Using WebSphere MQ V7 as JMS Provider for WebSphere Application Server V7, V8.0 and V8.5, and MQ 9.0 for WAS V9.0

IBM Techdoc: 7016505

http://www.ibm.com/support/docview.wss?uid=swg27016505

Date last updated: 4-Apr-2017

Angel Rivera – rivera@us.ibm.com
IBM WebSphere MQ Support

+++ Objective +++

To demonstrate the use of a simple but functional Message Driven Bean (MDB) in WebSphere Application Server V7, V8.0, V8.5 and V9.0, which interacts with WebSphere MQ as the Java™ Messaging Service (JMS) provider.

Shipped version of MQ RA with WAS:
- WAS 7.x and 8.0 ship MQ 7.0 Resource Adapter
- WAS 8.5 ships MQ 7.1 Resource Adapter
- The MQ V7.5 and V8 JMS clients are NOT shipped with any version of WAS.
- WAS 9.0 ships MQ 9.0 Resource Adapter.

The MDB was created with Rational Application Developer (RAD) 7.5 and the Enterprise Archive File (EAR) file which contains the MDB can be downloaded from this techdoc.

The main scenario is to show how to configure both WebSphere Application Server and WebSphere MQ V7.x, in order for an MDB to get messages from MQ, using a WebSphere Application Server Listener Port from a Queue (Point to Point).

++ What is new in this update:

- On Apr-2017, the procedures were tested with a queue manager running on MQ 9.0.2.0 and a WAS server 9.0.0.0 (using MQ RA 9.0.0.0), in Windows 10

- On Aug-2016, the procedures were tested with a queue manager running on MQ 9.0.0.0 and a WAS server 9.0.0.0 (using MQ RA 9.0.0.0)

- On May-2013, the procedures in this techdoc were successfully tested with a queue manager running on MQ 7.5.0.1 and a WAS server 8.5.0.2 (using MQ RA 7.1.0.2).

- On Aug-2012, the procedures in this techdoc were successfully tested with a queue manager running on MQ 7.0.1.9 and a WAS server 8.0.0.4 (using MQ RA 7.0.1.7).

- When using MQ 7.1 or 7.5 queue managers, if the MQ JMS client in WAS gets a RC 2035, one possibility is that the "channel authentication records" are enabled. For more details see the following technote:
http://www.ibm.com/support/docview.wss?uid=swg21577137
WMQ 7.1 / 7.5 / 8.0 / 9.0 queue manager RC 2035 MQRC_NOT_AUTHORIZED or AMQ4036 or JMSWMQ2013 when using client connection as an MQ Administrator

++ Additional scenarios are provided to illustrate the following:

- How to use MQ Topics to exploit Publish/Subscribe from MQ.

- How to use a WebSphere Application Server Activation Specification, instead of a WebSphere Application Server Listener Port.

This document shows all screen shots needed for beginners. The chapters are:

Chapter 1: MQ V7 configuration: queue and topic
Chapter 2: WebSphere Application Server configuration: JNDI objects
Chapter 3: WebSphere Application Server deployment and testing of MDB
Chapter 4: Additional scenarios: Topic, Activation Specification

+++ Related techdocs +++

The MDB used in this presentation, was created with IBM Rational Application Developer (RAD) 7.5 as part of an Enterprise Java Bean, which is provided in an EAR file. The steps for its creation and testing are documented in a related techdoc:

http://www.ibm.com/support/docview.wss?uid=swg27016507
Developing and testing an MDB using RAD 7.5, WebSphere Application Server V7 and MQ V7 as JMS Provider

Additional techdocs:

http://www.ibm.com/support/docview.wss?uid=swg27016581
Using an MDB with JMS message selectors with WebSphere MQ V7 and WebSphere Application Server V7
Includes sample code to create a message property "color"
Sample MDB: SampleJMSSelectorMdbEjbEAR.ear
Sample Java code: SampleJMSMsgProperty.java

http://www.ibm.com/support/docview.wss?uid=swg27016582
Using an MDB that always rolls back a message to test the handling of poison messages
(WebSphere MQ V7.x, V8, V9, WebSphere Application Server V7, V8.x, V9)

http://www.ibm.com/support/docview.wss?uid=swg21248089
Which version of WebSphere MQ Resource Adapter (RA) is shipped with WebSphere Application Server ?

+++ Requisite software +++

The following software was used:

SUSE Linux Enterprise Server (SLES) 9:
  WebSphere Application Server 7.0.0.5
  WebSphere MQ 7.0.0.2
  Firefox (also known as Mozilla)

Windows:
Initial Version of this techdoc:
        Windows XP SP3: Internet Explorer
04-Apr-2017:
        Windows 10, with Firefox

+++ Downloadable files +++

The following files are included as attachments to this techdoc

EAR file with MDB:
        SampleMDBEJB.ear

Text file with code excerpt:
        onMessage.txt

+++ About the EAR file and the MDB +++

The MDB used in this presentation, was created with IBM Rational Application Developer (RAD) 7.5 as part of an Enterprise Java Bean, which is provided in an EAR file. The steps for its creation and testing are documented in the techdoc: **Developing and testing an MDB using RAD 7.5, WebSphere Application Server V7 and MQ V7 as JMS Provider**

The Deployment Descriptor for the EAR already specifies the need to have a Listener Port for an MQ Queue. In that way, it is possible to do a "fast path" deployment of the application in WebSphere Application Server. In addition, to facilitate such fast deployment, the necessary MQ and JMS WebSphere Application Server objects will be defined and configured in the first chapters of this techdoc.

The onMessage() method of this MDB has the following source code which displays the type of contents (payload) and an "eye catcher string" (+++ SAMPLE MDB) which can let you find quickly the output of the MDB in the SystemOut.log file.

The whole source for this method is available in the following text file associated with this techdoc: onMessage.txt

```
public void onMessage(javax.jms.TextMessage msg) {
  try {
        if (msg instanceof javax.jms.TextMessage) {
                System.out.println("+++ SAMPLE MDB: Text Message
=> " + ((javax.jms.TextMessage)msg).getText());
        }
        else if (msg instanceof javax.jms.BytesMessage) {
                System.out.println("+++ SAMPLE  MDB: Bytes Mes-
sage ");
…
…
        }
  }
  catch(javax.jms.JMSException e) {
        System.out.println("+++ SAMPLE MDB: Error occurred ");
        e.printStackTrace();
  }
}
```

For a Text Message, the actual text of the message is displayed, such as "TESTING". Thus, upon receiving a message the MDB will display the following in the SystemOut.log:

```
+++ SAMPLE MDB: Text Message => TESTING
```

++
+++ Chapter 1: MQ V7 configuration: create queue and topic
++

The purpose of this chapter is to show the MQ commands and the steps to take using the MQ Explorer to create a new queue manager, a local queue (for Point to Point) and a topic (for Publish/Subscribe).

+ Summary of objects:

Queue Manager name:	QM_MDB	
Queue name:	Q_MDB	
Topic name:	T_MDB	=> Topic String: sports
MQ Listener name:	TCP_LISTENER	=> Port: 1420

Note that in this example the port 1420 was used to avoid a conflict with an existing queue manager in my test host that already uses the default 1414 port number. You can use specify another port number for the MQ Listener for your queue manager.

+ Summary of MQ commands to create these objects:

This is a summary intended for intermediate and advanced MQ users. If you are a beginner, you could just browse this section and skip to the next one.

Create queue manager, with a Dead Letter Queue (DLQ)
```
$ crtmqm -u SYSTEM.DEAD.LETTER.QUEUE QM_MDB
```

Start the queue manager
```
$ strmqm QM_MDB
```

Invoke the administration utility
```
$ runmqsc QM_MDB
```

Within runmqsc, define and start an MQ Listener:
```
DEFINE LISTENER(TCP.LISTENER) TRPTYPE(TCP) CONTROL(QMGR)
PORT(1420)

START LISTENER(TCP.LISTENER)
```

Define a channel to be used with the MQ Explorer (optional but very useful!):
```
DEFINE CHANNEL(SYSTEM.ADMIN.SVRCONN) CHLTYPE(SVRCONN)
```

Define a local queue:
```
DEFINE QLOCAL(Q_MDB)
```

Define a topic object:

```
DEFINE TOPIC(T_MDB) TOPICSTR('sports')
```

For MQ 7.1 and later, and if desiring to allow remote connections by an MQ Administrator:

```
set CHLAUTH(*) TYPE(BLOCKUSER) USERLIST('nobody','*MQADMIN')
set CHLAUTH(SYSTEM.ADMIN.*) TYPE(BLOCKUSER) USERLIST('nobody')
```

For MQ 8.0 and later, and if desiring to have the same behavior as in MQ 7.x in which passwords are not required, issue:

```
ALTER AUTHINFO(SYSTEM.DEFAULT.AUTHINFO.IDPWOS) AUTHTYPE(IDPWOS) +
   CHCKCLNT(OPTIONAL)
REFRESH SECURITY TYPE(CONNAUTH)
.
```

Exit runmqsc:

```
END
```

+ Detailed instructions using the MQ Explorer.

These instructions assume that you do not have yet a queue manager for testing the MDB. Thus, this section provides the detailed steps to create one and the necessary objects (queue and topic)

- Login as an MQ Administrator.

- Start the MQ Explorer:

Windows:

a) Short version:
Start > All Programs > IBM WebSphere MQ > WebSphere MQ Explorer

b) Longer version:
Right click on the MQ Icon on the System Tray (right bottom corner). Select WebSphere MQ Explorer.

Linux:

Open a command prompt and issue:
/opt/mqm/bin/strmqcfg

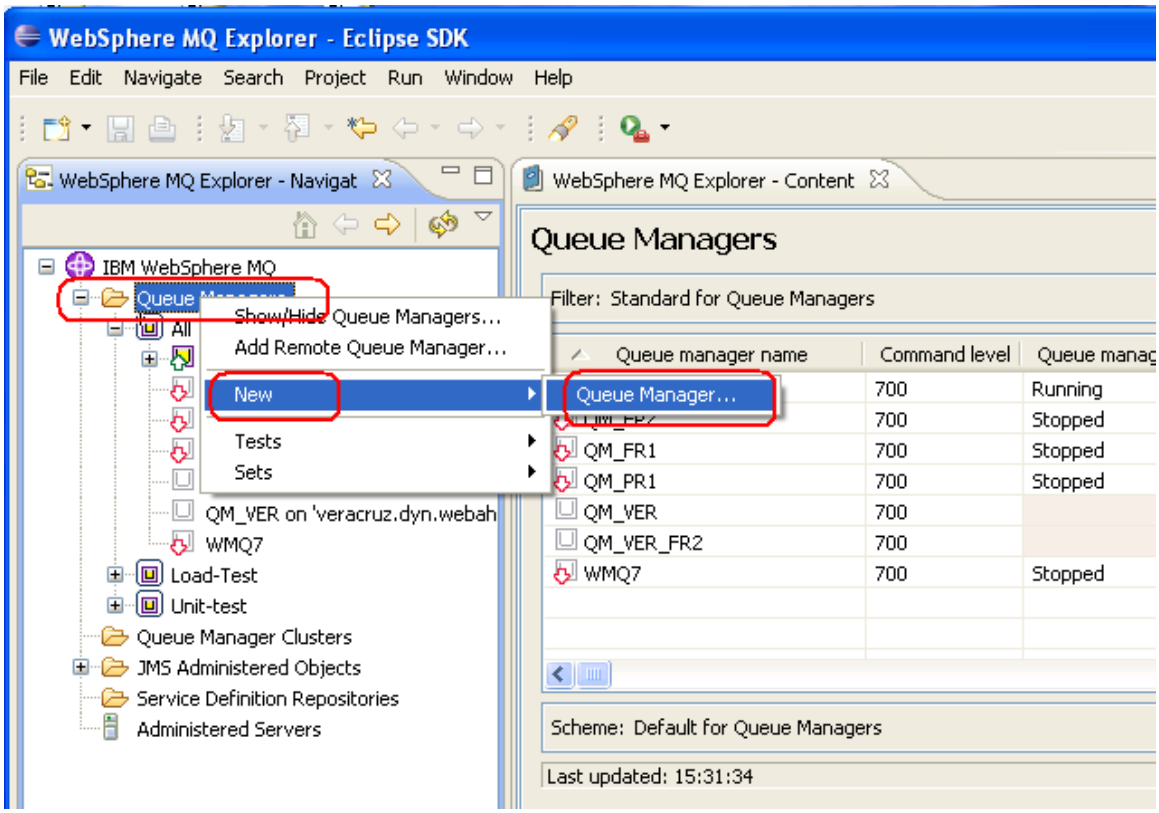
You could add an icon to your Linux desktop to run the command:
/opt/mqm/bin/strmqcfg

