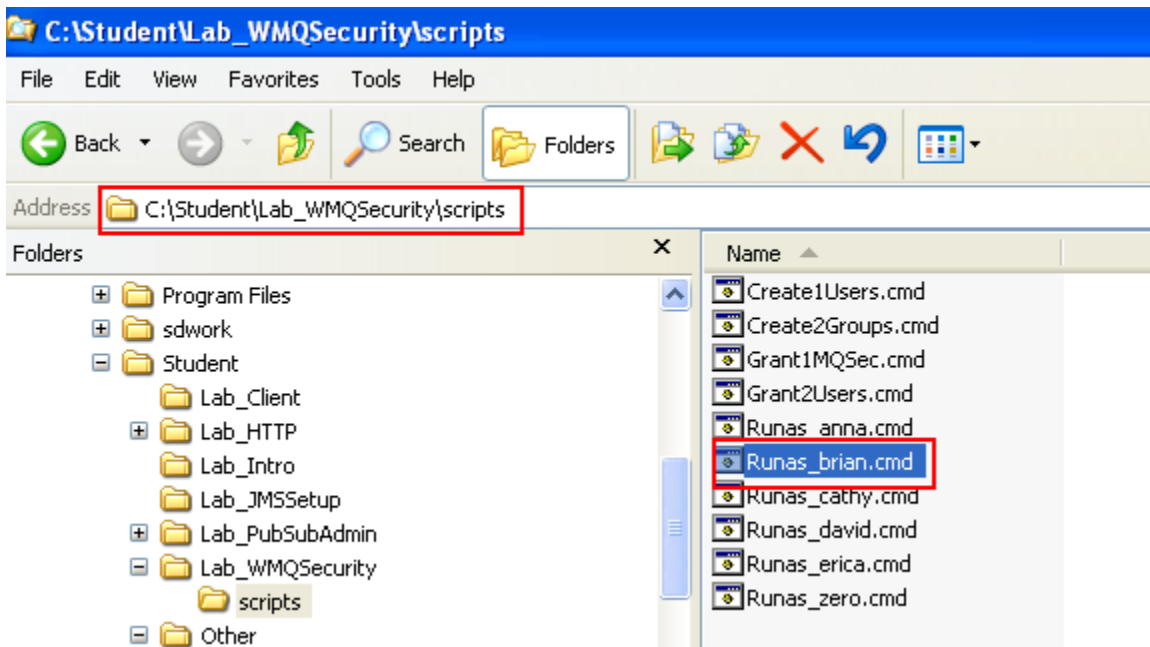


```

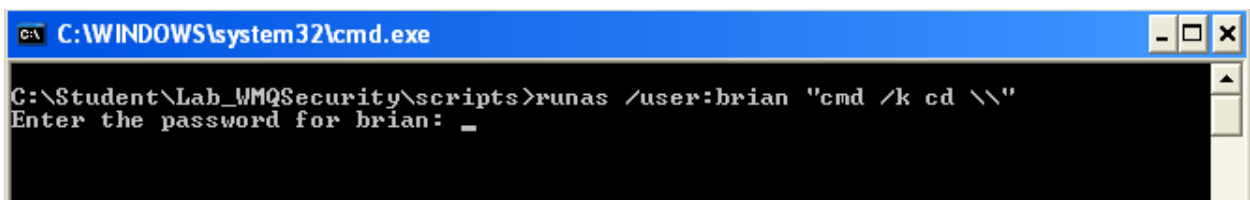
C:\>cmd /k cd c:\ (running as WMQV7STEW\anna)
C:\>amqspout JMS1
Sample AMQSPUT0 start
target queue is JMS1
some data
Sample AMQSPUT0 end
C:\>

```

- __3. Leave the runas Anna window open, you will need it in a later step.
- __4. Find the **Runas_brian.cmd** command and double-click to run it.



- __5. The password is ‘**passw0rd**’.



```

C:\WINDOWS\system32\cmd.exe
C:\Student\Lab_WMQSecurity\scripts>runas /user:brian "cmd /k cd \\'
Enter the password for brian: _

```

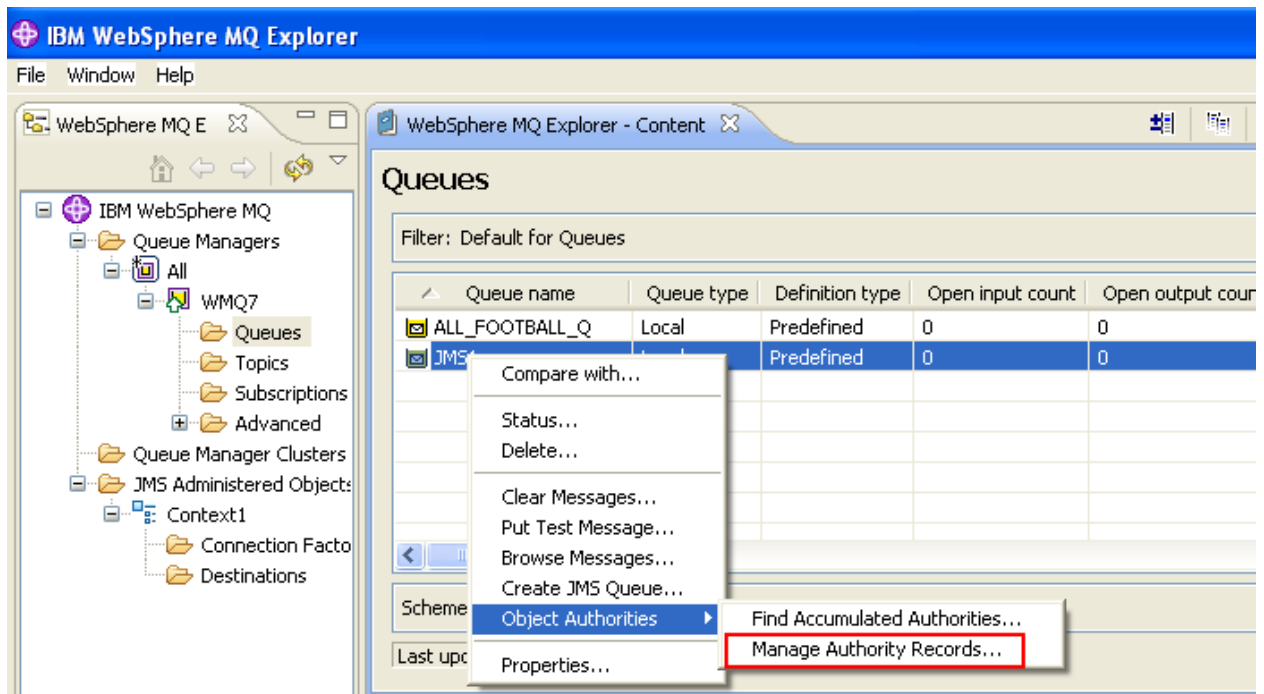
- ___6. Run the **amqsput** program again. (**amqsput JMS1**). User Brian is not authorized to write messages to that MQ queue.

```

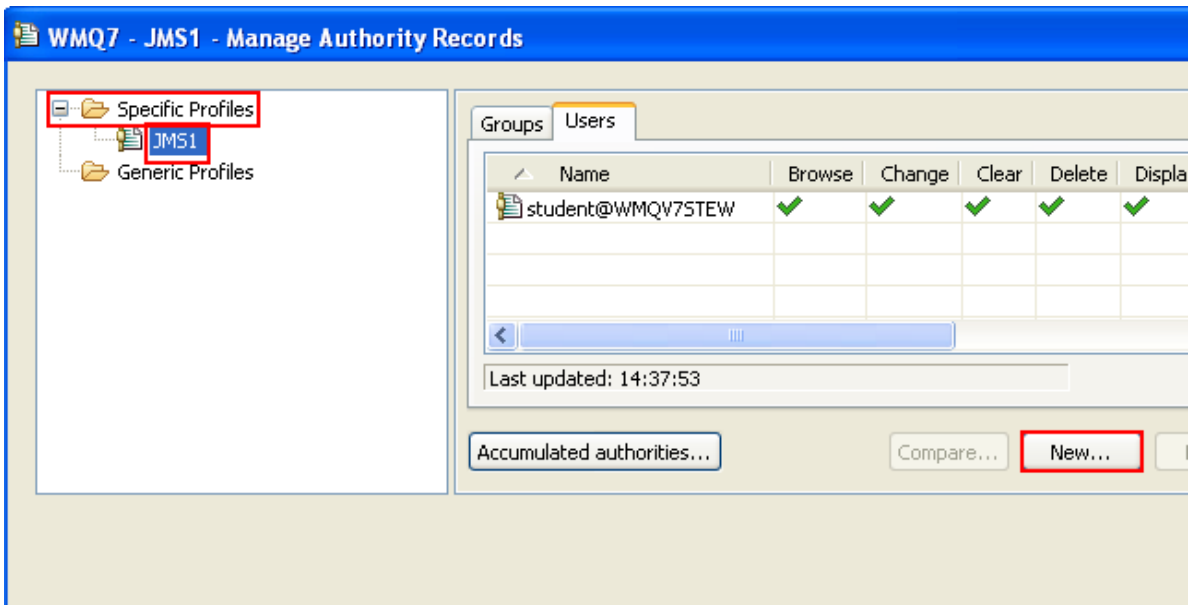
c:\ cmd /k cd \ (running as WMQV7STEW\brian)
C:\> amqsput JMS1
Sample AMQSPUT0 start
target queue is JMS1
MQOPEN ended with reason code 2035
unable to open queue for output
Sample AMQSPUT0 end
C:\>_

```

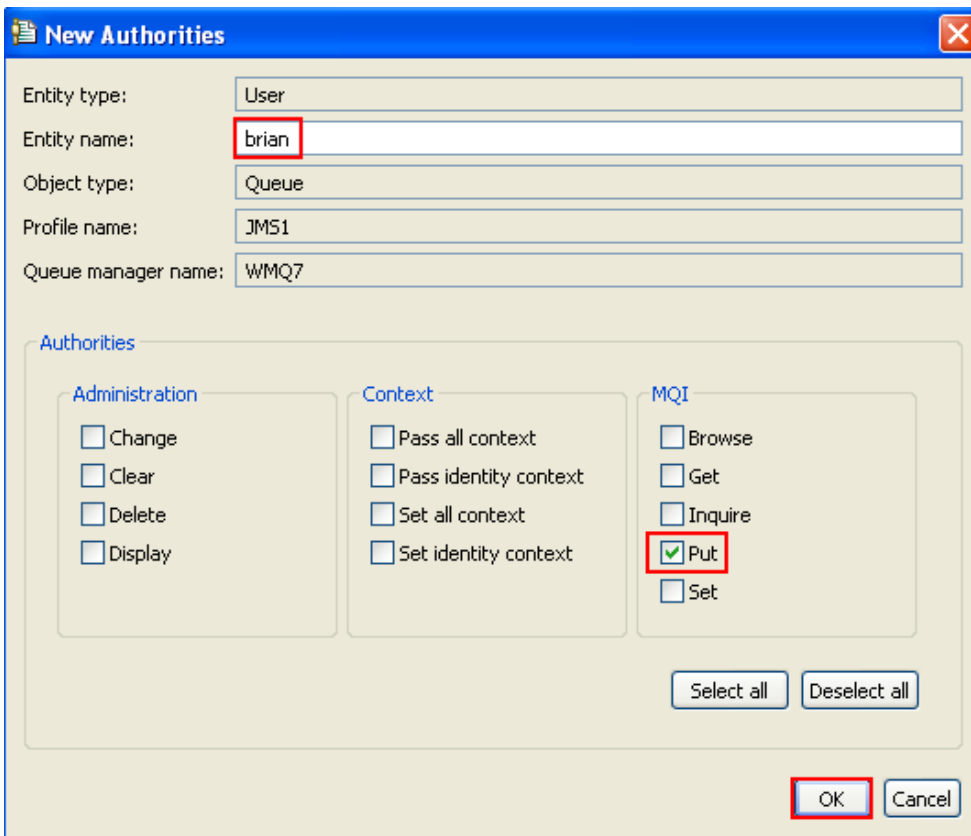
- ___7. There are several methods of administering Object Authority Manager permissions; programmatically, from the command line, or using the MQ Explorer. You will use the MQ Explorer. Open the MQ Explorer and then click on **Queues**. Right-click on the queue called **JMS1**, then on **Object Authorities**→**Manage Authority Records...**



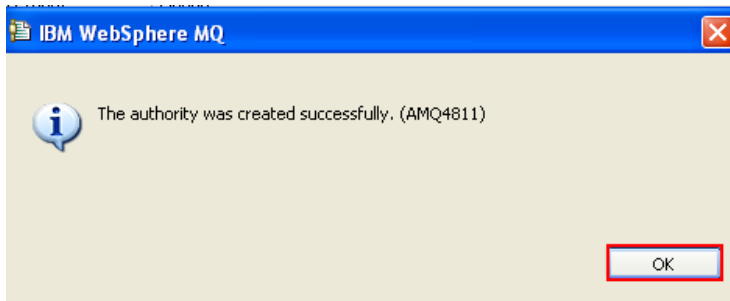
- ___8. Expand **Specific Profiles**, and then click on **JMS1**. Make sure that the **Users** tab is selected. Now click on the **New...** button



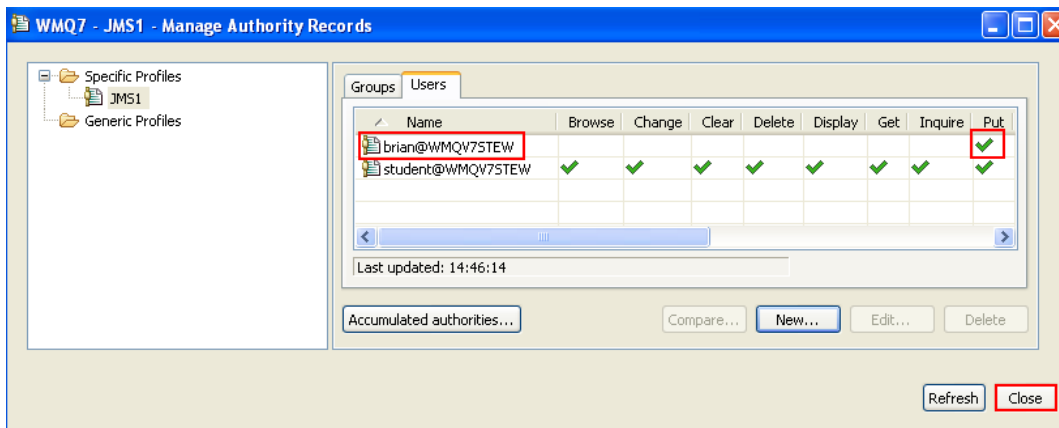
- ___9. You are going to set up a user entry for the user id **brian**. Type **brian** as the Entity Name. Under **Authorities** click next to **Put**. Then click on **OK**.



- ___10. Click on **OK** to close the confirmation window.



- ___11. Notice that a user record has been added with Put authority for user brian. Click on **Close** to close this window.



- ___12. Go back to the runas brian command window and type in **amqspu JMS1**. This time it should be successful. Enter some data; then press Enter again to stop the program.

```

C:\> cmd /k cd \ (running as WMQV7STEW\brian)
C:\> amqspu JMS1
Sample AMQSPUT0 start
target queue is JMS1
this time it works!!
Sample AMQSPUT0 end
C:\> _

```

- __13. You gave the user id brian the ability to put messages to queues, but not to read or browse messages. In the runas brian command window type in **amqsget JMS1**. This should fail with a return code 2035, **MQRC_NOT_AUTHORIZED**.

```

C:\ cmd /k cd \ (running as WMQV7STEW\brian)

C:\>amqsput JMS1
Sample AMQSPUT0 start
target queue is JMS1
this time it works!!

Sample AMQSPUT0 end

C:\>amqsget JMS1
Sample AMQSGET0 start
MQOPEN ended with reason code 2035
unable to open queue for input
Sample AMQSGET0 end

C:\>_

```

This concludes this portion of Lab 4.

4.5 Pub/Sub (Topic) Security

- __1. You are going to do the same tests using Pub/Sub security to secure topics. As a test, try to publish to topic *sport/football/results/hursley*. In the “runas” window for Anna, type **amqspub sport/football/results/hursley** and press Enter. This should be successful, as Anna is a “superuser”.

```

C:\ cmd /k cd c:\ (running as WMQV7STEW\anna) - amqspub sport/f

C:\>amqspub sport/football/results/hursley
Sample AMQSPUBA start
target topic is sport/football/results/hursley

```

- __2. Press Enter to end the amqspub program. Then type in **Exit** to close the runas Anna window.

```

C:\ cmd /k cd c:\ (running as WMQV7STEW\anna)

C:\>amqspub sport/football/results/hursley
Sample AMQSPUBA start
target topic is sport/football/results/hursley

Sample AMQSPUBA end

C:\>exit

```

- ___3. Now try a userid that is not authorized to publish to a topic. Return to the **Runas_brian.cmd**. Again, try to publish to topic *sport/football/results/hursley*. In the window type **amqspub sport/football/results/hursley** and press Enter. This should end with a 2035 return code. Type in the command **mqrc 2035** and you should see that this is a security error.

```

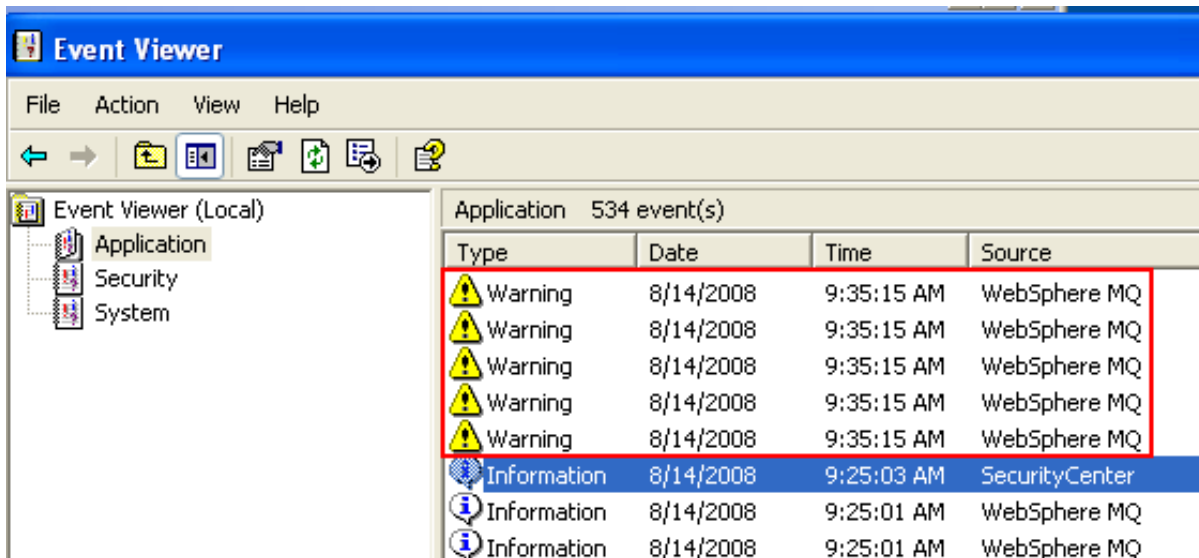
C:\>cmd /k cd \ (running as WMQV7STEW\brian)
C:\>amqspub sport/football/results/hursley
Sample AMQSPUBA start
target topic is sport/football/results/hursley
MQOPEN ended with reason code 2035
unable to open topic for publish
Sample AMQSPUBA end
C:\>mqrc 2035
2035 0x0000007f3 MQRC_NOT_AUTHORIZED
C:\>

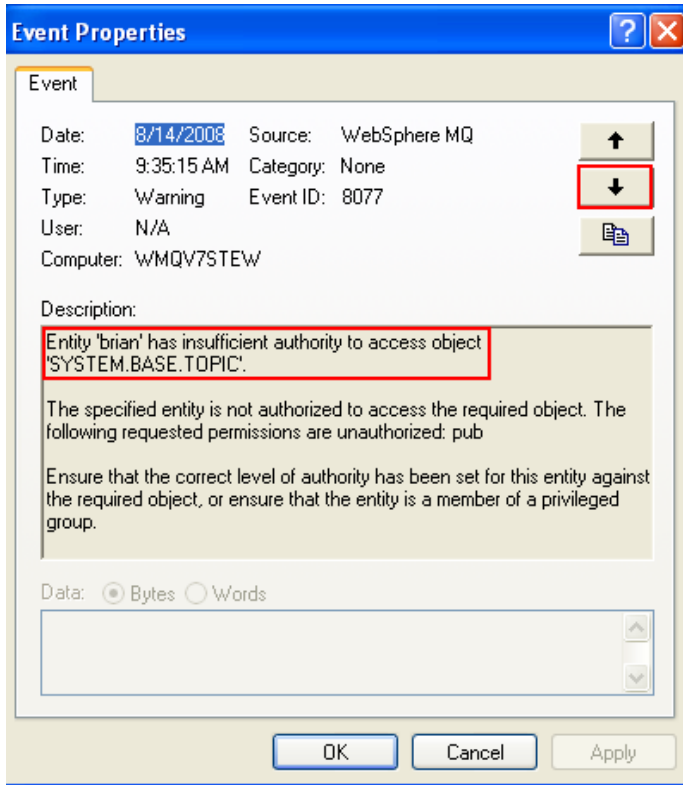
```

- ___4. You will see more interesting information in the Windows event viewer. To open the event viewer find the shortcut in the Start Menu.

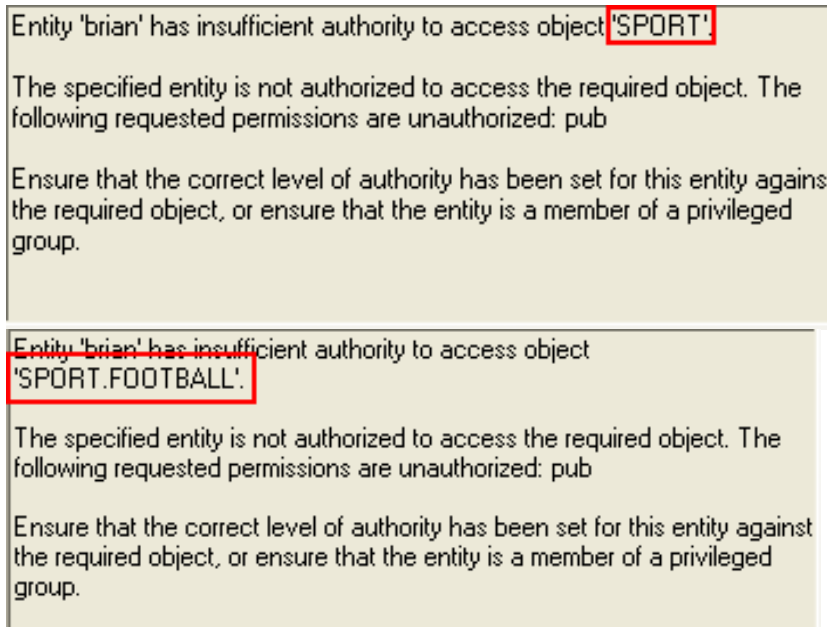


- ___5. You will see a number of warnings produced because of the 2035 return code. Double-click on the first warning to display its contents





- __6. **SYSTEM.BASE.TOPIC** is the root topic object for the topic tree as a whole. Click on the down arrow to browse through the remaining warnings.



Entity 'brian' has insufficient authority to access object 'SPORT.FOOTBALL.RESULTS'.

The specified entity is not authorized to access the required object. The following requested permissions are unauthorized: pub

Ensure that the correct level of authority has been set for this entity against the required object, or ensure that the entity is a member of a privileged group.

Entity 'brian' has insufficient authority to access object 'SPORT.FOOTBALL.RESULTS.HURSLEY'.

The specified entity is not authorized to access the required object. The following requested permissions are unauthorized: pub

Ensure that the correct level of authority has been set for this entity against the required object, or ensure that the entity is a member of a privileged group.

This concludes this portion of Lab 4.

4.6 Using the MQ Explorer to manage security

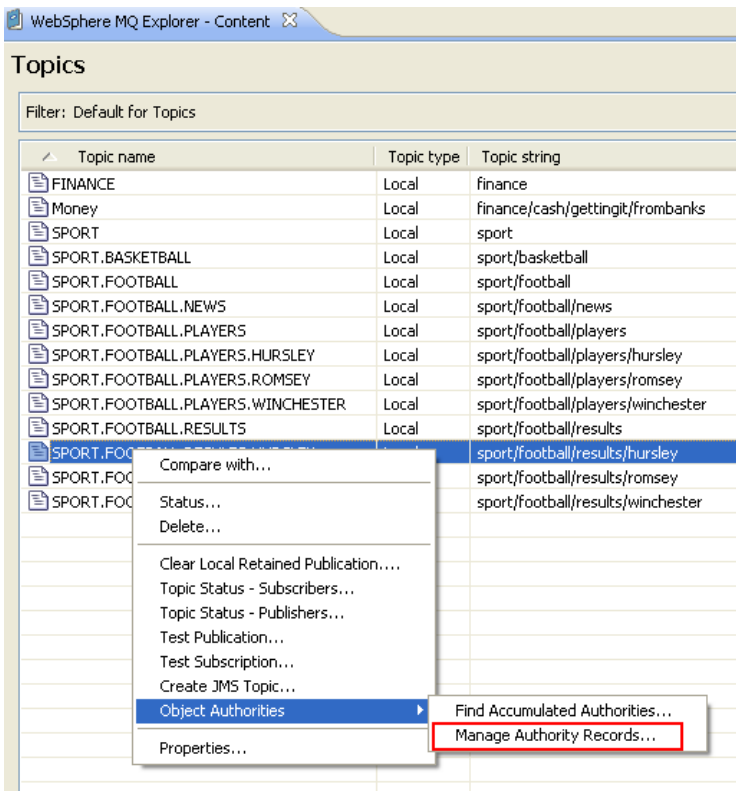
User Brian needs access to this topic! So in the next section you will use the MQ Explorer to create the permissions that Brian needs to access the topic **sport/football/results/hursley**.