For KDE users:

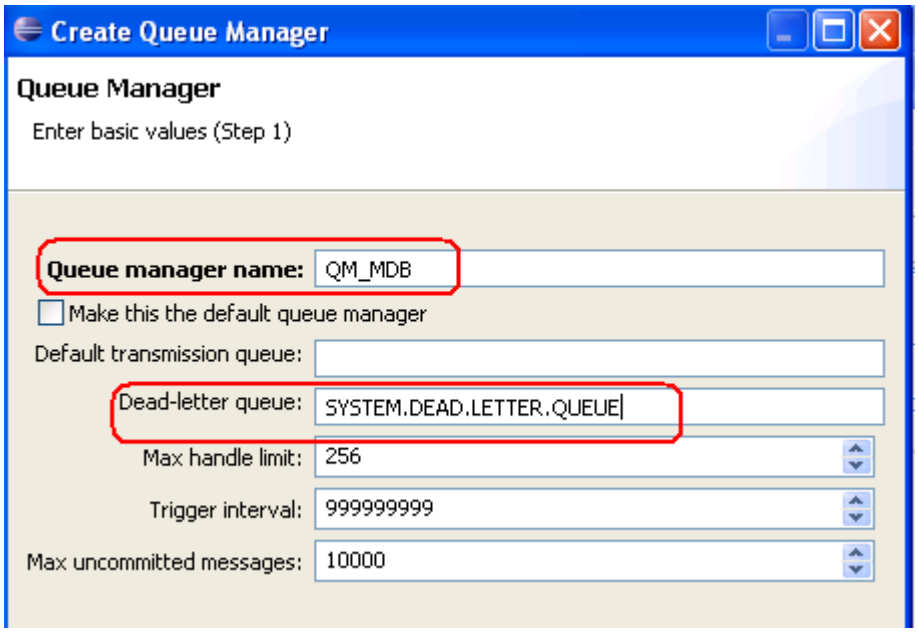
On the desktop wallpaper, click on the right mouse to bring the context menu.
Then select: Create New > File > Link to Application …
You will see a new window:
- In the tab “General” enter the title for the icon.
- In the tab “Application” enter in the field “Command”:
/opt/mqm/bin/strmqcfg

For Gnome users:
The main mouse option for creating an icon is "Create Launcher..."
The field where the pathname goes is also labeled "Command":
/opt/mqm/bin/strmqcfg

- From the MQ Explorer, invoke the wizard to create a new queue manager: from the left navigation panel (Navigator), right click on the folder “Queue Managers”, then “New”, then “Queue Manager…”



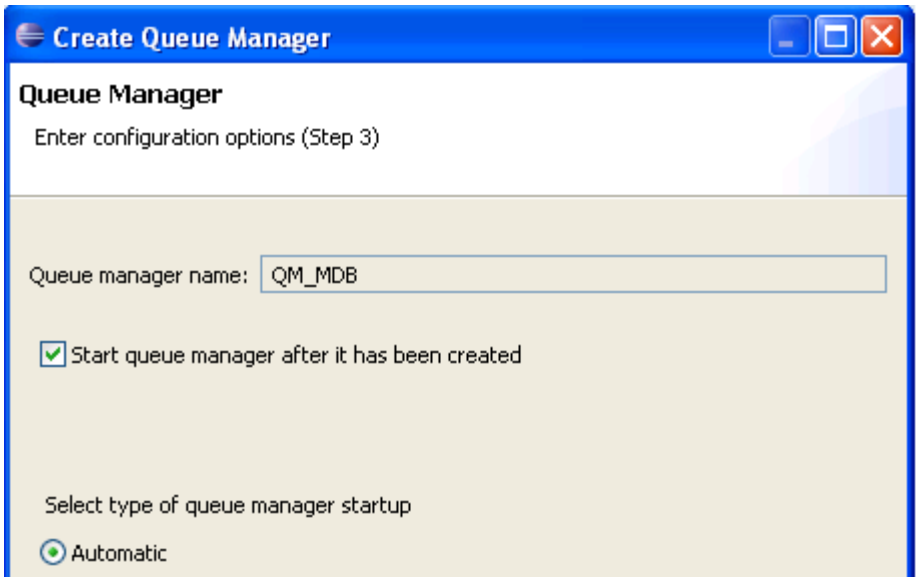
- From the first screen of the wizard “Create Queue Manager”, specify:
  + The name of the queue manager, in this case is: QM_MDB
  + The dead letter queue (DLQ): SYSTEM.DEAD.LETTER.QUEUE



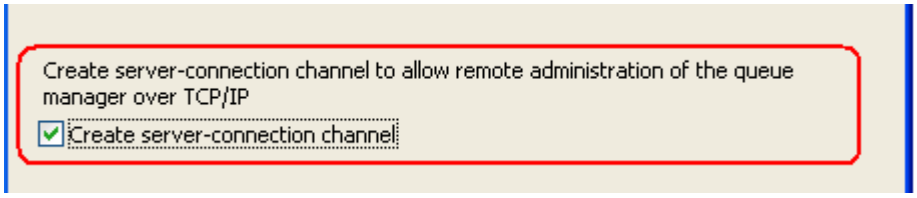
Then click on Next.

- Because this is a test Queue Manager, accept the defaults for the transactional and recovery logs (Step 2).

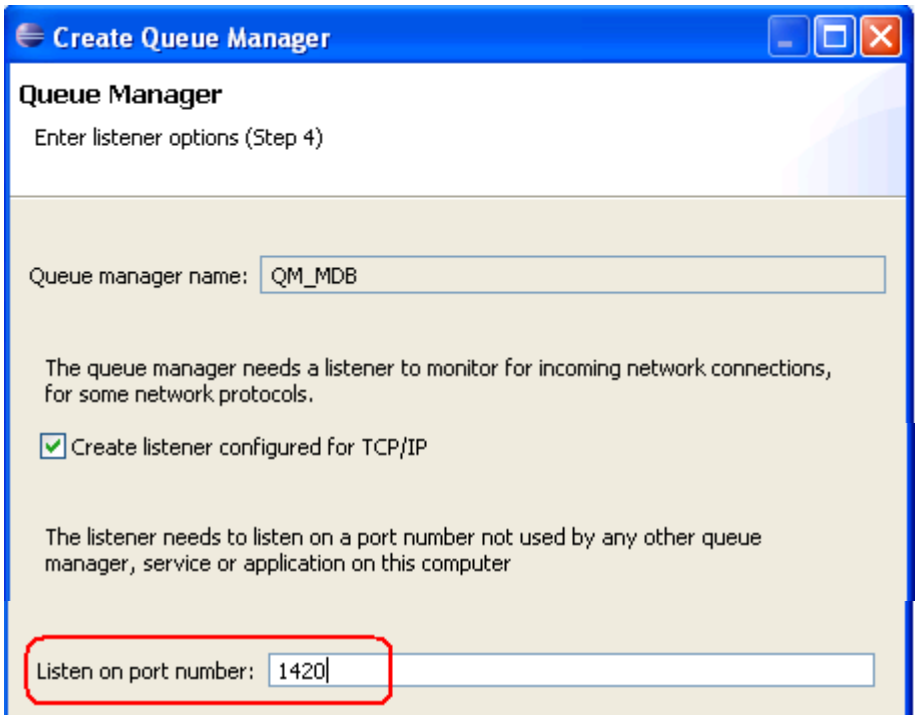
- In the next screen, accept the defaults, but indicate that you want to interact with the wizard to “Create server-connection channel”, to allow MQ clients, either local (bindings) or remote (via TCP/IP) to access the queue manager.
Note: The channel to be created is named: `SYSTEM.ADMIN.SVRCONN`



...



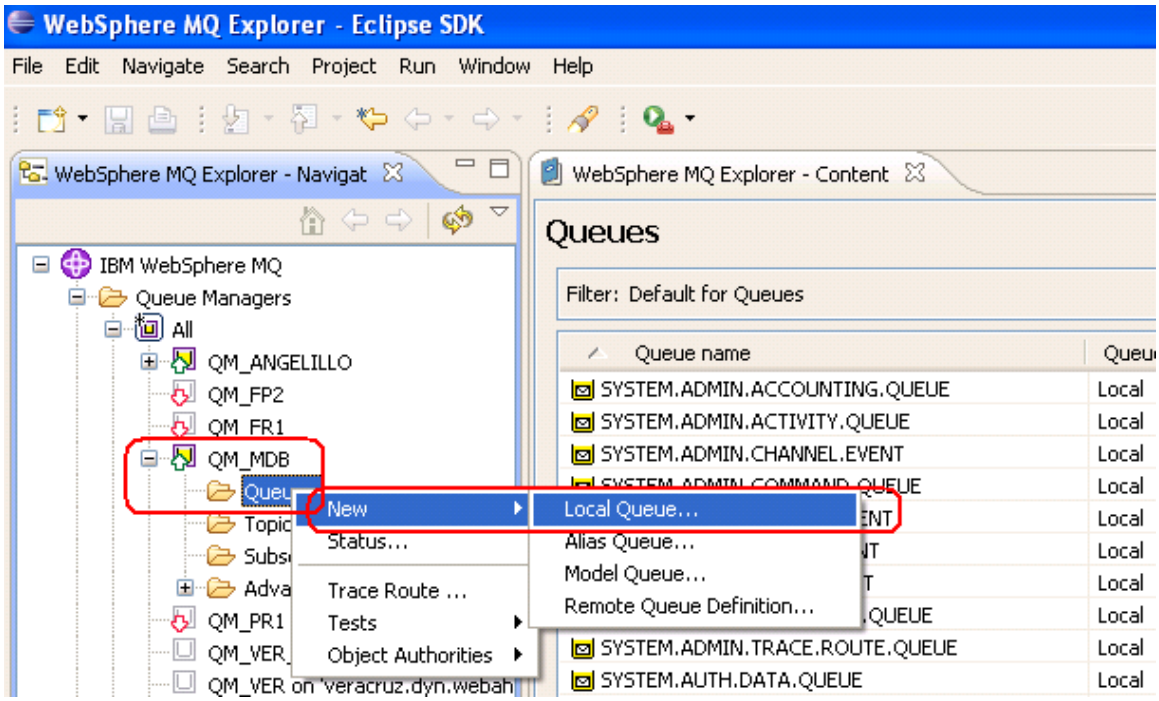
- In the next screen, accept the default to “Create listener configured for TCP/IP”.
The default port number is “1414”, but for the purposes of this exercise, the port “1420” is used.
This is the MQ Listener and it is not related to the Listener Port from the Web-Sphere Application Server.



Click on Next.

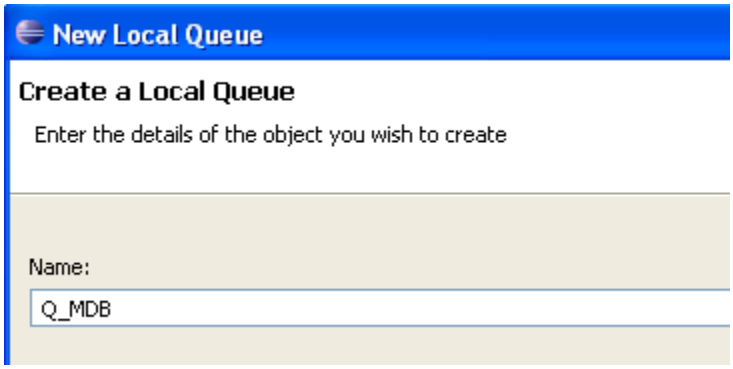
- In the next screen, accept the defaults and click on Finish.
- The new queue manager QM_MDB will be created.

- Create a local queue, named “Q_MDB”. From the Navigator (left panel), expand the queue manager “QM_MDB” and right click on the folder for “Queues”. Select: New > Local Queue…



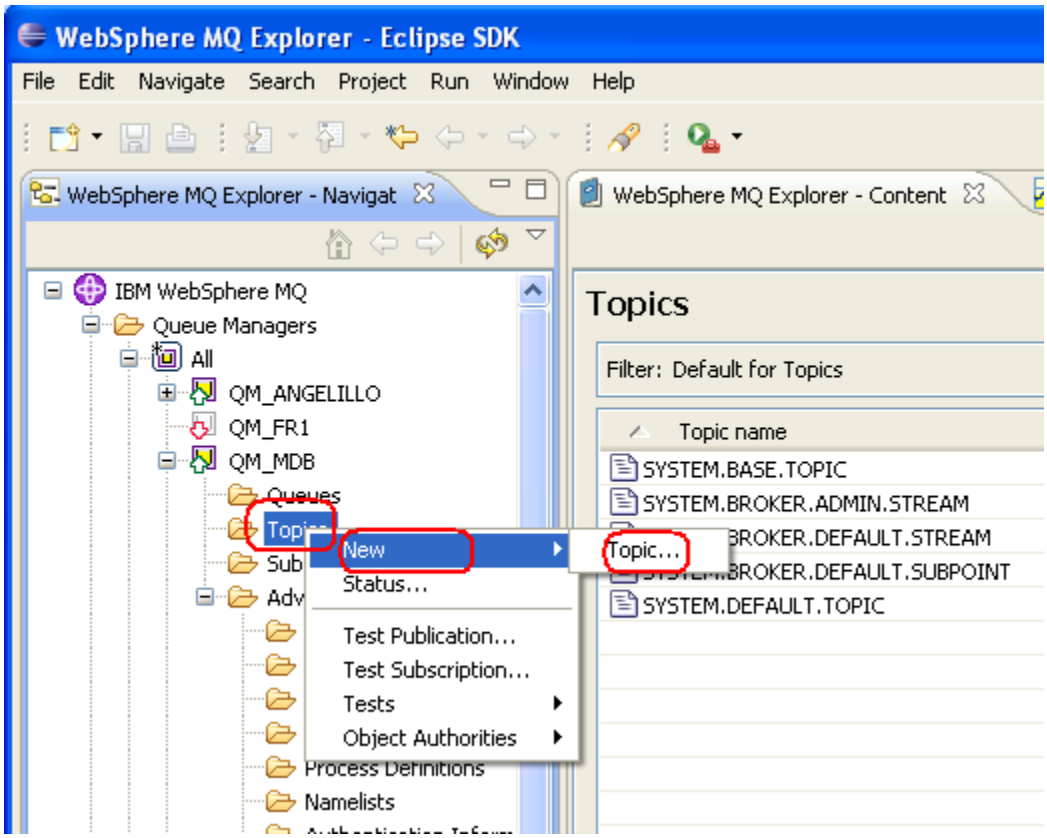
- From the wizard “New Local Queue” enter the name for the queue: Q_MDB
Accept the defaults.

Applications running in WebSphere Application Server use JNDI objects defined in the application server, not in WebSphere MQ. Therefore we will not check the box "Start Wizard to create matching JMS Queue".
However, If you plan to run JMS programs outside WebSphere Application Server, you may want to check this box and proceed with the wizard to create the corresponding JNDI objects for MQ.

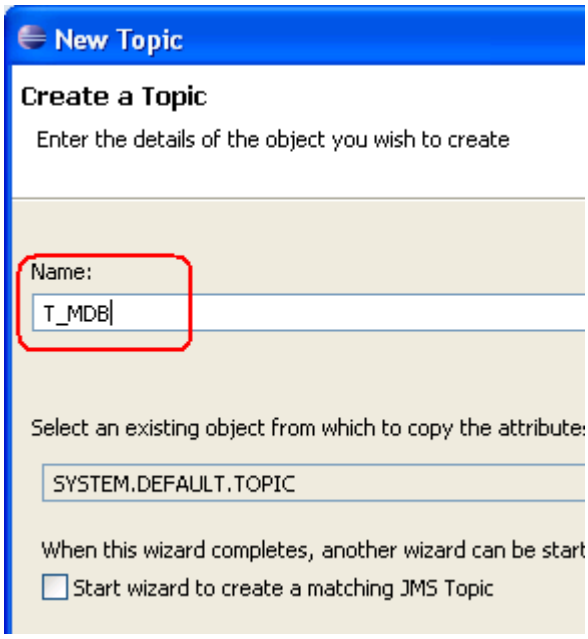


Click on Finish.

- Create a topic object, named “T_MDB”, with topic string “sports”.
From the Navigator (left panel), expand the queue manager “QM_MDB” and right click on the folder for “Topics”. Select: New > Topic…

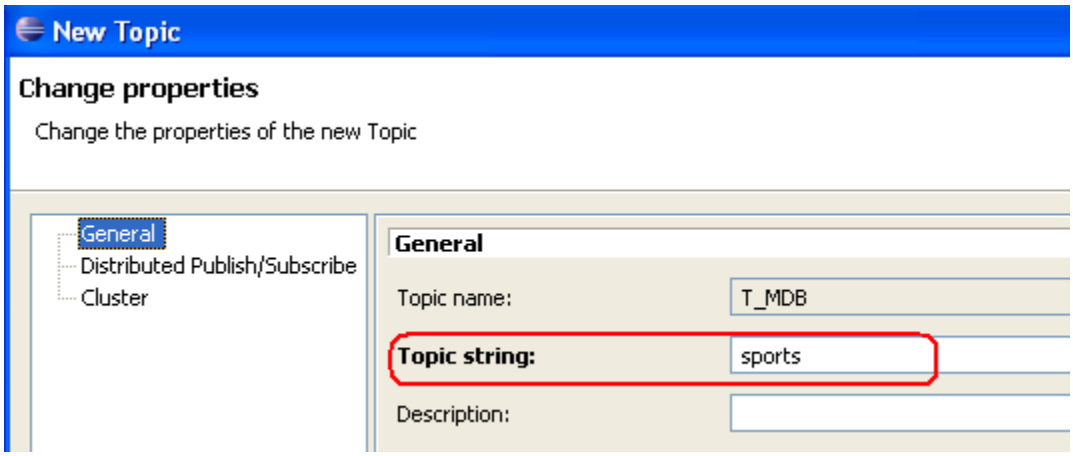


- From the wizard “New Topic” enter the name for the topic object: T_MDB



Click on Next.

Then specify the topic string of “sports”

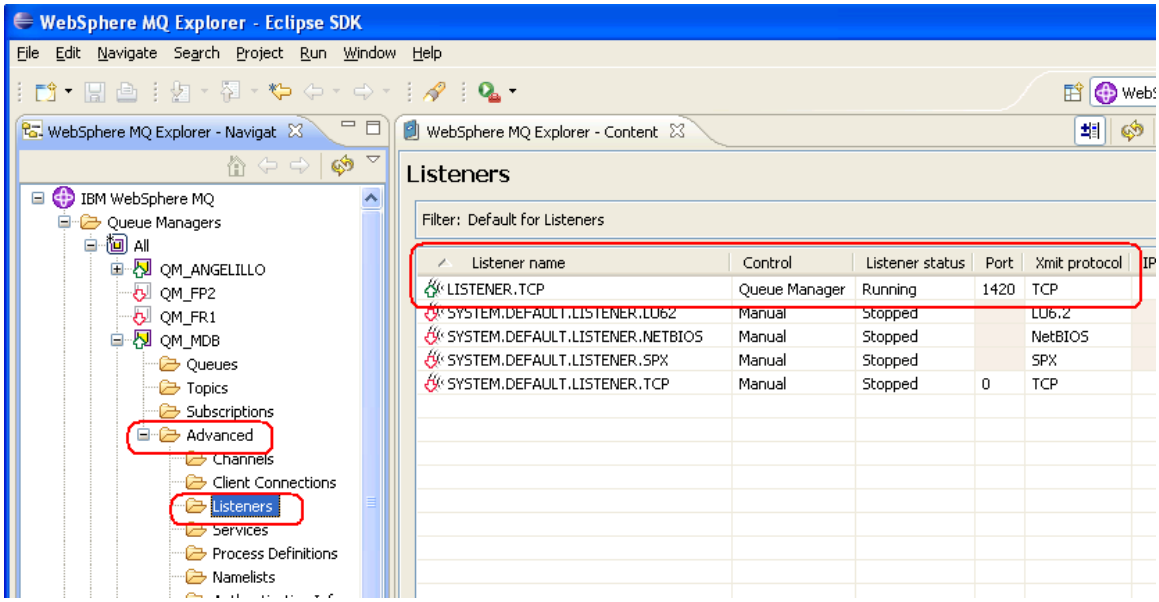


Accept the defaults.
Click on Finish.

Verify that the MQ Listener “LISTENER.TCP” has been defined.

The wizard for create a queue manager already asked us if we wanted to create a listener. To confirm, expand the folders “Advanced” and then “Listeners”. Notice that the listener “LISTENER.TCP” using Port 1420 was created and it is shown in green to indicate that it is active.

NOTICE: An MQ Listener is NOT the same as a WebSphere Application Server Listener Port. They are NOT related at all!



That is all for the configuration of MQ!

Proceed with the next chapter to configure the JNDI objects within WebSphere Application Server.

++
+++ Chapter 2: WebSphere Application Server configuration: JNDI objects
++

The purpose of this chapter is to show you how to use the WebSphere Application Server Administrative Console to create JMS administration objects which will be stored in the Java Naming and Directory Interface (JNDI) directory service provided by WebSphere Application Server.

The defaults will be used as much as possible, in order to keep this example simple but functional.

+ Summary of the characteristics of the JNDI and other objects.

WebSphere Application Server server name: server1
URL of WebSphere Application Server Administrative console:
http://localhost:9060/ibm/console/unsecureLogon.jsp

Connection Factory:

Name:	SampleMDBConnectionFactory
JNDI Name:	jms/SampleMDBConnectionFactory
Queue Manager:	QM_MDB
Transport:	Bindings, then client
Hostname:	localhost
Port:	1420
Server connection channel:	SYSTEM.DEF.SVRCONN

Destination: Queue

Name:	SampleMDBQueue
JNDI Name:	jms/SampleMDBQueue
Queue Name in MQ:	Q_MDB
Queue Manager:	QM_MDB

Destination: Topic

Name:	SampleMDBTopic
JNDI Name:	jms/SampleMDBTopic
Topic Name in MQ:	sports
Broker publication queue manager:	QM_MDB

Note:
The WebSphere Application Server field “Topic Name” refers to the Topic String, and not to the name of the MQ Topic Object.

Activation Specification for a Queue:

Name:	SampleMDBQueueActivationSpec
JNDI Name:	jms/SampleMDBQueueActivationSpec
Destination JNDI name:	jms/SampleMDBQueue
Destination Type:	Queue
Queue manager:	QM_MDB
Transport :	Bindings, then client
Hostname:	localhost
Port:	1420
Server connection channel:	SYSTEM.DEF.SVRCONN

Activation Specification for a Topic:

Name:	SampleMDBTopicActivationSpec
JNDI Name:	jms/SampleMDBTopicActivationSpec
Destination JNDI name:	jms/SampleMDBTopic
Destination Type:	Topic
Type of subscription:	Nondurable subscription
Queue manager:	QM_MDB
Transport :	Bindings, then client
Hostname:	localhost
Port:	1420
Server connection channel:	SYSTEM.DEF.SVRCONN

Listener Port for Queue:

Name	SampleMDBQueueLP
Initial State	Started
Connection factory JNDI name	jms/SampleMDBConnectionFactory
Destination JNDI name	jms/SampleMDBQueue

Listener Port for Topic:

Name	SampleMDBTopicLP
Initial State	Stopped
Connection factory JNDI name	jms/SampleMDBConnectionFactory
Destination JNDI name	jms/SampleMDBTopic

NOTICE: A Listener Port in WebSphere Application Server is not the same as an MQ Listener. They are NOT related at all!