__1. Open the MQ Explorer and click on **Topics**

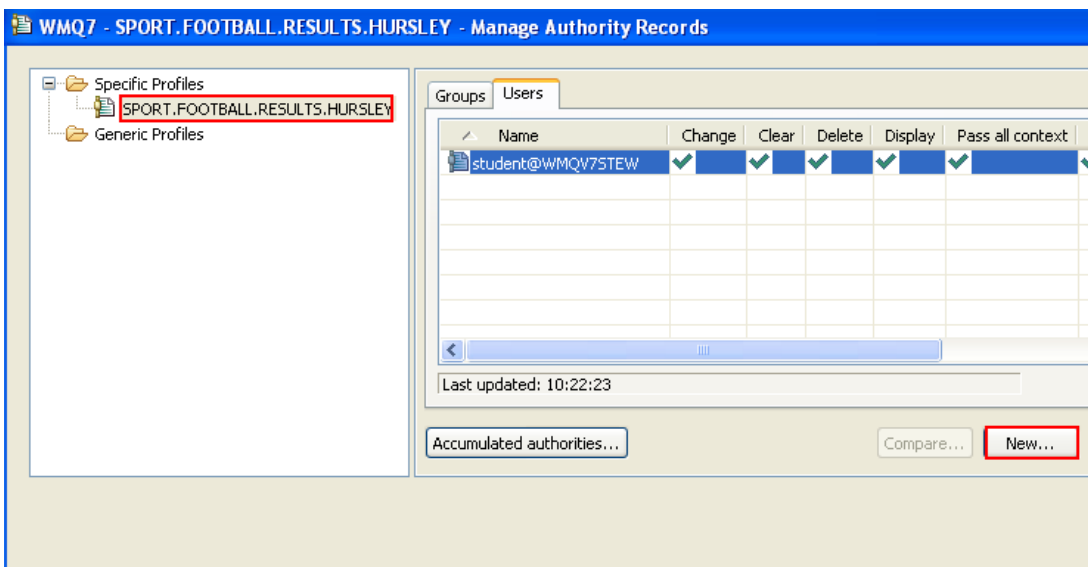
The screenshot shows the IBM WebSphere MQ Explorer interface. The left pane displays a tree view of the MQ environment, with 'Topics' selected under 'WMQ7'. The right pane shows a table of topics. The topic 'SPORT.FOOTBALL.RESULTS.HURSLEY' is highlighted with a red box.

Topic name	Topic type	Topic string
FINANCE	Local	finance
Money	Local	finance/cash/gettingit/frombanks
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

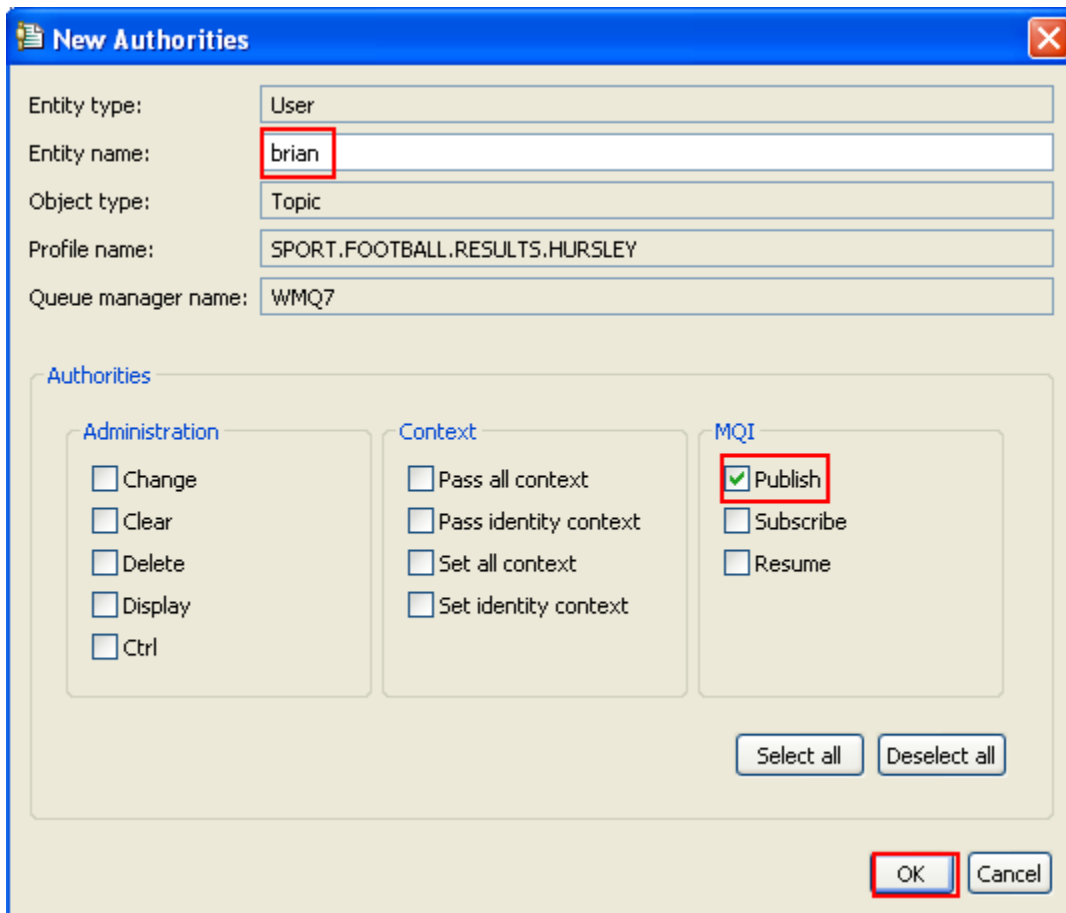
- ___2. Right-click on the topic name **SPORT.FOOTBALL.RESULTS.HURSLEY**. Select **Object Authorities**→**Manage Authority Records...**



- ___3. A new window will open showing groups and users for the specific profile. Click on the Specific Profile **SPORT.FOOTBALL.RESULTS.HURSLEY**. Now click on the **Users** tab. Finally, click on the **New...** button. You are going to create a user record for the user id brian.



- __4. Enter **brian** as the user name. Under **Authorities**→**MQI** click on the **Publish** checkbox. Then click on **OK**



The screenshot shows the 'New Authorities' dialog box. The fields are filled as follows:

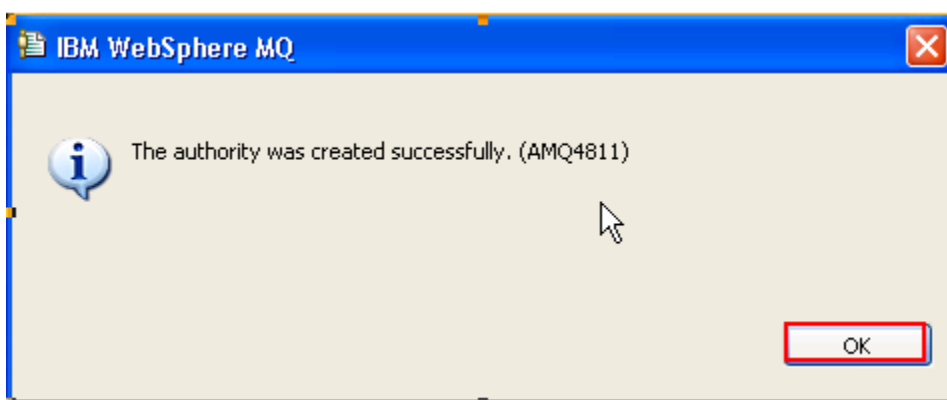
- Entity type: User
- Entity name: brian
- Object type: Topic
- Profile name: SPORT.FOOTBALL.RESULTS.HURSLEY
- Queue manager name: WMQ7

The 'Authorities' section contains three panels:

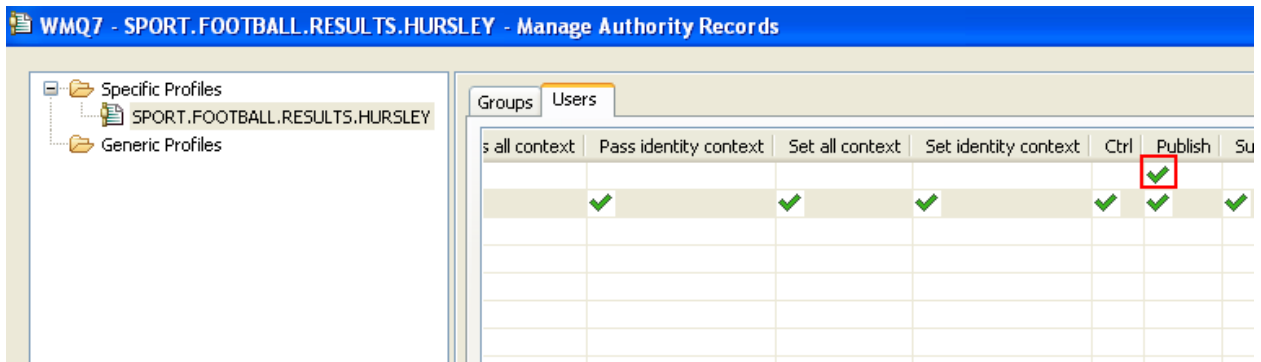
- Administration:** Change, Clear, Delete, Display, Ctrl (all unchecked)
- Context:** Pass all context, Pass identity context, Set all context, Set identity context (all unchecked)
- MQI:** Publish (checked), Subscribe, Resume (all unchecked)

Buttons at the bottom include 'Select all', 'Deselect all', 'OK', and 'Cancel'. The 'OK' button is highlighted with a red box.

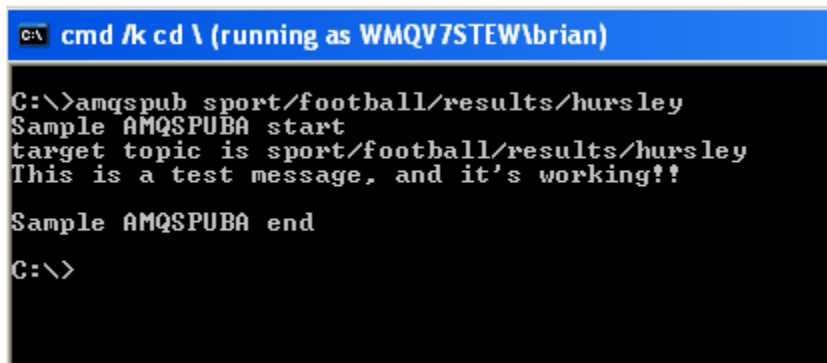
- __5. Click OK to dismiss the confirmation box.



- __6. You have now built a security tree where Brian is granted authority to publish football results. Scroll to the right to see the green checkmark under the Publish column.



- __7. Test that Brian can publish to the **sport/football/results/hursley** topic (double-click on the **Runas_brian.cmd** script, enter the **password** then enter the command **amqspub sport/football/results/hursley** and type a test message to the Topic



This concludes the Publish/Subscribe lab.

Lab 5 WebSphere MQ HTTP Bridge

In this lab you will be introduced to the WebSphere MQ – HTTP Bridge. You will exercise some of the features of the Bridge and see how easy it is to enable browser-based applications to access MQ resources.

This lab does use some external tools that are not shipped as part of WebSphere MQ. For example, you will employ a standalone HTTP Listener program available from IBM® as a SupportPac™ (SupportPac MA94) – this is a WebSphere MQ product extension, but is not shipped with the product. You will also use cURL. cURL is a freeware tool that provides a command line facility for transferring files using URL syntax. The strong point of cURL is the number of data transfer protocols it supports. You will use it to submit HTTP requests as part of this lab.

In this lab you will:

- Start the stand-alone HTTP Listener program.
- Use cURL to send HTTP requests to the listener.
- See the options and responses available to the listener.
- Use a sample JavaScript program to put and get messages, illustrating a zero footprint MQ client.

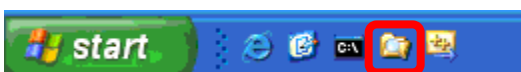
The http-listener program and all other resources needed in the lab are supplied in the c:\Student\Lab_HTTP directory.

This lab assumes that queue manager WMQ7 exists. You were directed to create this during Lab 1.

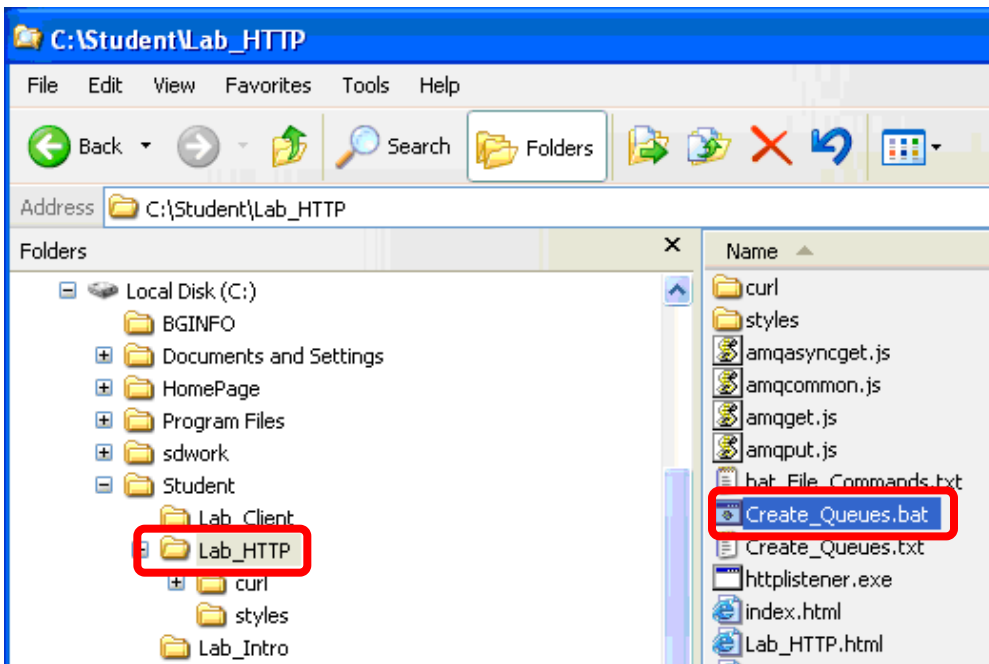
5.1 Create the Required Queues

In Lab 1 you used the MQ Explorer to create some queues. In this lab you will use a command script file for that task so you can see another option that can be used to do this administrative work. This approach uses an MQ utility called **runmqsc**. This program will take a series of MQ administrative commands and execute them. Those commands can come from the keyboard or from a file. In this case a file will be used.