+ Instructions for creating the JMS Administrative objects.

Login with a user that has the authority to start the WebSphere Application Server server. We are using the default name of “server1”.

Open a command prompt. Issue the following WebSphere Application Server command:

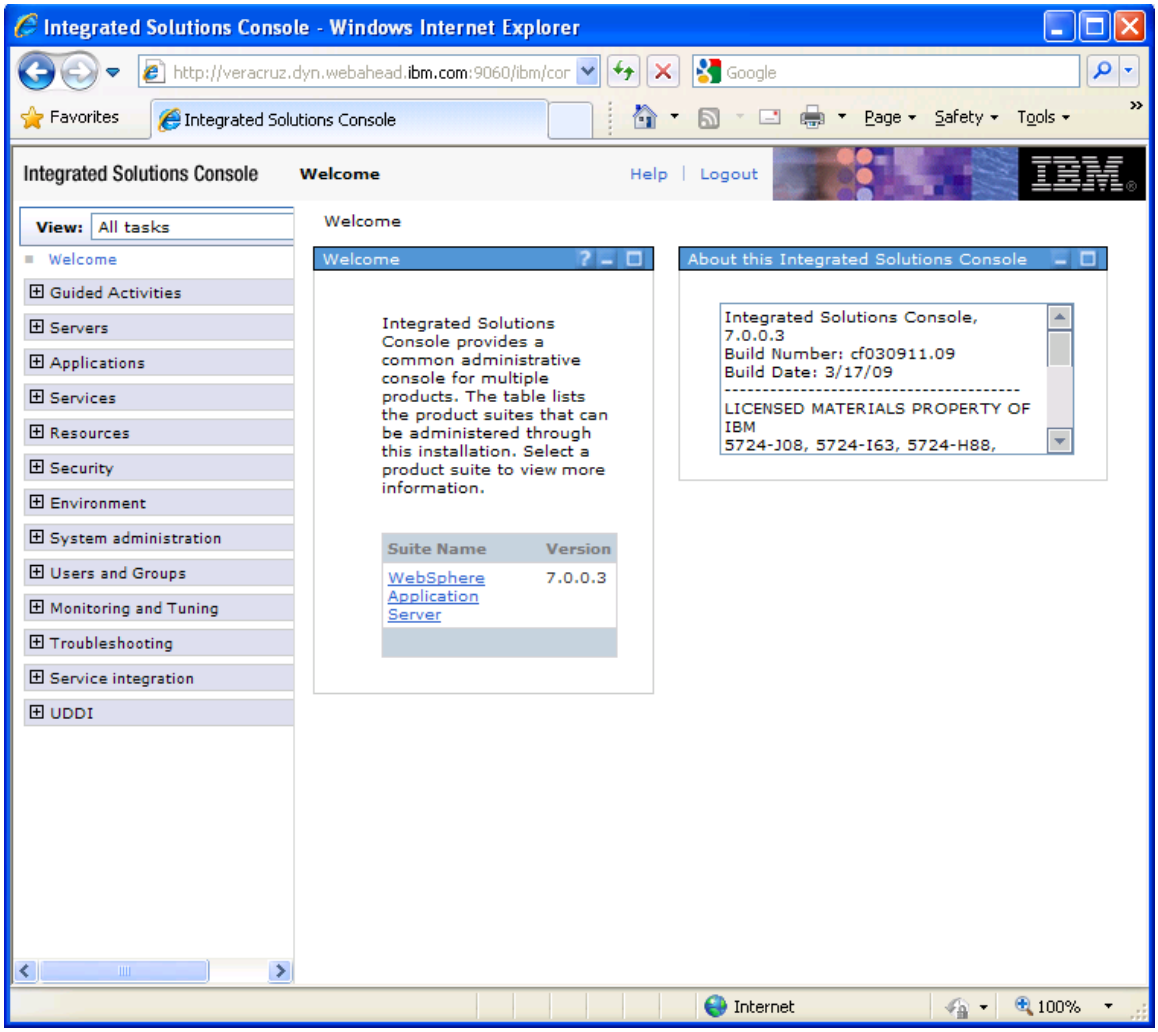
```
$ startServer.sh server1
```

Wait for the server to be active:

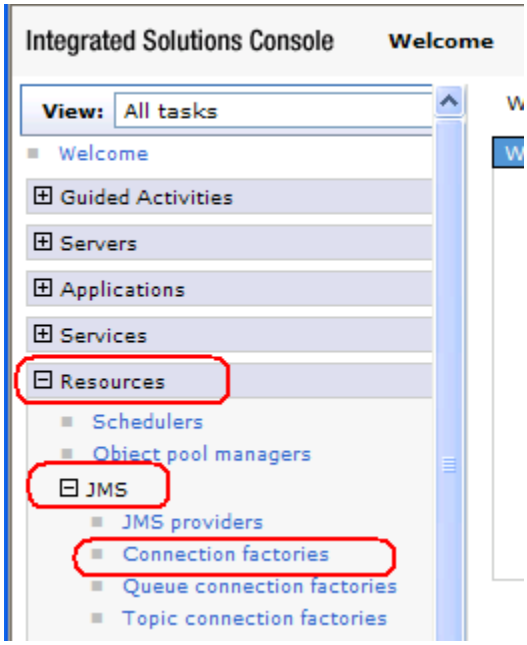
```
ADMU3000I: Server server1 open for e-business: process id is XXXX
```

Start a web browser and specify the URL for the WebSphere Application Server Administrative Console, such as:

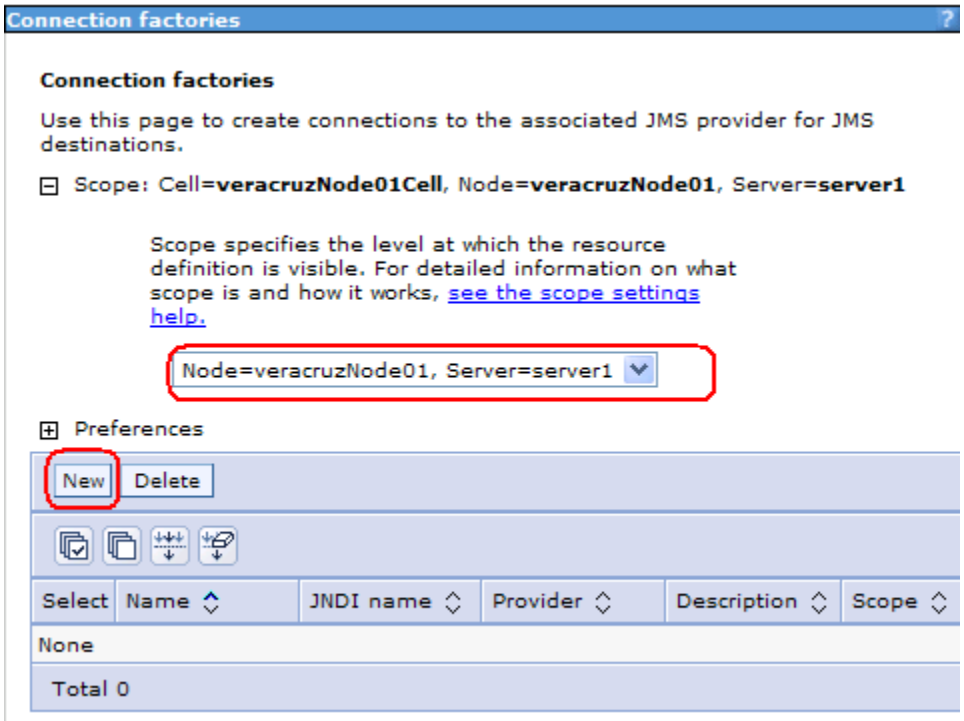
http://localhost:9060/ibm/console/unsecureLogon.jsp



- Create the following JMS admin object: Connection Factory
From the left panel, select: Resources > JMS > Connection factories



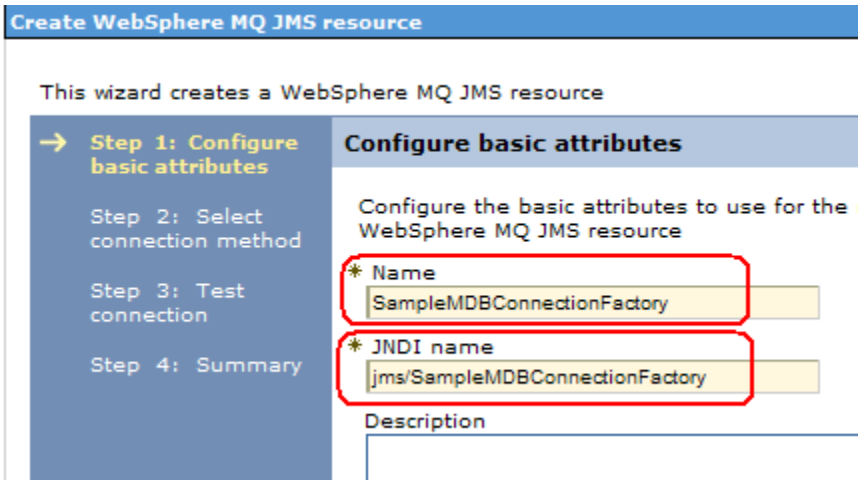
- The console will be refreshed. In the middle panel called "Connection factories" do:
1: Ensure that the scope is: Node=, Server=
2: Click on "New"



For the “JMS resource provider” specify:
WebSphere MQ messaging provider

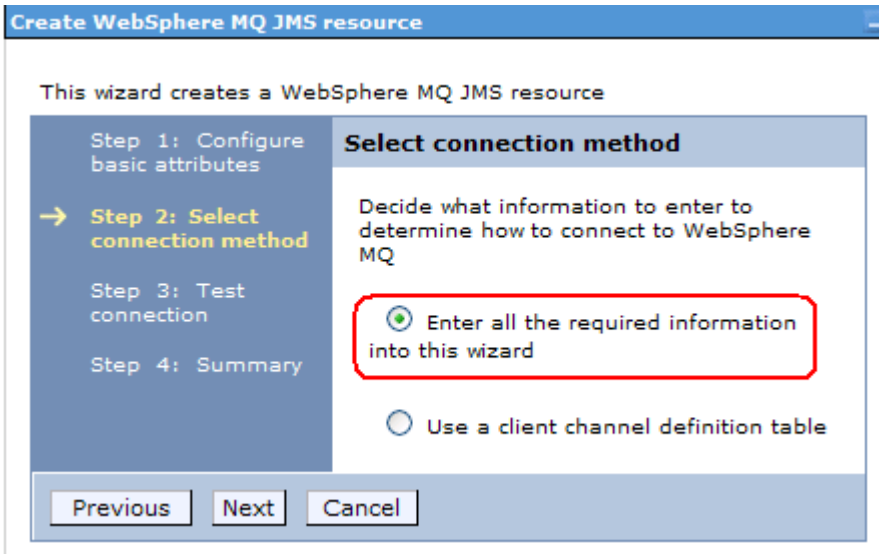
Click OK

In the “Step 1: Configure basic attributes” specify:
Name: SampleMDBConnectionFactory
JNDI Name: jms/SampleMDBConnectionFactory



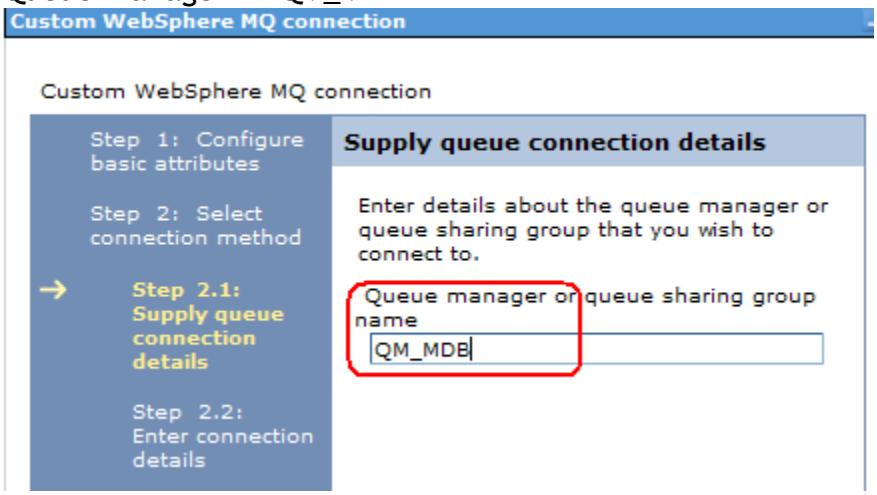
Click Next

In the “Step 2: Select connection method” accept the default:
Enter all the required information into this wizard



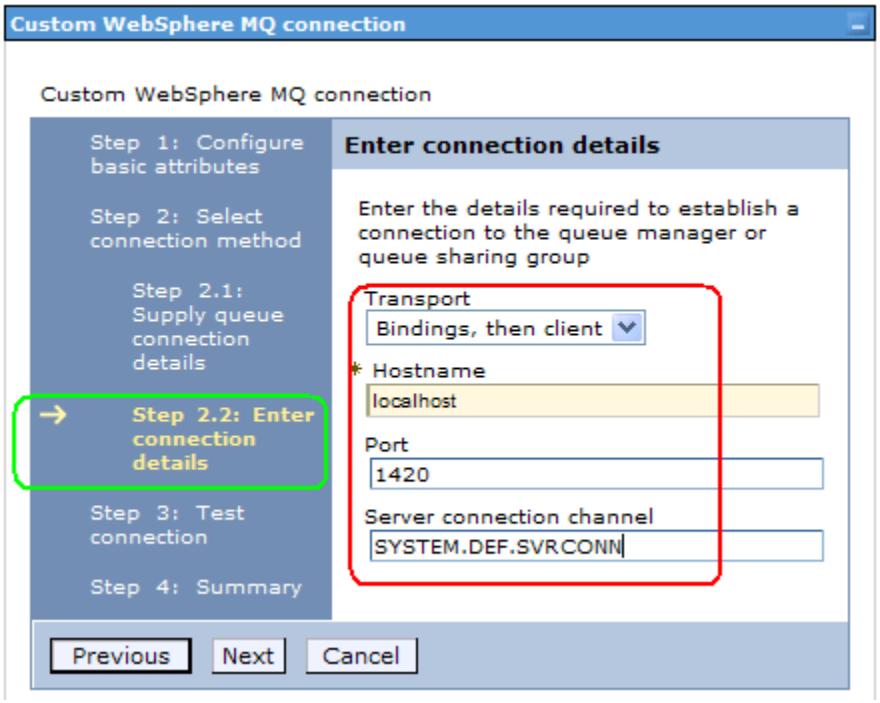
Click Next

In “Step 2.1: Supply queue connection details” (notice that it really means: supply details for the connection to the queue manager) specify:
Queue manager => QM_MDB

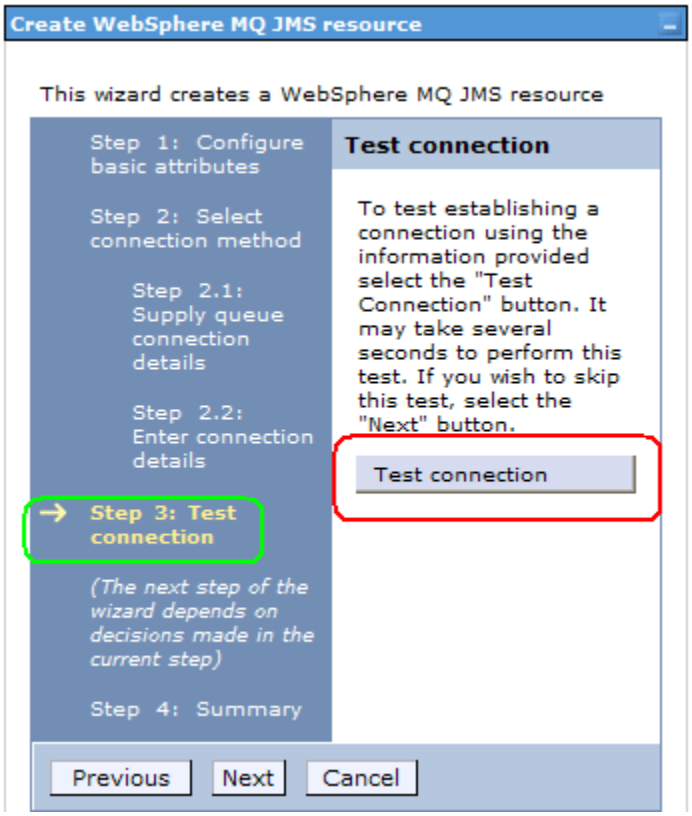


In “Step 2.2: Enter connection details” specify:
Transport => Bindings, then client
Hostname => localhost
Port => 1420
Server connection channel => SYSTEM.DEF.SVRCONN

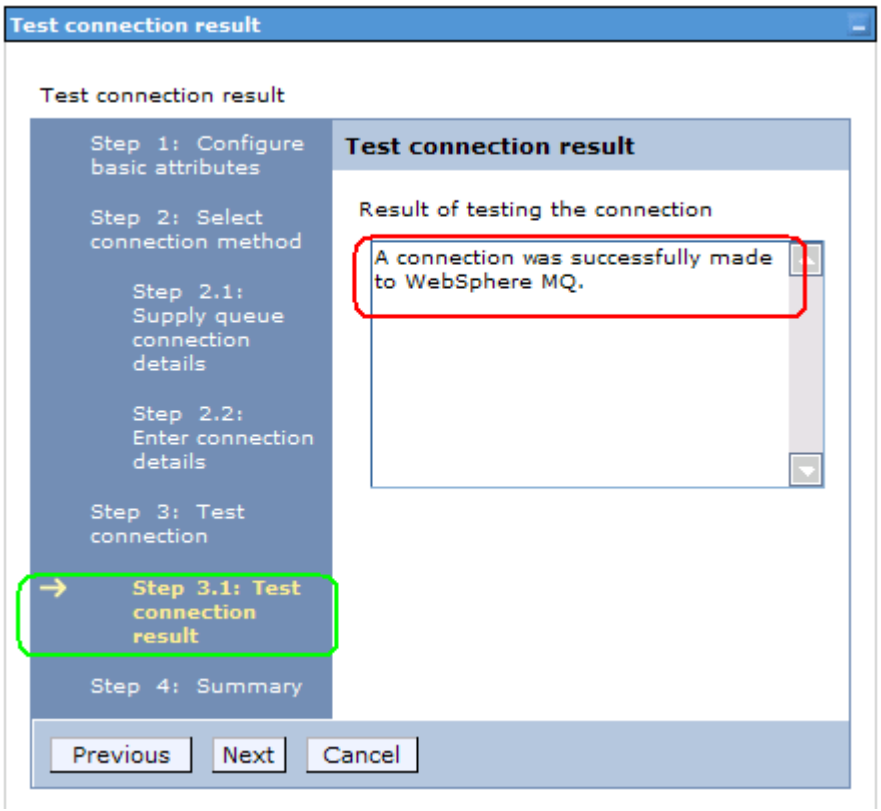
Note: the default port number for MQ is 1414. The port 1420 is used here as an example.



In “Step 3: Test Connection” click on the button “Test connection”

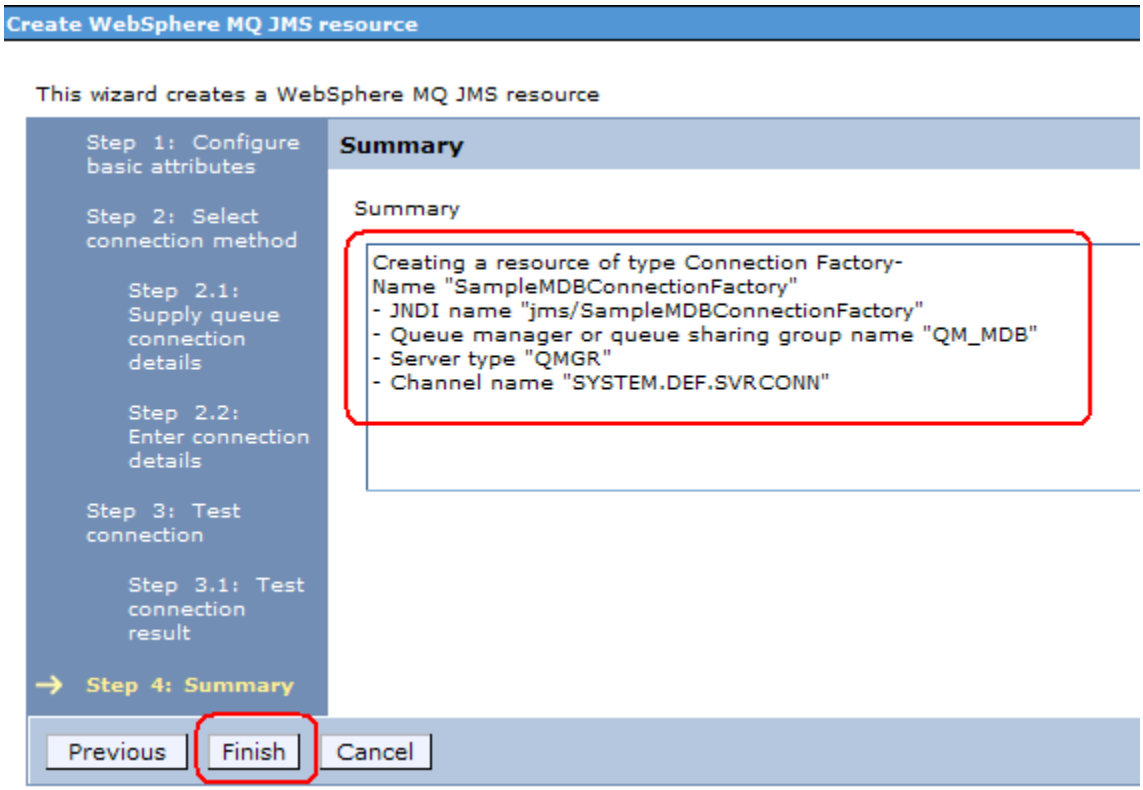


In “Step 3.1: Test connection result”, you should see a successful report:
“A connection was successfully made to WebSphere MQ”