- ___1. Launch **Windows Explorer** using the icon in the system tray (if you have Windows Explorer already launched then bring it into active view).

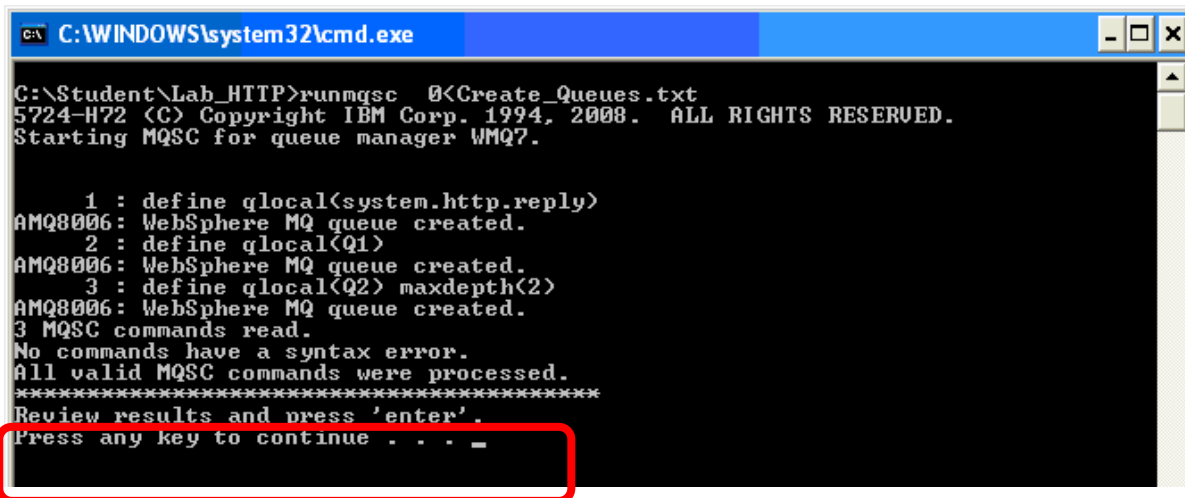


__2. Navigate to **C:\Student\Lab_HTTP** and double-click on the **Create_Queues.bat** file.



__3. The following command is issued: **runmqsc < Create_Queues.txt** The “<” indicates that the commands will come from a file and that the path to that file follows. You don’t have to enter a queue manager name for this command because you specified that WMQ7 was the default queue manager on this system when you created it. If that were not the case the name of the queue manager would follow runmqsc immediately (i.e. runmqsc WMQ7 < Create_Queues.txt).

Below you see the results from executing the command....hit the Enter key after reviewing.

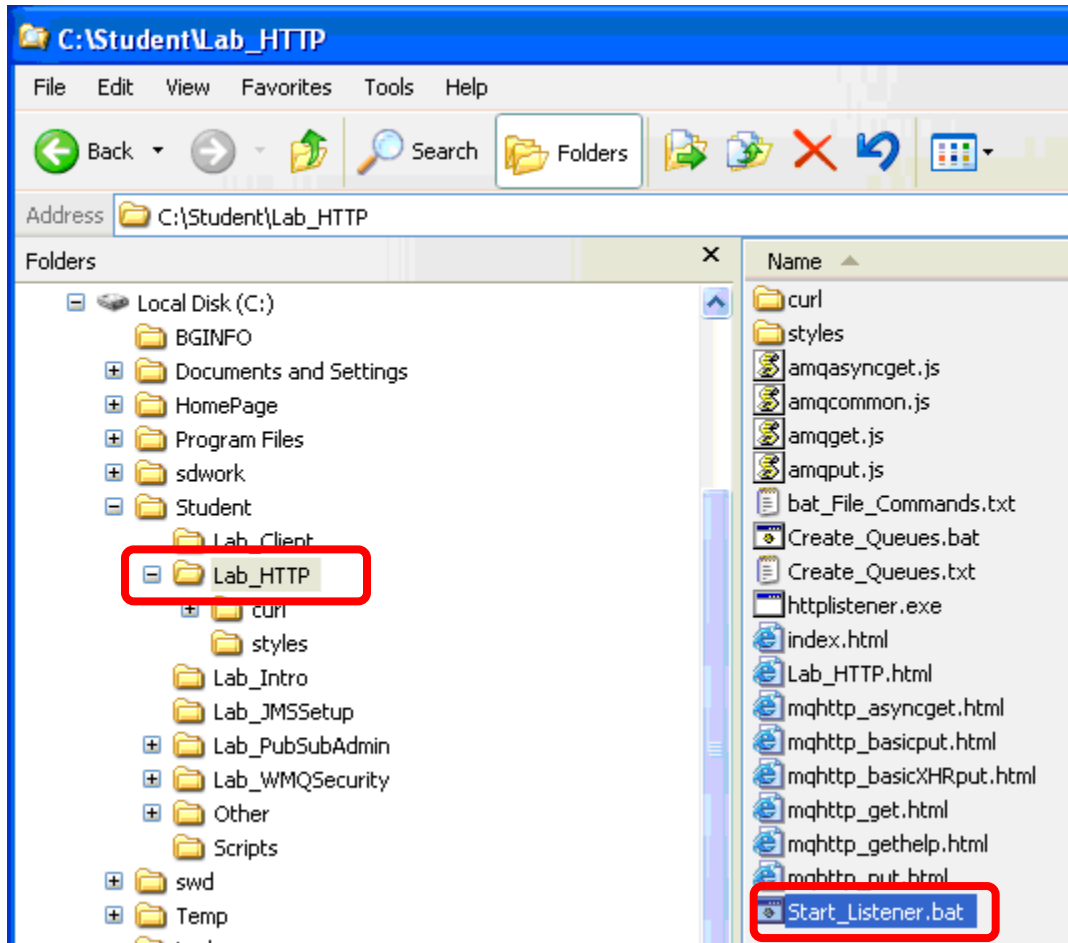


Note that the **Q2** queue has been assigned a **maximum queue depth of 2**.

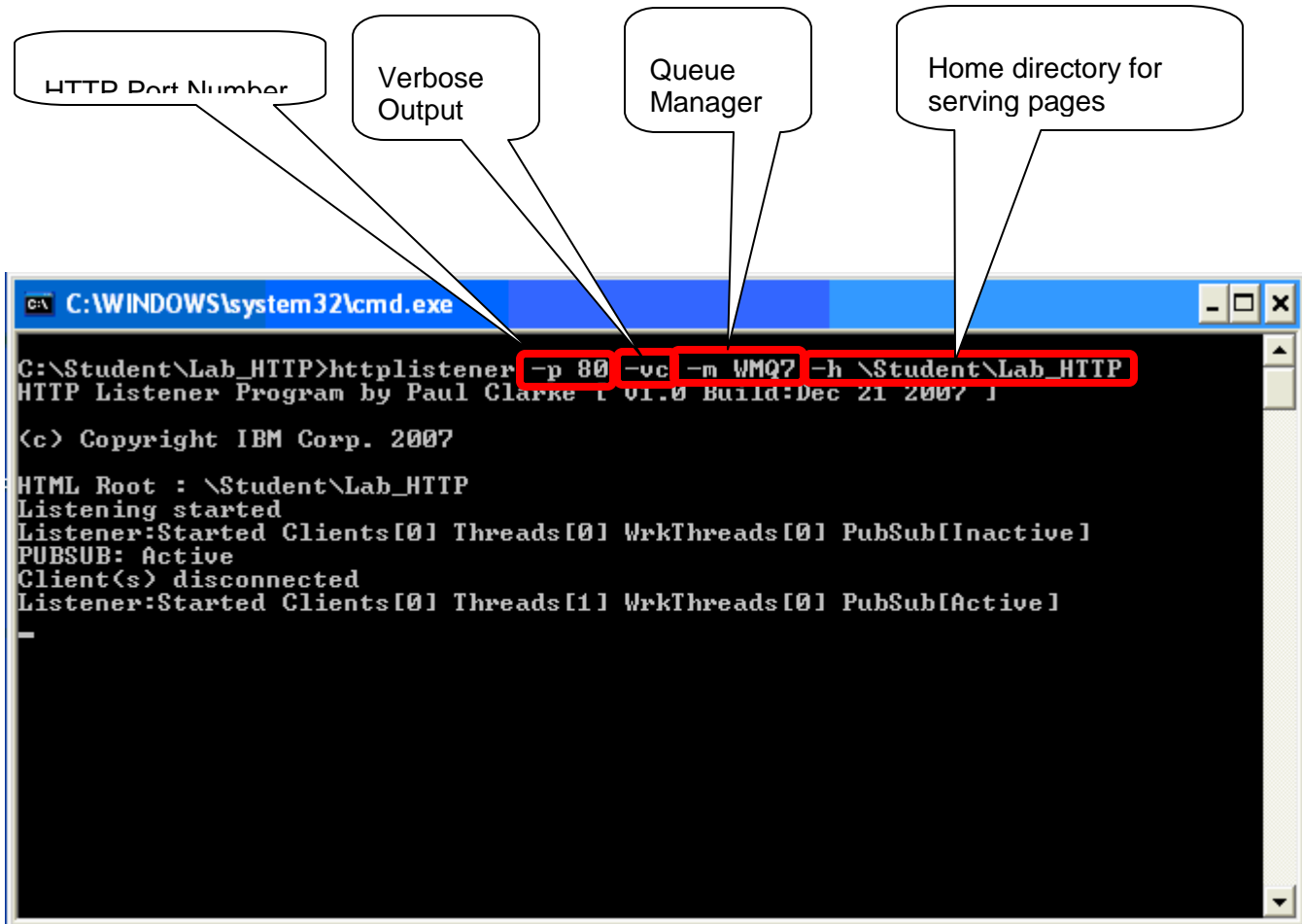
This concludes this portion of Lab 5.

5.2 Start The HTTP Listener

1. Double click on the **Start_Listener.bat** file.



__2. The following command is executed.....review its output. Note that this command window will stay open since the MQ HTTP Listener is going to continue to run in this session.



__3. This will start a listener on HTTP port 80, connecting to queue manager WQM7. The root for serving web pages will be \Student\Lab_HTTP...

Do not close this command window!

- ___4. Verify that the listener is running by **opening a web browser** and pointing it to <http://localhost>. You should see the web page shown below.

WebSphere MQ AJAX Messaging Demo - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Back Forward Stop Home Search Favorites Refresh Print Mail

Address <http://localhost/>

Google Go Bookmarks 0 blocked Check AutoLink AutoFill Send to

WebSphere MQ HTTP Demo

Home

WebSphere MQ HTTP AJAX Samples

Put message

Get Message

Async Gets

Basic Put

Basic AJAX Put

Help

These web pages demonstrate how messages can be sent to and received from WebSphere MQ Queues and Topics using HTML, Javascript, and the WMQ HTTP API. The messaging takes place without the need to install any WMQ software or browser plugins on the client. These samples have been tested on Internet Explorer and Firefox they should work on other browsers, but please let us know if you have problems.

AJAX Demonstrations

The [AJAX Put Message Demo](#) shows how messages can be sent to a WMQ Queue or Topic. It allows the queue/topic name to be entered, along with request message headers and message text, and then displays the status of the HTTP POST request.

The [AJAX Get Message Demo](#) shows how messages can be retrieved - one at a time - from a given WMQ Queue or Topic. It allows the queue/topic name and request message headers to be entered, and then displays the status of the HTTP DELETE request, along with the message headers and body (if a message was successfully retrieved).

The [AJAX Asynchronous Get Sample](#) also shows how messages can be retrieved from a given WMQ Queue or Topic. Unlike the Get Message Sample described above, this application keeps polling the destination with HTTP DELETE requests and displays the retrieved messages in a list as they arrive.

Basic Samples

The [Basic Put Sample](#) contains a very simple HTML form which puts a multi-part message to SYSTEM.DEFAULT.LOCAL.QUEUE when a button is clicked. It demonstrates a minimal WMQ HTTP client and should work in browsers which do not support Javascript.

The [Basic AJAX Put Sample](#) contains a very simple JavaScript AJAX program which uses XMLHttpRequest to put a message to SYSTEM.DEFAULT.LOCAL.QUEUE when a button is clicked. It demonstrates a minimal WMQ HTTP AJAX client and provides a good basis for building more complex AJAX applications.

WMQ HTTP Listener SupportPacs

The WMQ HTTP API server code is available in two SupportPacs - a servlet-based implementation which can plug into an JEE existing application server and a stand-alone 'Native' implementation. For more information, see:

- ___5. This indicates the listener is running. Close the browser.