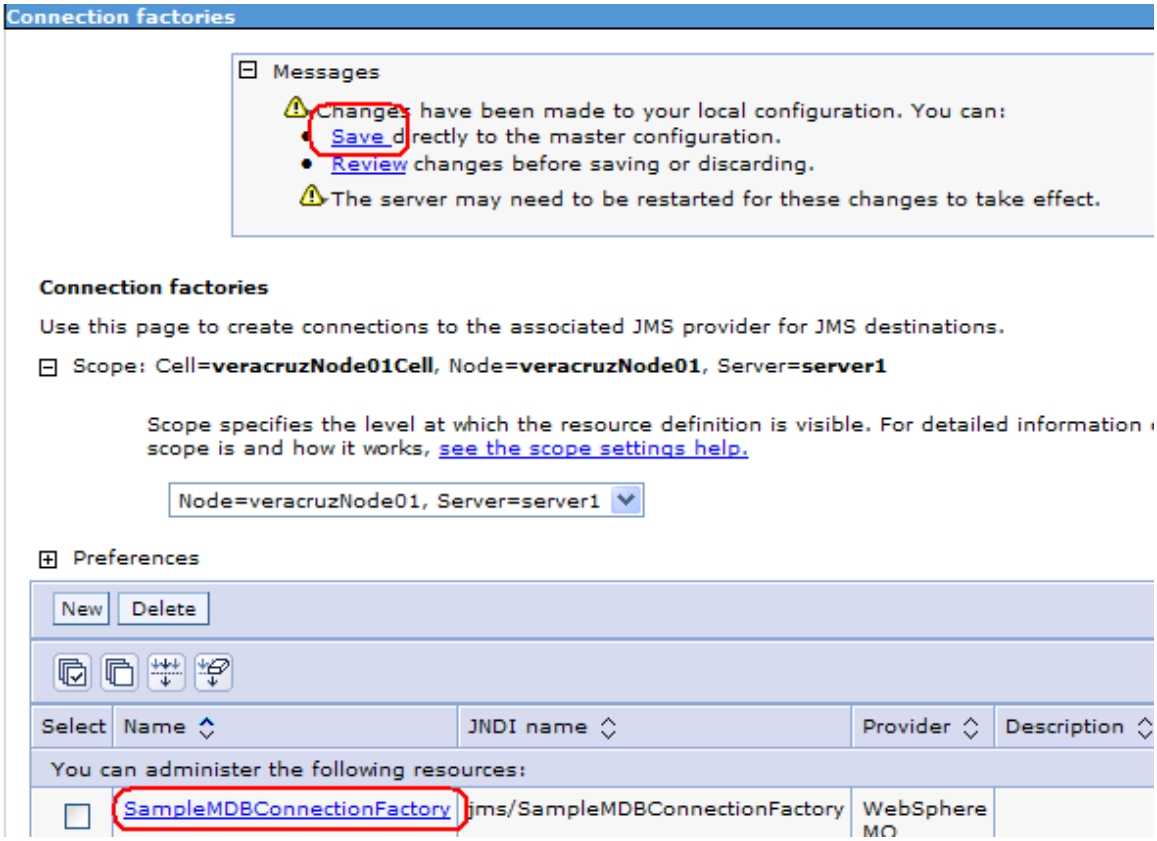
Note:
If you do NOT have a successful result at this point, then STOP!
Fix any problems before you proceed.

In “Step 4: Summary” you should see the data to be used to create the object.



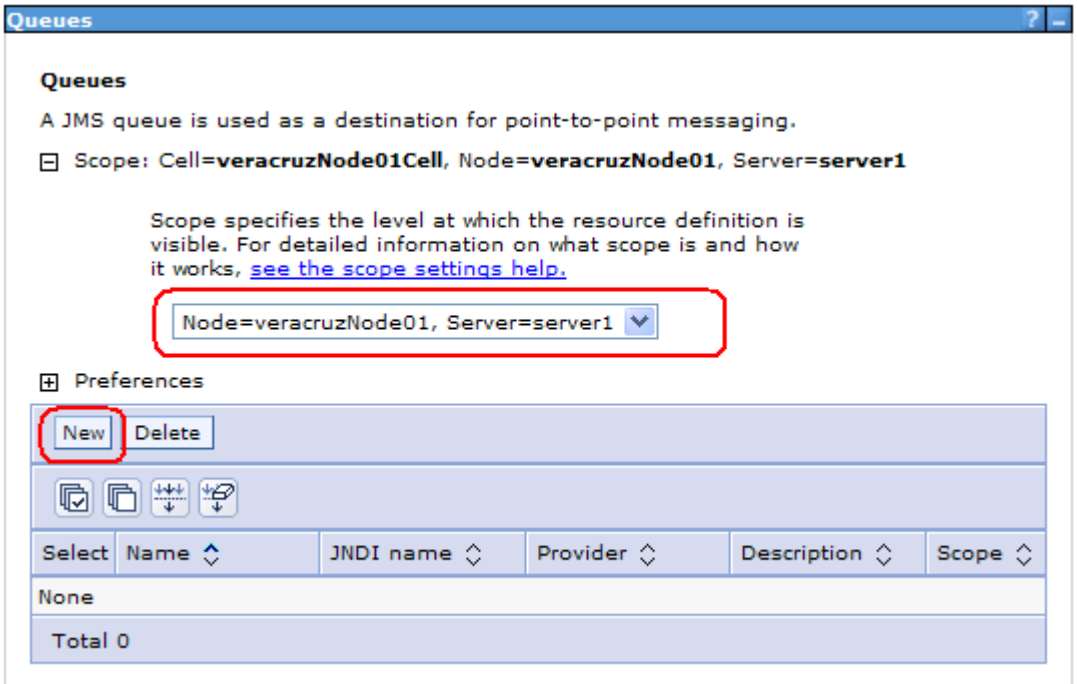
Click on Finish.

After this JMS object is created, you should see a new connection factory. Click on “Save”.



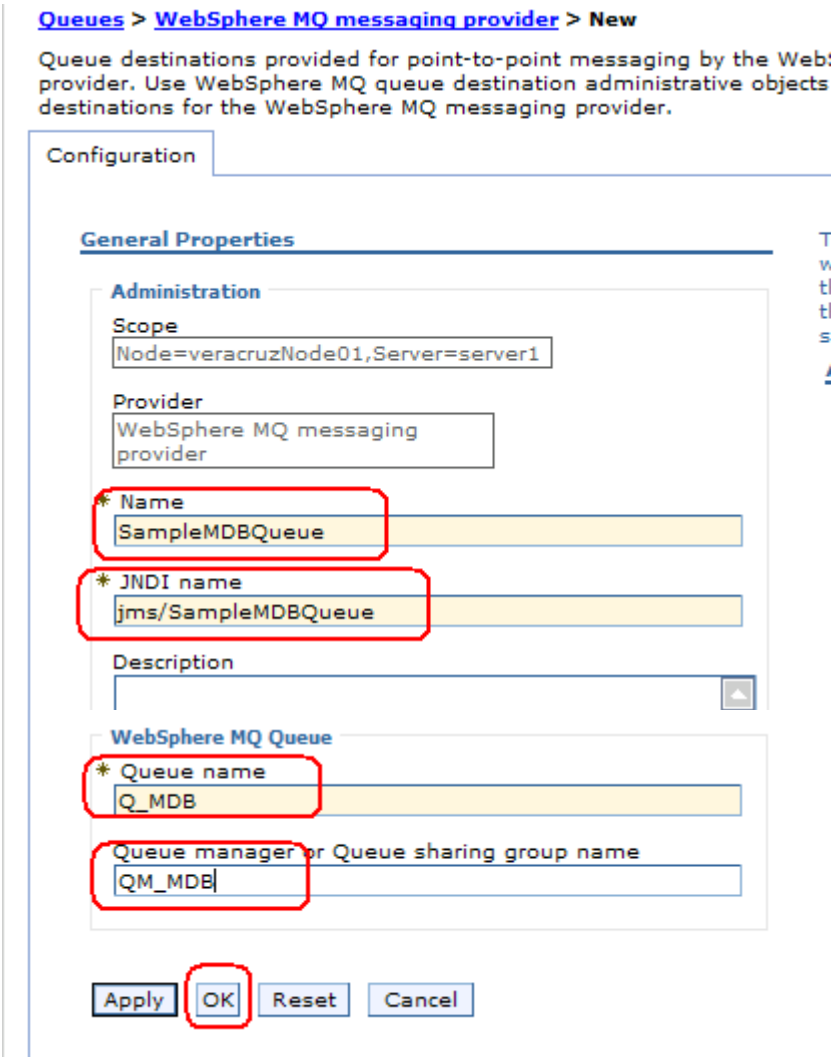
- Create the new Queue Destination by clicking on:
Resources > JMS > Queues

From the “Queues” window:
1: Ensure that the scope is: Node=, Server=
2: Click on “New”



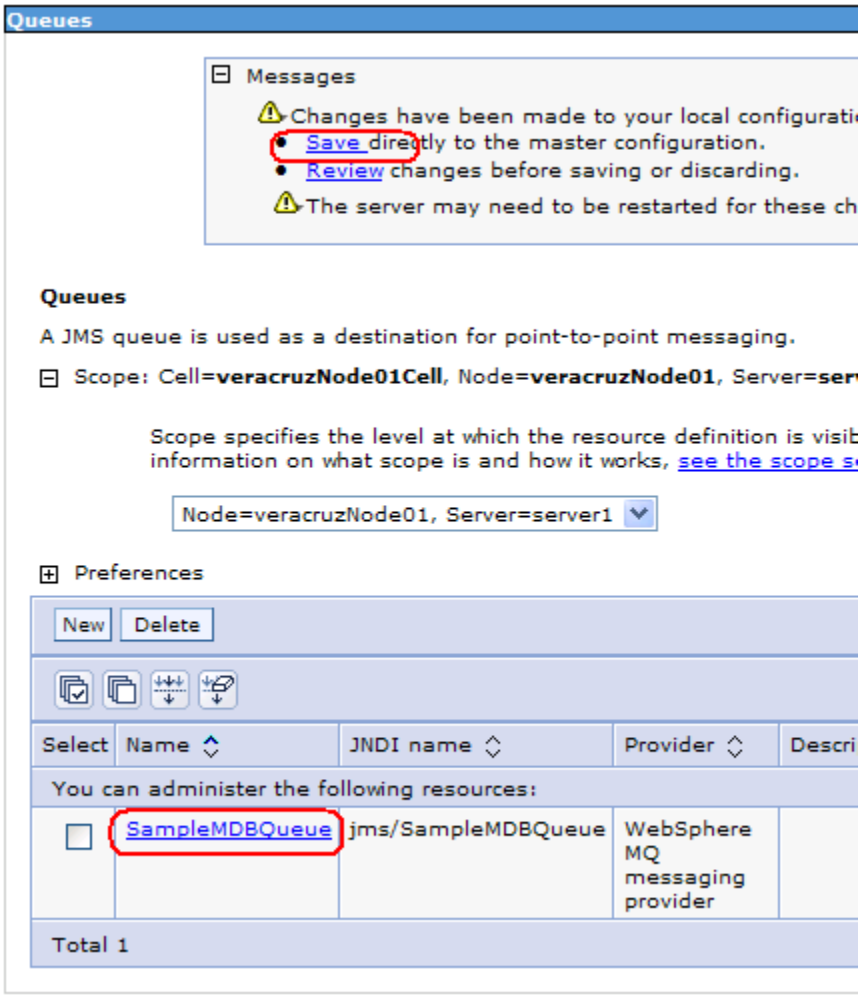
For the “JMS resource provider” specify:
WebSphere MQ messaging provider
Click on OK.

In the Configuration tab, specify:
Name: SampleMDBQueue
JNDI Name: jms/SampleMDBQueue
Queue Name in MQ: Q_MDB
Queue Manager: QM_MDB



Click OK.

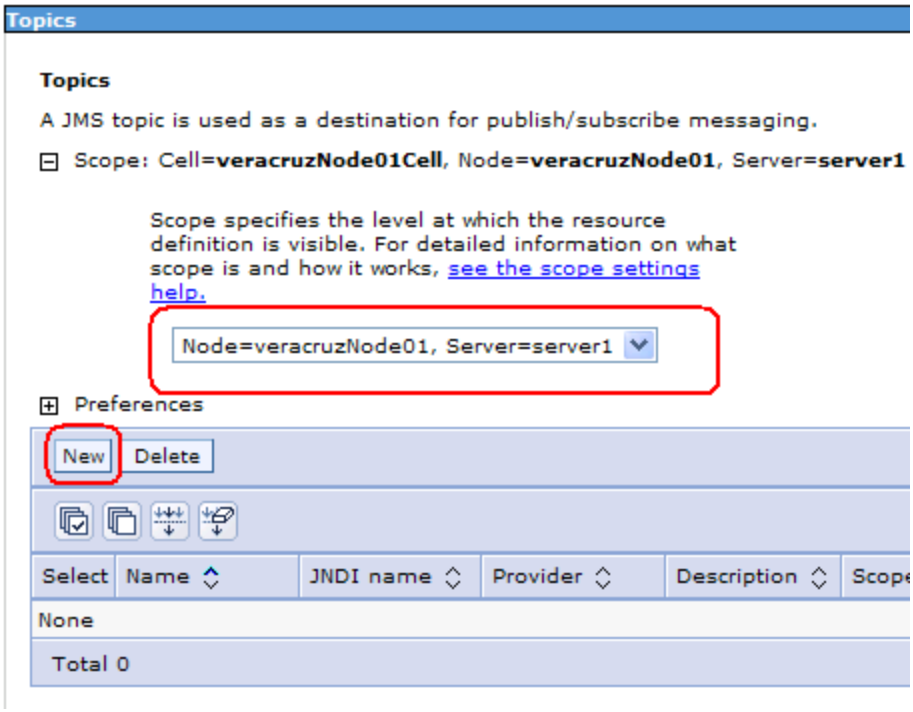
You will see a new JMS Queue:



Click on Save.

- Create the new Topic Destination by clicking on:
Resources > JMS > Topics

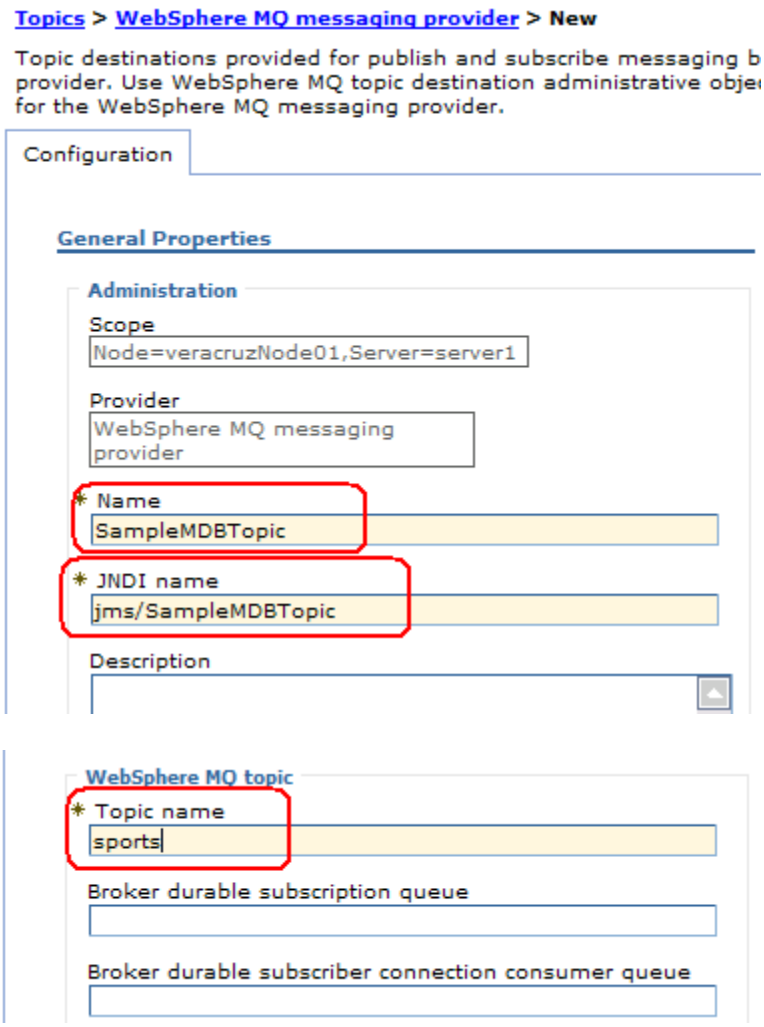
From the “Topics” window:
1: Ensure that the scope is: Node=, Server=
2: Click on “New”



For the “JMS resource provider” specify:
WebSphere MQ messaging provider
Click on OK.

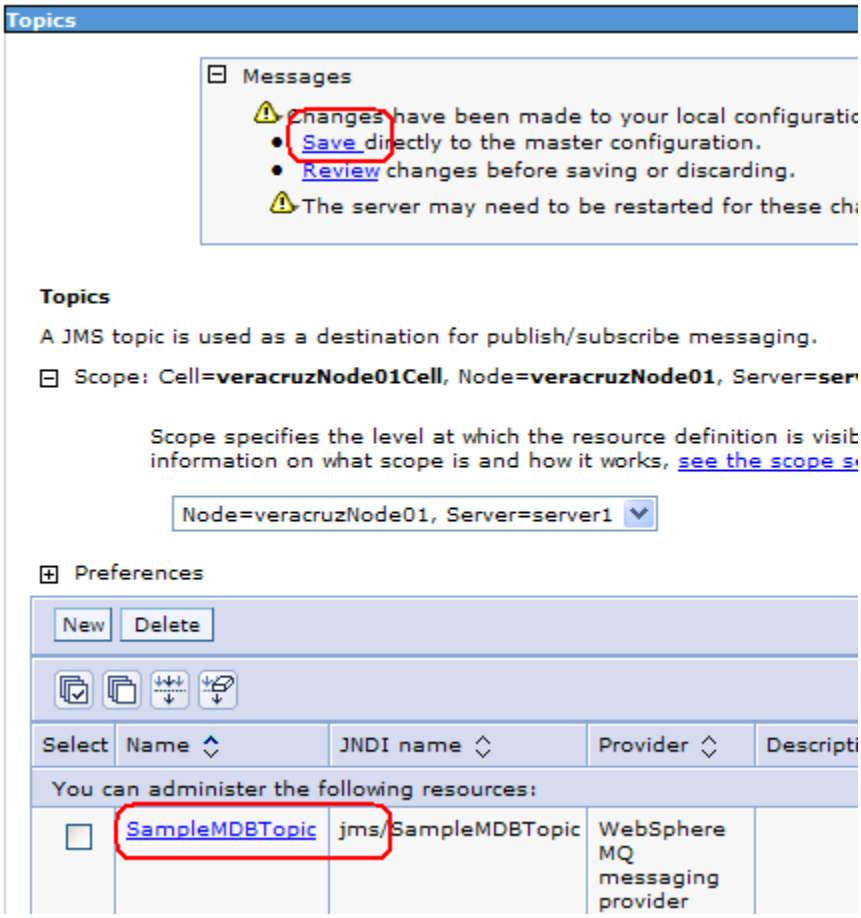
In the Configuration tab, specify:
Name: SampleMDBTopic
JNDI Name: jms/SampleMDBTopic
Topic Name in MQ: sports
Broker publication queue manager: QM_MDB

Note:
Notice that the WebSphere Application Server field “Topic Name in MQ” refers to the Topic String (in this example, “sports”) and not to the Topic Object (in this example “T_MDB”).



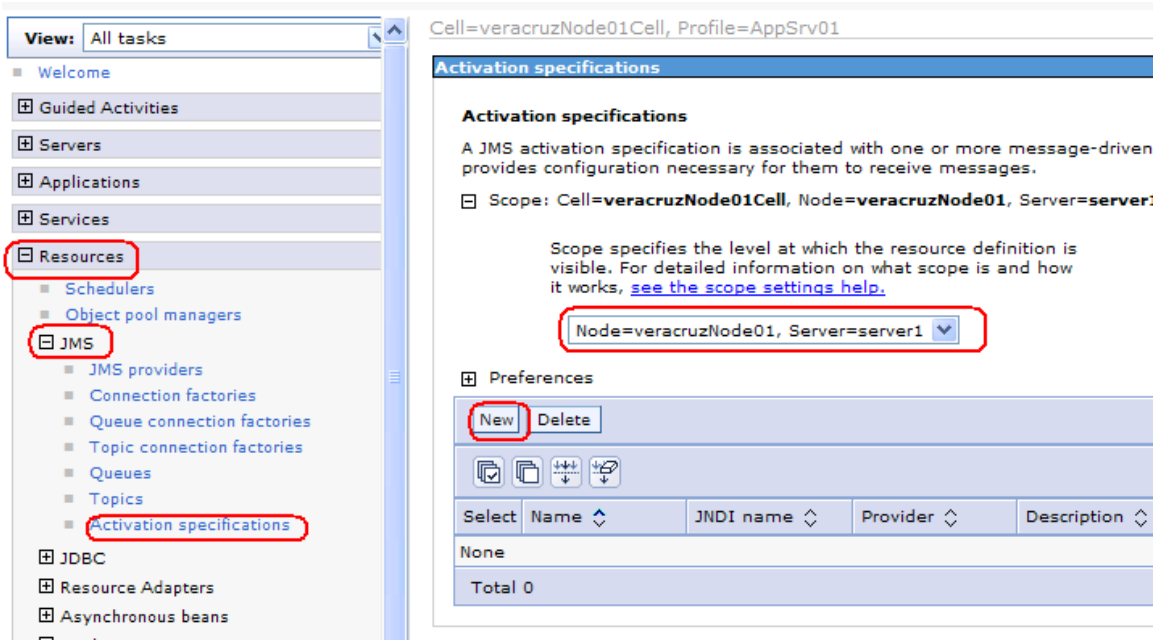
Click OK.

You will see a new JMS Topic:

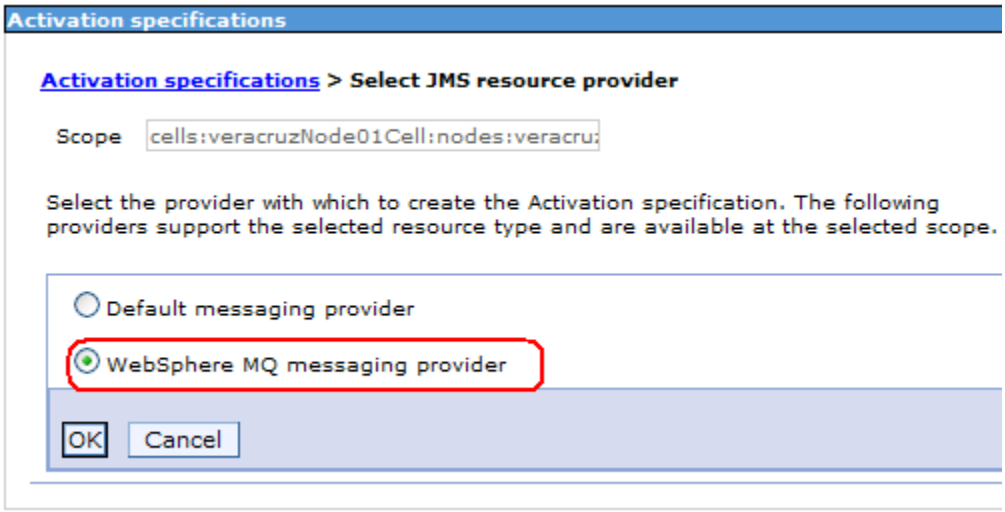


Click on Save.

- Create the Activation Specifications.
From the left panel, select: Resources > JMS > Activation specifications
Then in the “Activation specifications” window, select the scope of “Node,Server” and click on New.