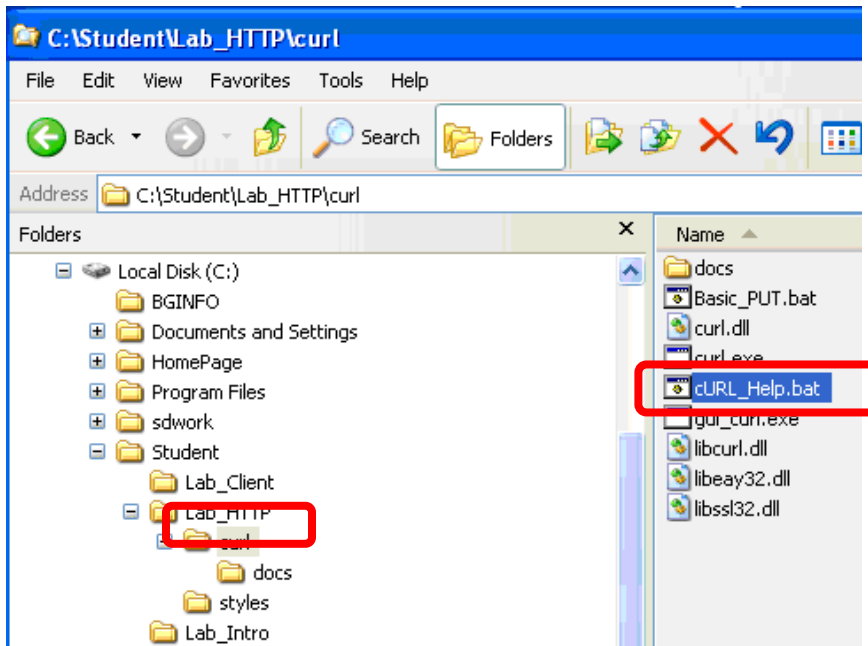
This concludes this portion of Lab 5.

5.3 Test the Listener Using cURL

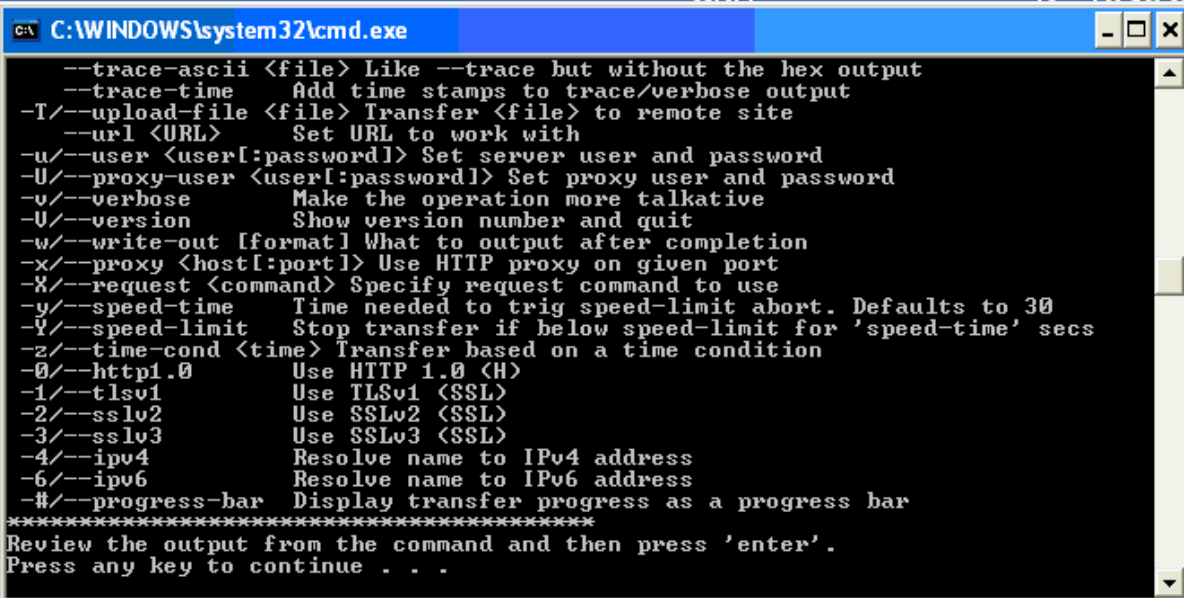
In this section you use the cURL program to send HTTP requests to the listener and examine the responses.

5.4 Posting Messages Using HTTP Post

- __1. Navigate to the **cURL** folder and double click on **cURL_Help.bat**....



- __2. The following command is run. **Examine the output** from the command and then hit the Enter key.....



```

C:\WINDOWS\system32\cmd.exe

--trace-ascii <file> Like --trace but without the hex output
--trace-time      Add time stamps to trace/verbose output
-T/--upload-file <file> Transfer <file> to remote site
--url <URL>       Set URL to work with
-u/--user <user[:password]> Set server user and password
-U/--proxy-user <user[:password]> Set proxy user and password
-v/--verbose      Make the operation more talkative
-U/--version      Show version number and quit
-w/--write-out [format] What to output after completion
-x/--proxy <host[:port]> Use HTTP proxy on given port
-X/--request <command> Specify request command to use
-y/--speed-time  Time needed to trig speed-limit abort. Defaults to 30
-Y/--speed-limit Stop transfer if below speed-limit for 'speed-time' secs
-z/--time-cond <time> Transfer based on a time condition
-0/--http1.0     Use HTTP 1.0 (H)
-1/--tlsv1       Use TLSv1 (SSL)
-2/--sslv2       Use SSLv2 (SSL)
-3/--sslv3       Use SSLv3 (SSL)
-4/--ipv4        Resolve name to IPv4 address
-6/--ipv6        Resolve name to IPv6 address
-#/--progress-bar Display transfer progress as a progress bar
*****
Review the output from the command and then press 'enter'.
Press any key to continue . . .

```

- __3. You will use only a few of the options and these are shown below:

```
curl --help
```

```
Usage: curl [options...] <url>
```

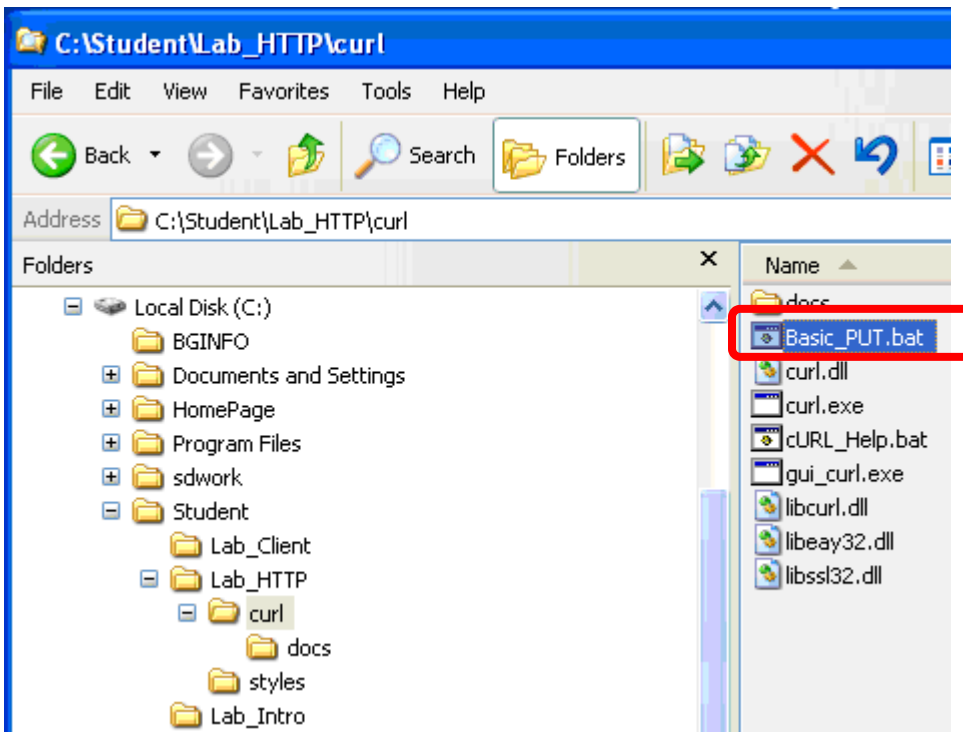
```
-d/--data <data>    HTTP POST data (H)
```

```
-H/--header <line> Custom header to pass to server (H)
```

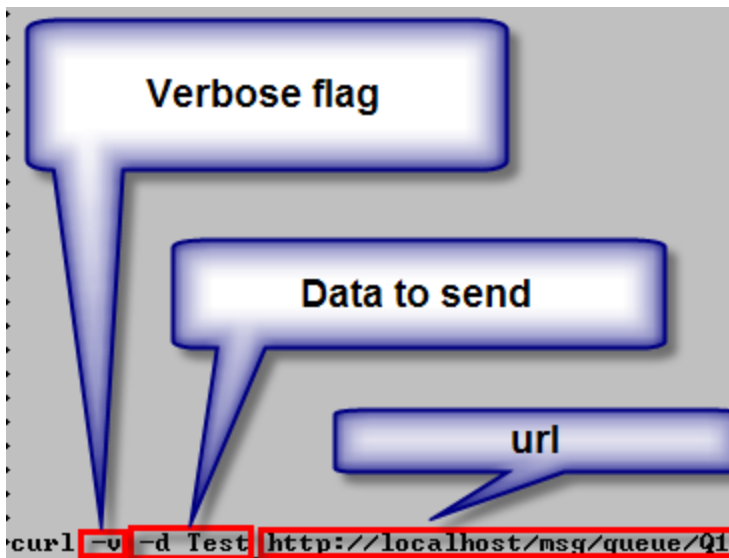
```
-v/--verbose        Make the operation more talkative
```

5.5 A Very Basic PUT

__1. Double click on the Basic_PUT.bat file.....



__2. The following command will be run.



- __3. Examine its output then hit the Enter key.

```

C:\WINDOWS\system32\cmd.exe
C:\Student\Lab_HTTP\curl>curl -v -d Test http://localhost/msg/queue/Q1
* About to connect() to localhost port 80 (#0)
* Trying 127.0.0.1... connected
* Connected to localhost (127.0.0.1) port 80 (#0)
> POST /msg/queue/Q1 HTTP/1.1
User-Agent: curl/7.16.4 (i586-mingw32msvc) libcurl/7.16.4 OpenSSL/0.9.8e zlib/1.2.2
Host: localhost
Accept: */*
Content-Length: 4
Content-Type: application/json
HTTP/1.1 200 OK
Content-Length: 0
Cache-Control: no-cache
* Connection #0 to host localhost left intact
* Closing connection #0
*****
Review the output from the command and then press 'enter'.
Press any key to continue . . .
  
```

- __4. Bring the **MQ Explorer** into active view. Note that you now have a message on **Q1**.

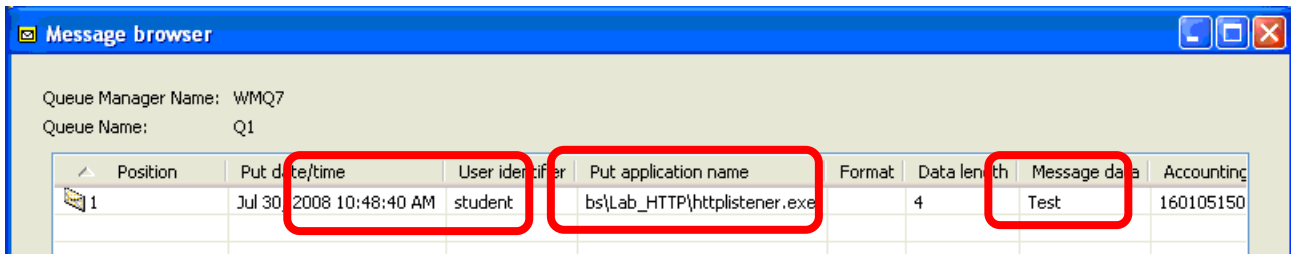
Queue name	Queue type	Definition type	Open input count	Open output count	Current queue depth	Max queue depth
Q1	Local	Predefined	0	1	1	5000
Q2	Local	Predefined	0	0	0	2

- __5. Right click on **Q1** and select **Browse Messages...**

Queue name	Queue type	Definition type	Open input count	Open output count	Current queue depth	Max queue depth
Q1	Local	Predefined	0	1	1	5000
Q2	Local	Predefined	0	0	0	2

- Compare with...
- Status...
- Delete...
- Clear Messages...
- Put Test Message...
- Browse Messages...**
- Create JMS Queue...
- Object Authorities
- Properties...

- __6. View the summary data about the message and pay particular attention to the **Put date/time** and the **Put application name**.

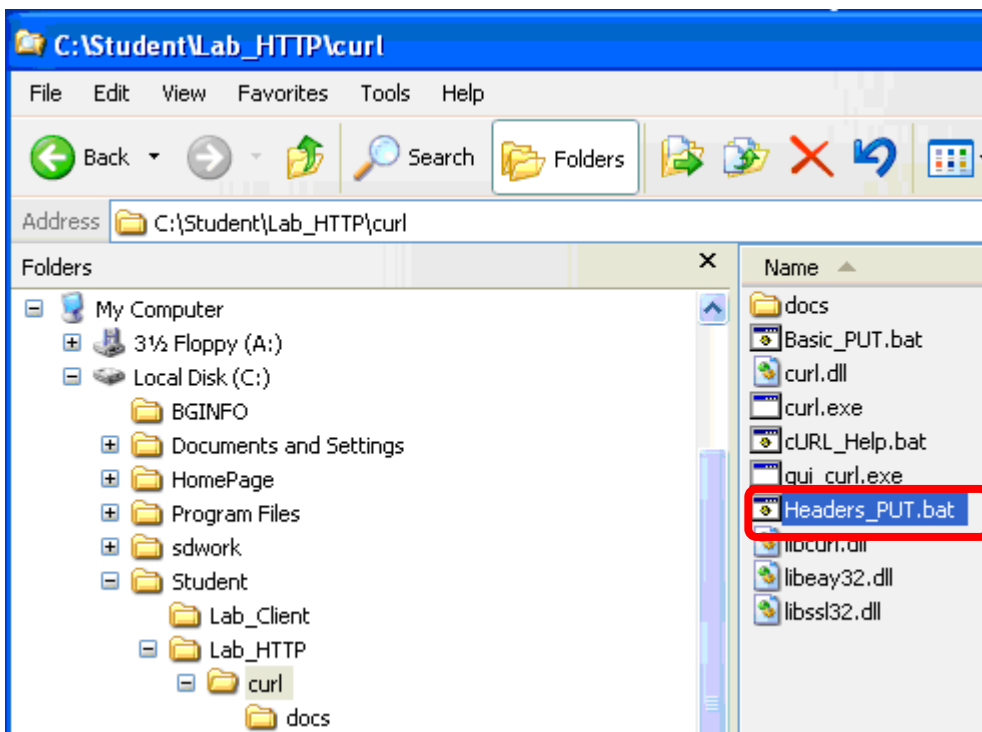


- __6. Close the Message browser window.

5.6 A PUT with an MQ Property Specified in the Headers

When using the WebSphere MQ – HTTP Bridge, MQ message header fields (MQMD) are conveyed in HTTP headers. This approach conforms to the HTTP specification. All HTTP headers used by MQ are prefixed with “x-msg-”. In this portion of the lab you will specify an MQ-specific header property for the HTTP message.

- __1. Double click on the **Headers_PUT.bat** file.



- __2. The following command will be run. **Examine its output** and then hit the **Enter** key.

```

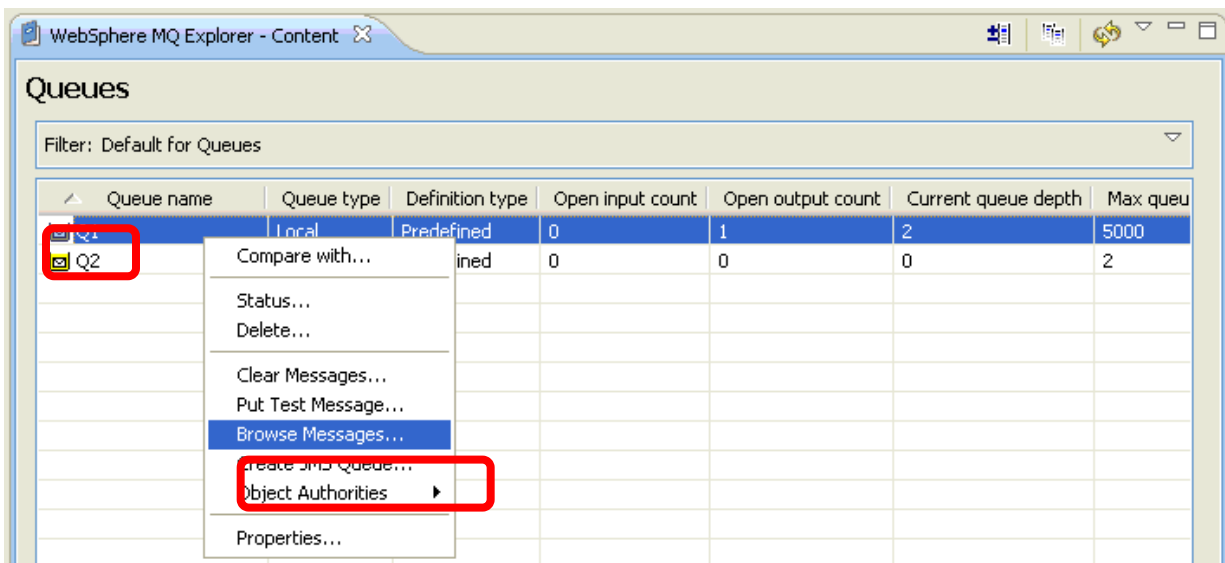
C:\WINDOWS\system32\cmd.exe

C:\Student\Lab_HTTP\curl>curl -v -d Test -H "x-msg-persistence:PERSISTENT" http://localhost/msg/queue/Q1
* About to connect() to localhost port 80 (#0)
*   Trying 127.0.0.1... connected
* Connected to localhost (127.0.0.1) port 80 (#0)
> POST /msg/queue/Q1 HTTP/1.1
> User-Agent: curl/7.16.4 (i586-pc-mingw32msvc) libcurl/7.16.4 OpenSSL/0.9.8e zlib/1.2.2
> Host: localhost
> Accept: */*
> x-msg-persistence:PERSISTENT
Content-Length: 4
Content-Type: application/x-www-form-urlencoded
<
HTTP/1.1 200 OK
Content-Length:0
Cache-Control:private
<
* Connection #0 to host localhost left intact
* Closing connection #0
*****
Review the output from the command and then press 'enter'.
Press any key to continue . . . _
  
```

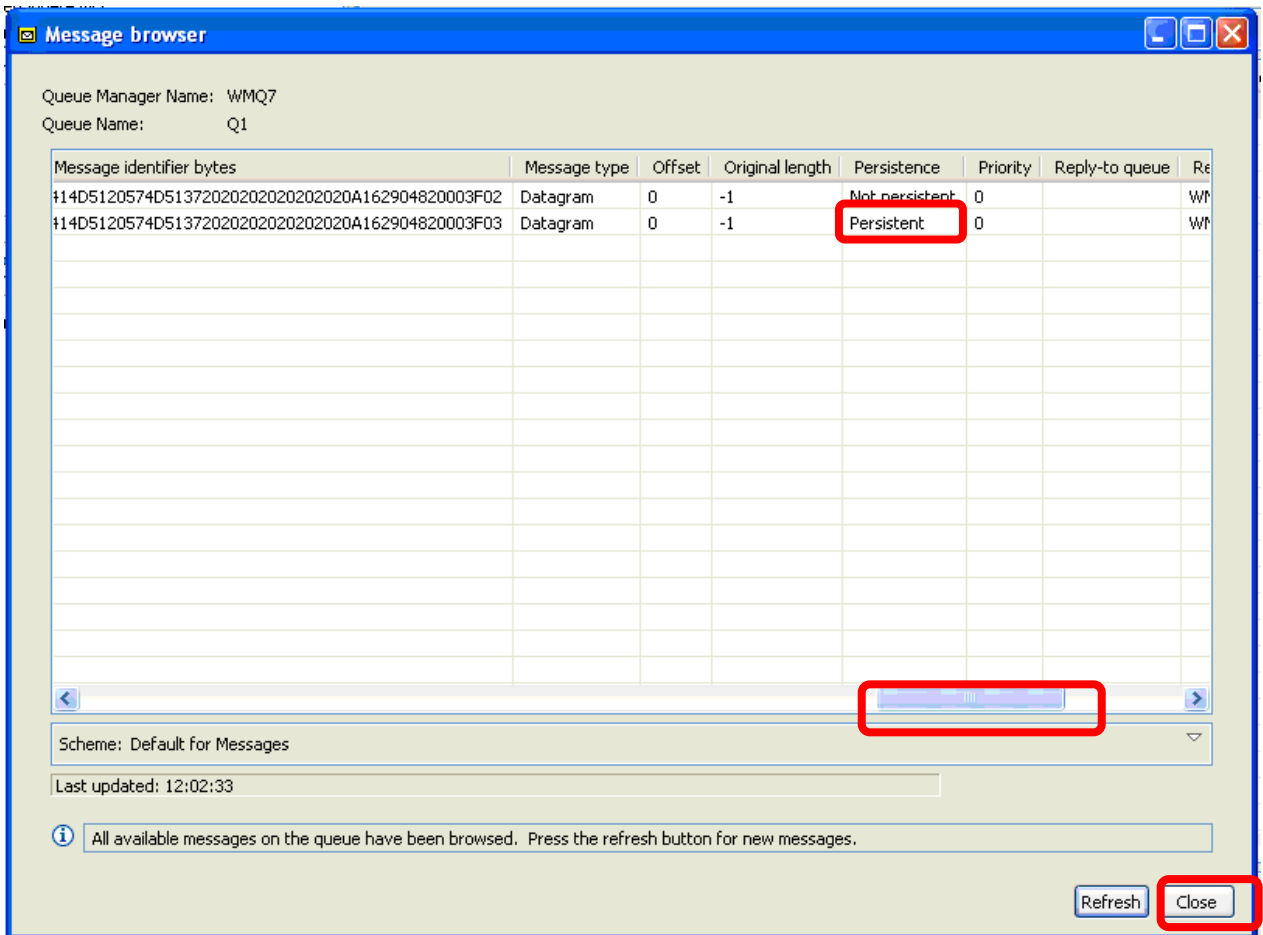
Setting the header

The header as sent by cURL

- __7. Bring the **MQ Explorer** into active view. Note that the Current queue depth is now 2. **Right click** on **Q1** and select **Browse Messages...**



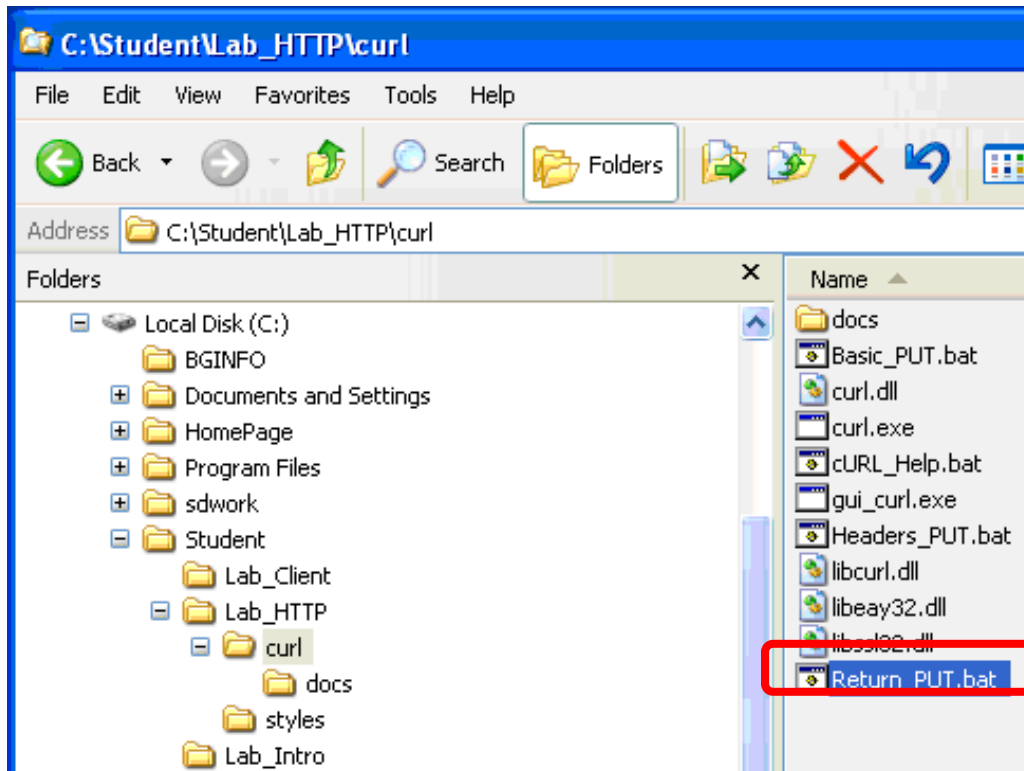
- 8. Examine the second message to **verify that it is persistent**. You will need to **scroll** quite a bit to the right to see this message property in the display. Click the **Close** button.



5.7 Requesting Information To Be Returned

cURL allows us to use the `-H` flag repeatedly to add as many headers to a message as you would like. In the next example you will use that capability to request certain information be returned to us in the response headers.

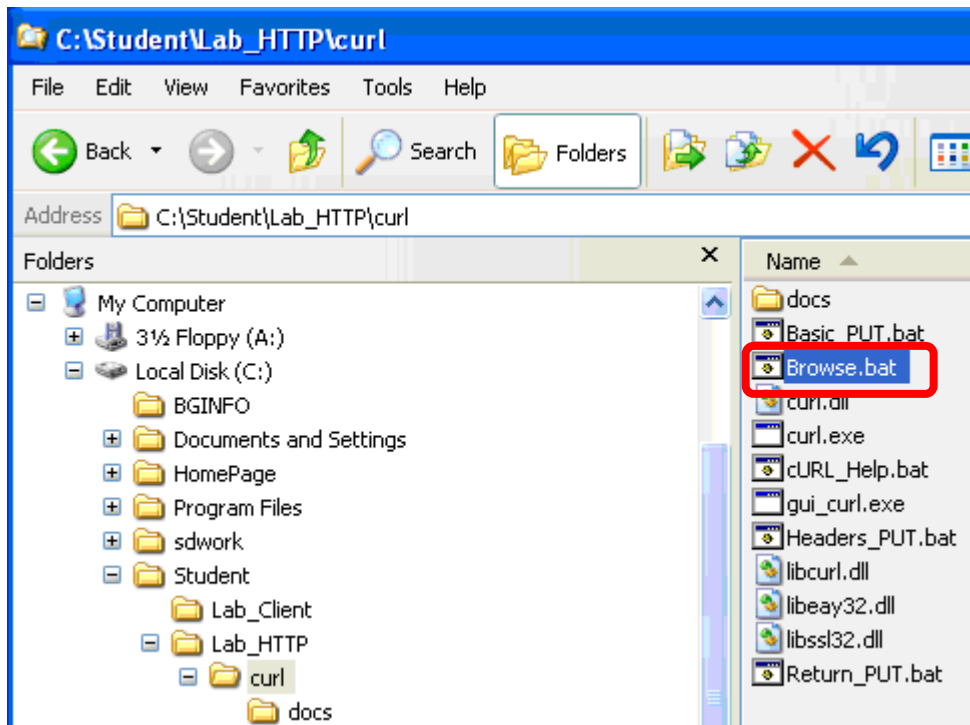
__1. Double click on the Return_PUT.bat file.....



5.8 Using cURL To Browse Messages

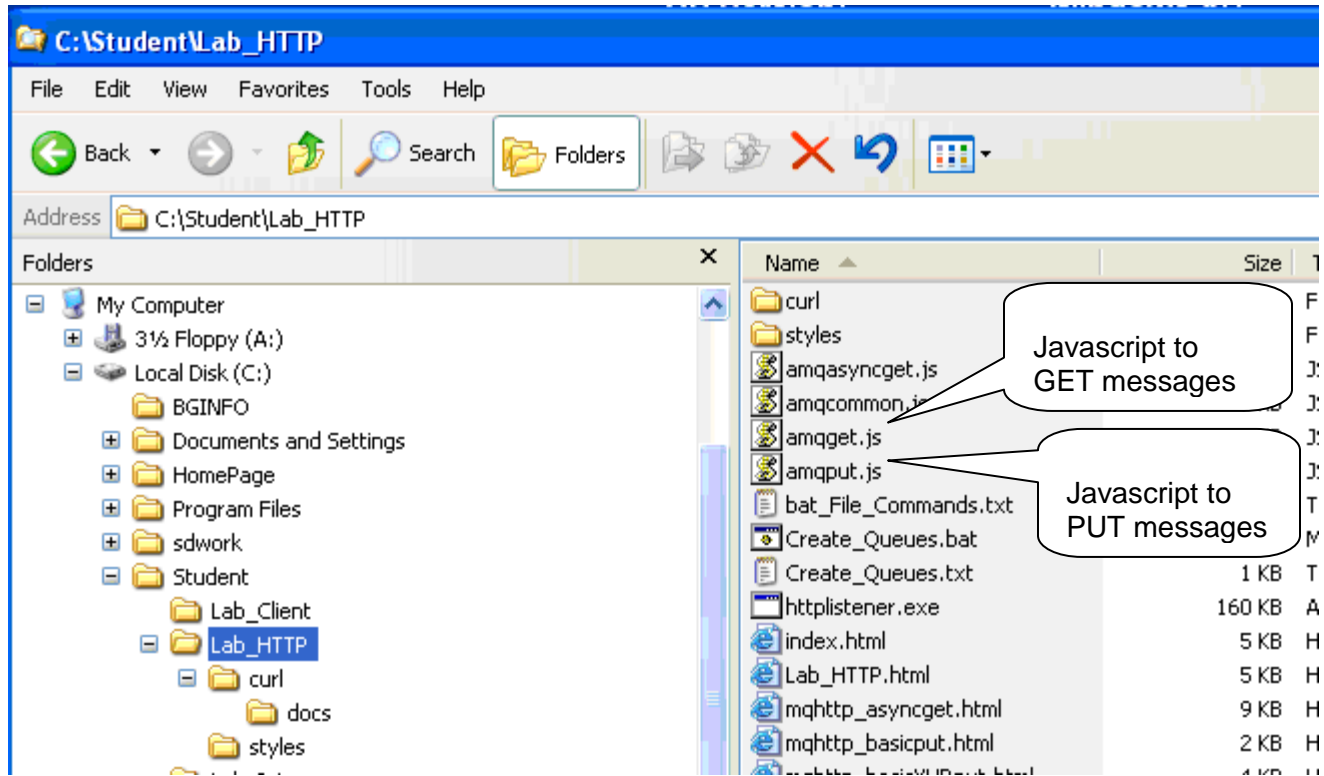
If the `-d` flag is omitted (remember this flag indicates HTTP post data) cURL can be used to issue the HTTP GET command. The MQ – HTTP Bridge interprets an HTTP GET as an MQ Browse request...the message content is returned but the message is not removed from the queue....the Browse operation is not destructive.

__1. Double click on the **Browse.Bat** file.....



This part of the lab shows how a simple JavaScript can be used to put and get messages to and from queues.

The C:\Student\Lab_HTTP directory is the home directory for the HTTP listener to serve pages from. In that directory are two simple JavaScript pages, *amqput.js* and *amqget.js*.



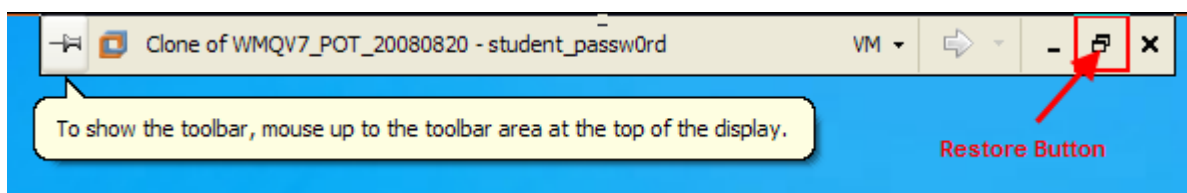
If you understand JavaScript you may want to look at these files to understand them.

One critical point is that a client needs no MQ libraries or code installed to run these JavaScripts.

You demonstrate this by running the next part of the lab **NOT ON THE VMWARE IMAGE** but on the **NATIVE OPERATING SYSTEM OF YOUR COMPUTER** which does not have any MQ code installed on it.

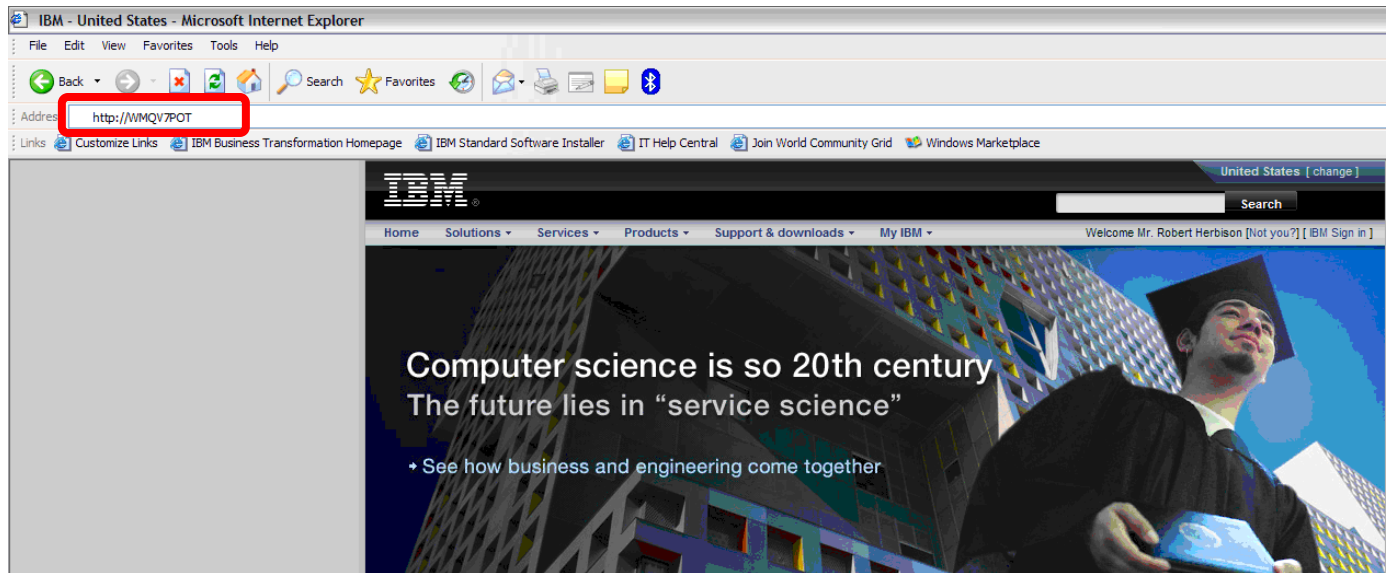
So you will now **switch to the desktop** of the **native system** on your computer.

- ___1. At the top of the window, find the VMware Toolbar. It may be hidden, or “unpinned”. Place your cursor near the top of the display, and the toolbar will automatically scroll down. Then click on the **Restore button**. Your display should change. You can now more easily switch between the native operating system and the VMware image.

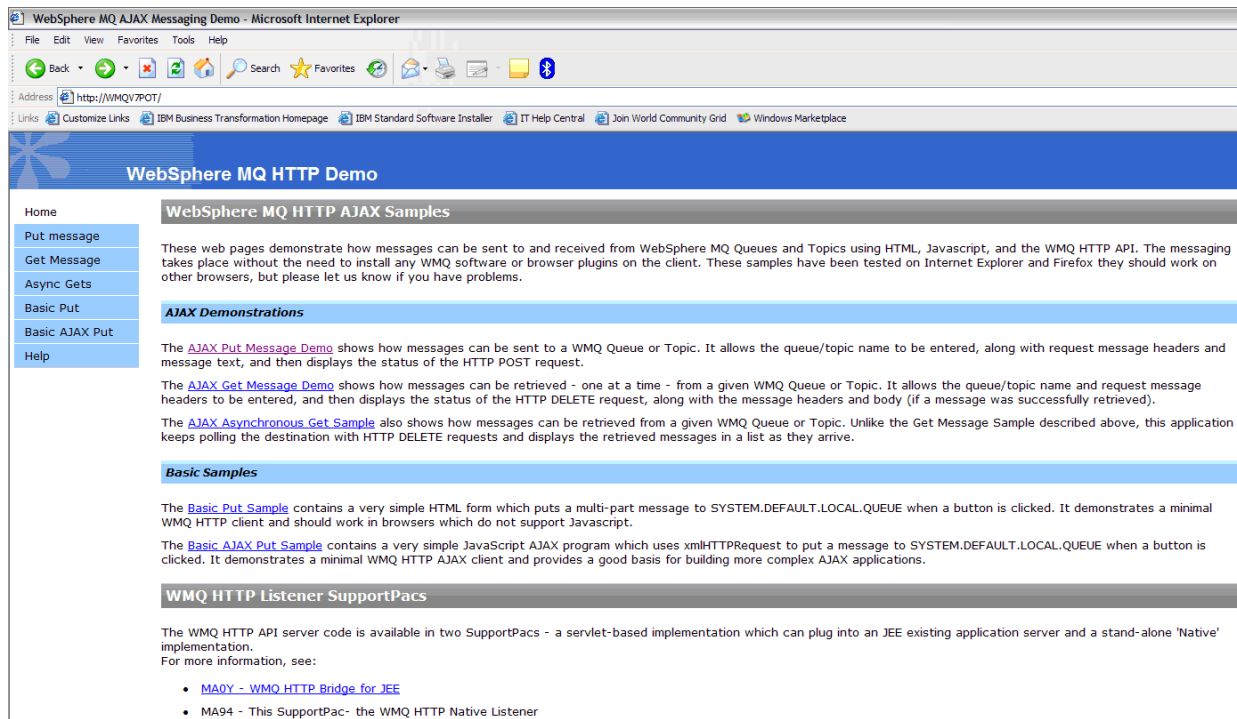


If you have any difficulty switching to the native desktop, ask the instructor for assistance.

- __12. **From the native desktop of your machine, open a web browser and enter <http://WMQV7POT> as the url and hit Enter**

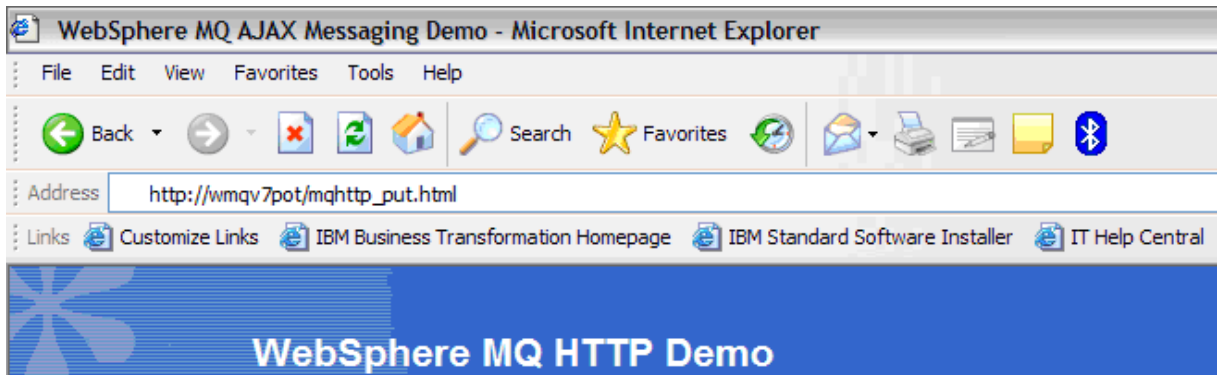


- __13. **Verify that the following welcome screen is displayed.**

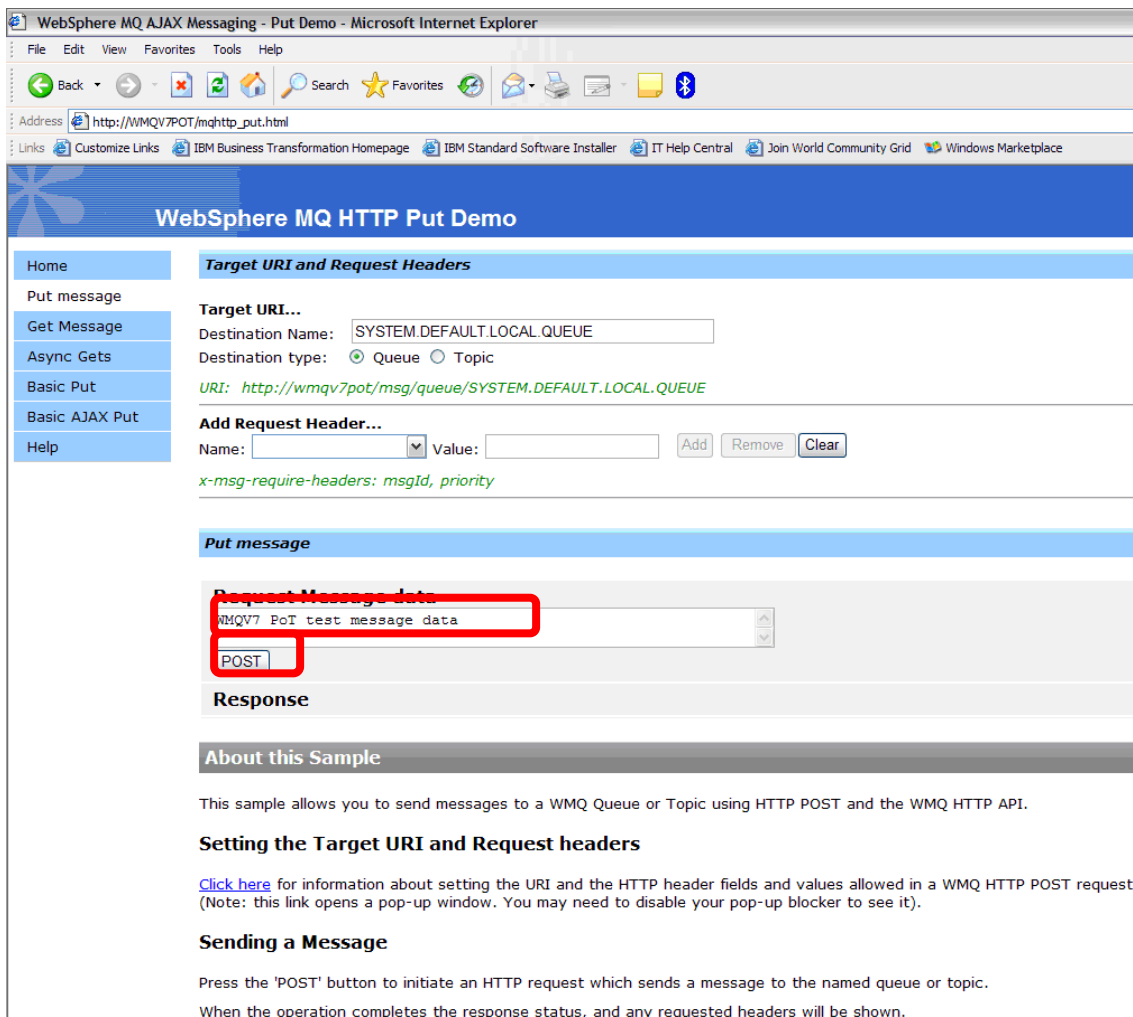


5.10 Using the PUTQ JavaScript

- __1. Enter the url http://wmgv7pot/mqhttp_put.html and hit Enter



- __2. Enter “WMQV7 PoT test message data” in the Request Message data box and press the POST button.



- 3. Observe that you get an **HTTP 200** response as well as the priority and **msgid** of the message you successfully placed on the queue.

The screenshot shows a web browser window titled "WebSphere MQ AJAX Messaging - Put Demo - Microsoft Internet Explorer". The address bar shows "http://WMQV7POT/mqhttp_put.html". The page content includes a navigation menu on the left with options like "Home", "Put message", "Get Message", "Async Gets", "Basic Put", "Basic AJAX Put", and "Help". The main area is titled "WebSphere MQ HTTP Put Demo" and contains sections for "Target URI and Request Headers", "Put message", "Request Message data", and "Response".

Target URI and Request Headers

Target URI...
Destination Name: SYSTEM.DEFAULT.LOCAL.QUEUE
Destination type: Queue Topic
URI: *http://wmqv7pot/msg/queue/SYSTEM.DEFAULT.LOCAL.QUEUE*

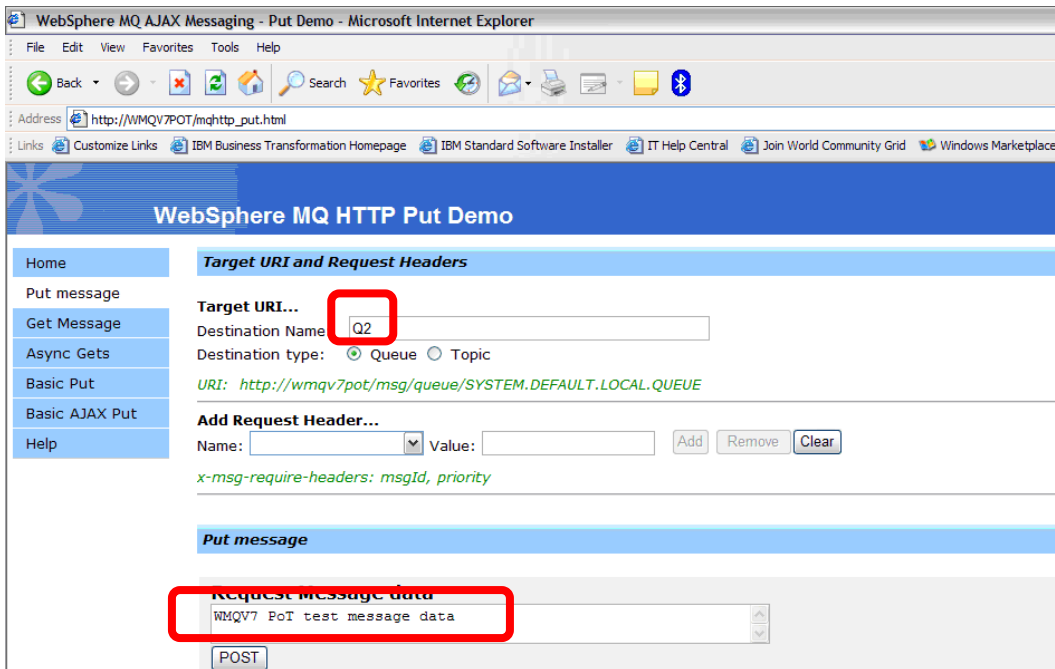
Add Request Header...
Name: Value:
x-msg-require-headers: msgId, priority

Put message

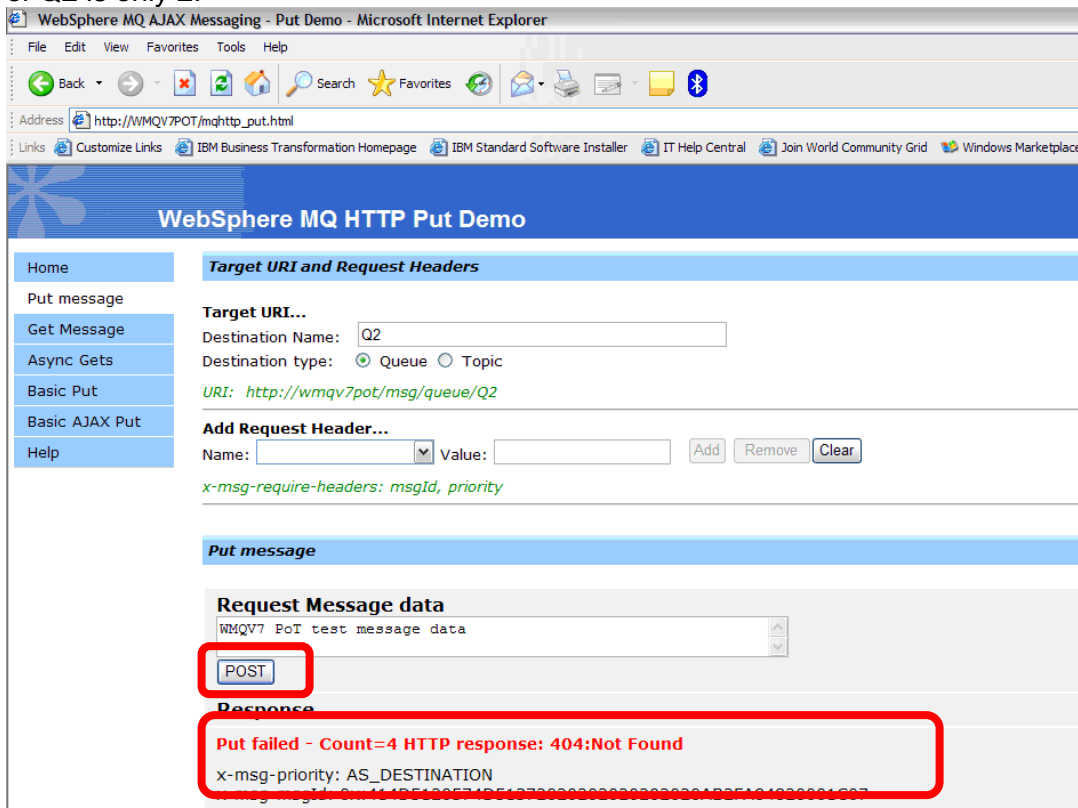
Request Message data
WMQV7 PoI test message data

Response
PUT: Count=1 - HTTP Status: 200
x-msg-priority: AS_DESTINATION
x-msg-msgId: 0x:414D5120574D513720202020202020AB2FA84820001C05

4. Change the Destination Name to Q2 and enter “WMQV7 PoT test message data” in the Request Message data box.



5. Press the **POST** button **three times**. The third time you **should get an error** – The max depth of Q2 is only 2.



This concludes Lab 5.

Appendix A. Notices

This information was developed for products and services offered in the U.S.A.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not grant you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
U.S.A.

For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

IBM World Trade Asia Corporation
Licensing
2-31 Roppongi 3-chome, Minato-ku
Tokyo 106-0032, Japan

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law: INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have

been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurements may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

All statements regarding IBM's future direction and intent are subject to change or withdrawal without notice, and represent goals and objectives only.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental. All references to fictitious companies or individuals are used for illustration purposes only.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs.

Appendix B. Trademarks and copyrights

The following terms are trademarks of International Business Machines Corporation in the United States, other countries, or both:

IBM	AIX	CICS	ClearCase	ClearQuest	Cloudscape
Cube Views	DB2	developerWorks	DRDA	IMS	IMS/ESA
Informix	Lotus	Lotus Workflow	MQSeries	OmniFind	
Rational	Redbooks	Red Brick	RequisitePro	System i	
<i>System z</i>	<i>Tivoli</i>	<i>WebSphere</i>	<i>Workplace</i>	<i>System p</i>	

Adobe, Acrobat, Portable Document Format (PDF), and PostScript are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States, other countries, or both.

Cell Broadband Engine is a trademark of Sony Computer Entertainment, Inc. in the United States, other countries, or both and is used under license therefrom.

Java and all Java-based trademarks and logos are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both. See Java Guidelines

Microsoft, Windows, Windows NT, and the Windows logo are registered trademarks of Microsoft Corporation in the United States, other countries, or both.

Intel, Intel logo, Intel Inside, Intel Inside logo, Intel Centrino, Intel Centrino logo, Celeron, Intel Xeon, Intel SpeedStep, Itanium, and Pentium are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.

UNIX is a registered trademark of The Open Group in the United States and other countries.

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

ITIL is a registered trademark and a registered community trademark of the Office of Government Commerce, and is registered in the U.S. Patent and Trademark Office.

IT Infrastructure Library is a registered trademark of the Central Computer and Telecommunications Agency which is now part of the Office of Government Commerce.

Other company, product and service names may be trademarks or service marks of others.



© Copyright IBM Corporation 2011.

The information contained in these materials is provided for informational purposes only, and is provided AS IS without warranty of any kind, express or implied. IBM shall not be responsible for any damages arising out of the use of, or otherwise related to, these materials. Nothing contained in these materials is intended to, nor shall have the effect of, creating any warranties or representations from IBM or its suppliers or licensors, or altering the terms and conditions of the applicable license agreement governing the use of IBM software. References in these materials to IBM products, programs, or services do not imply that they will be available in all countries in which IBM operates. This information is based on current IBM product plans and strategy, which are subject to change by IBM without notice. Product release dates and/or capabilities referenced in these materials may change at any time at IBM's sole discretion based on market opportunities or other factors, and are not intended to be a commitment to future product or feature availability in any way.

IBM, the IBM logo and ibm.com are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both. If these and other IBM trademarked terms are marked on their first occurrence in this information with a trademark symbol (® or ™), these symbols indicate U.S. registered or common law trademarks owned by IBM at the time this information was published. Such trademarks may also be registered or common law trademarks in other countries. A current list of IBM trademarks is available on the Web at "Copyright and trademark information" at ibm.com/legal/copytrade.shtml

Other company, product and service names may be trademarks or service marks of others.



Please Recycle