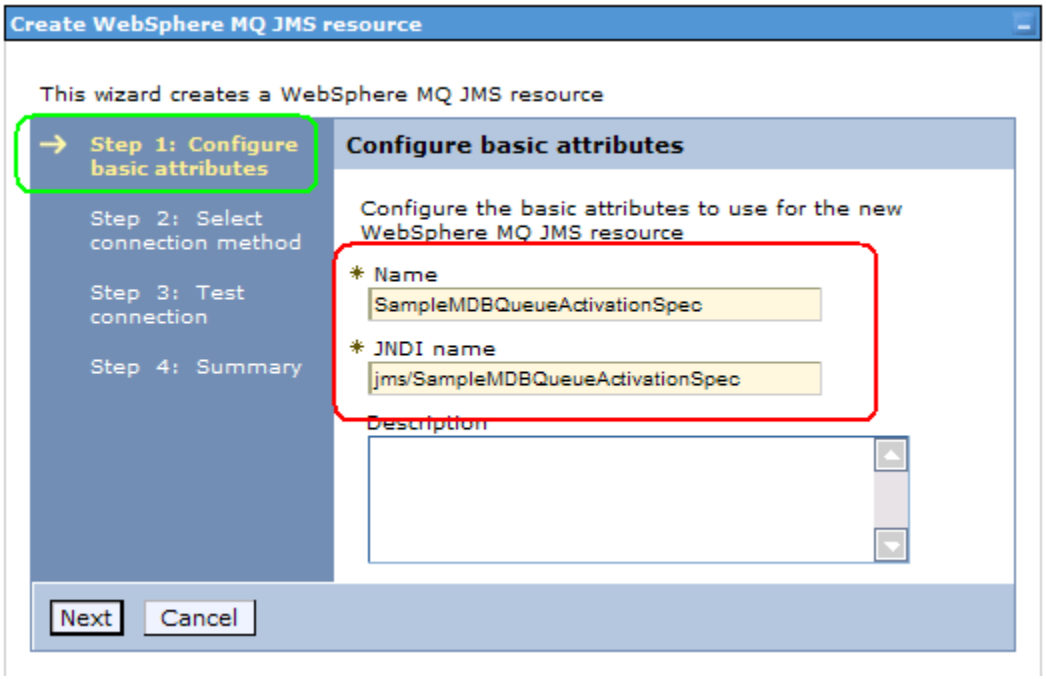


Ensure that the JMS resource provider is:
WebSphere MQ messaging provider



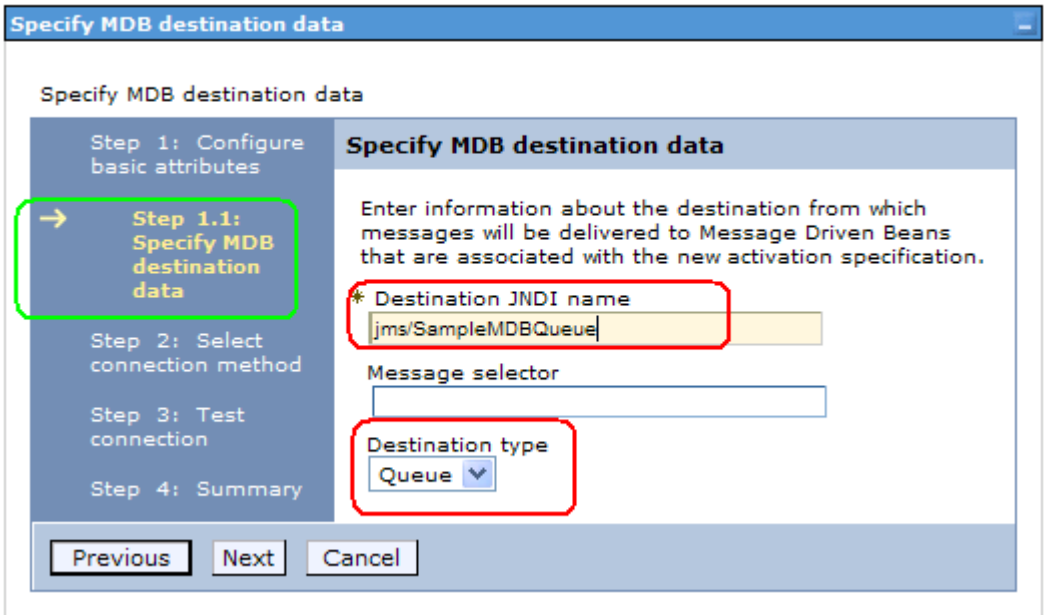
Click on OK.

Specify the following for the new Activation Specification for a Queue:
Name: SampleMDBQueueActivationSpec
JNDI Name: jms/SampleMDBQueueActivationSpec



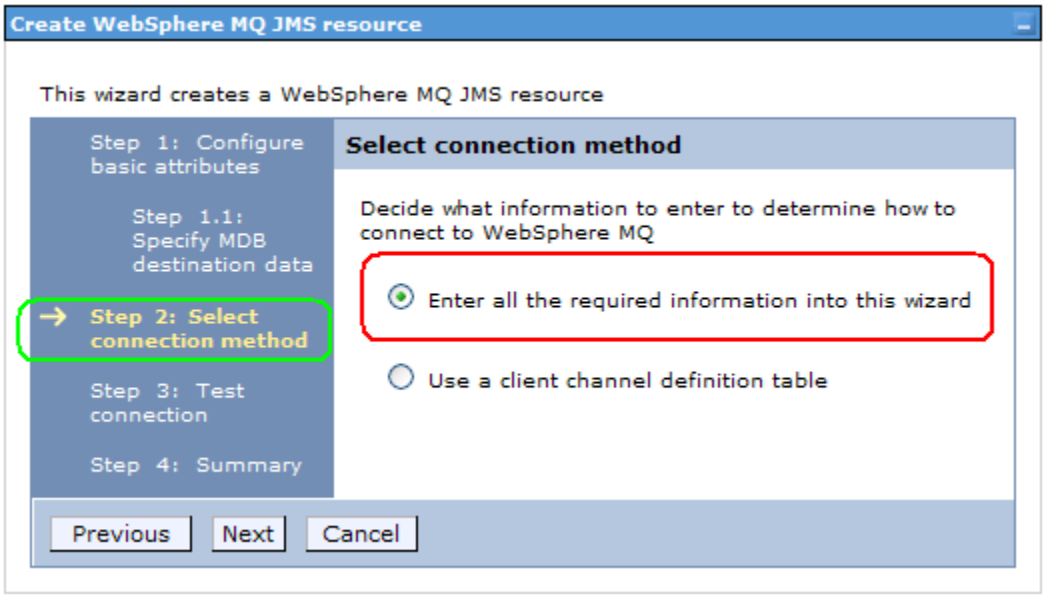
Destination JNDI name: jms/SampleMDBQueue
Destination Type: Queue

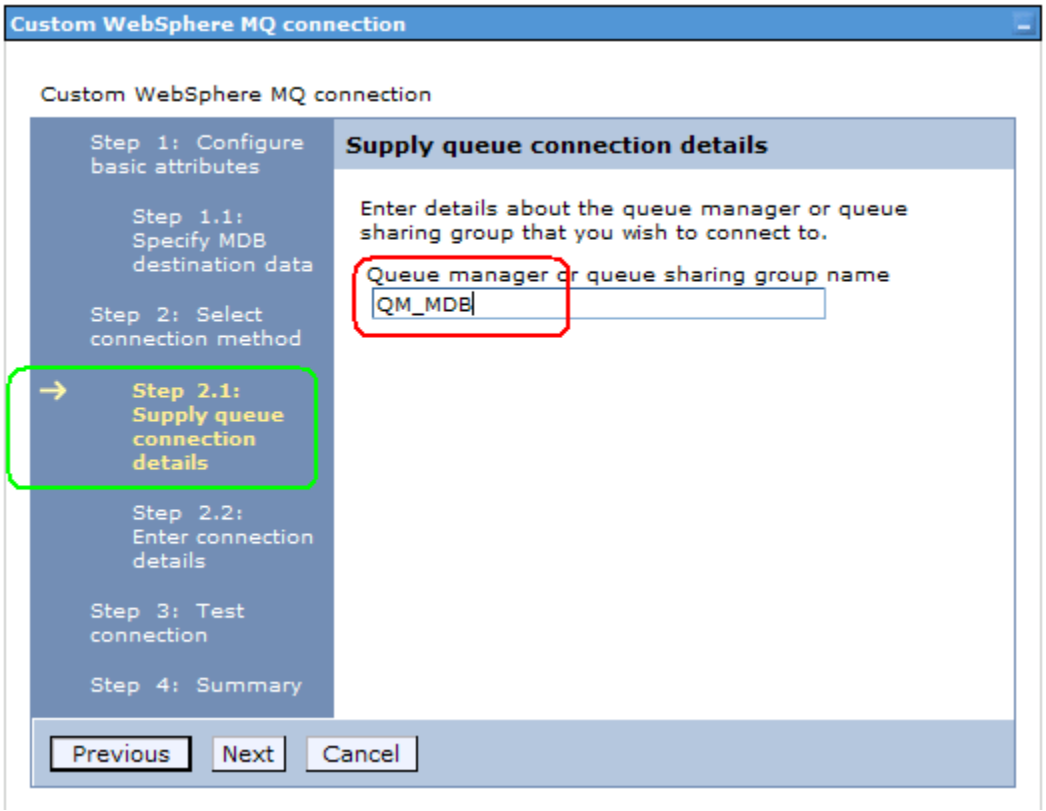
Specify MDB destination data



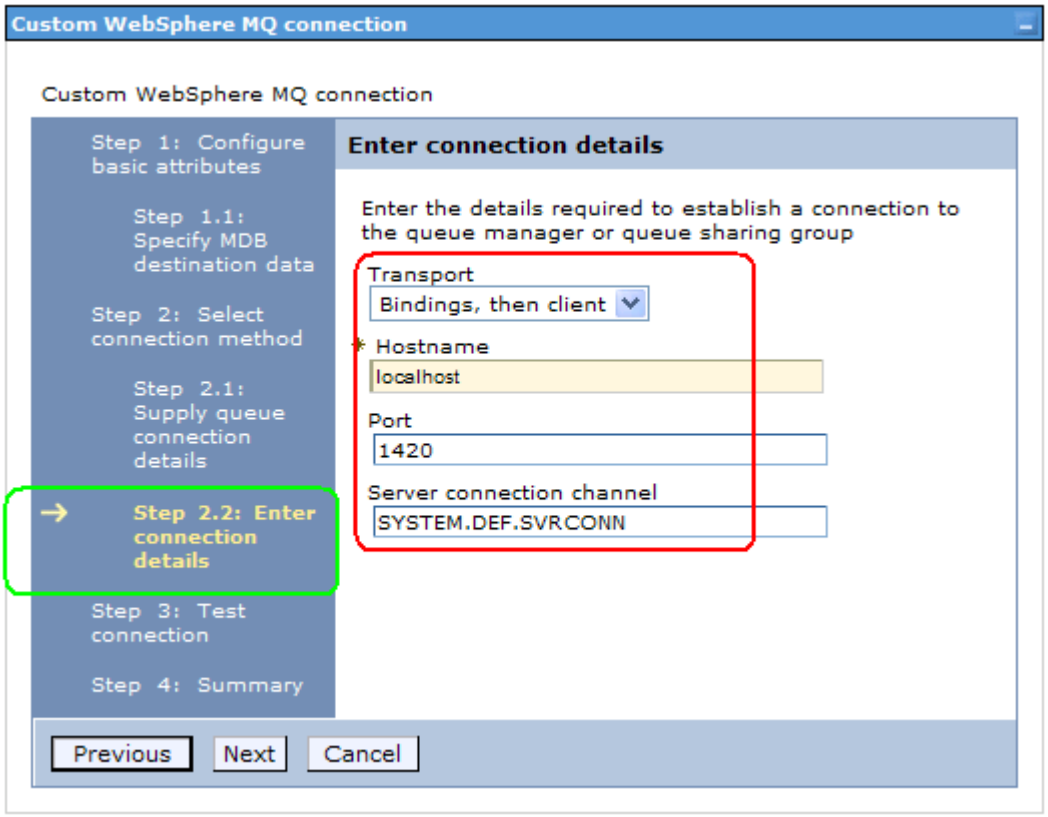
Specify:

Queue manager: QM_MDB





Transport :	Bindings, then client
Hostname:	localhost
Port:	1420
Server connection channel:	SYSTEM.DEF.SVRCONN

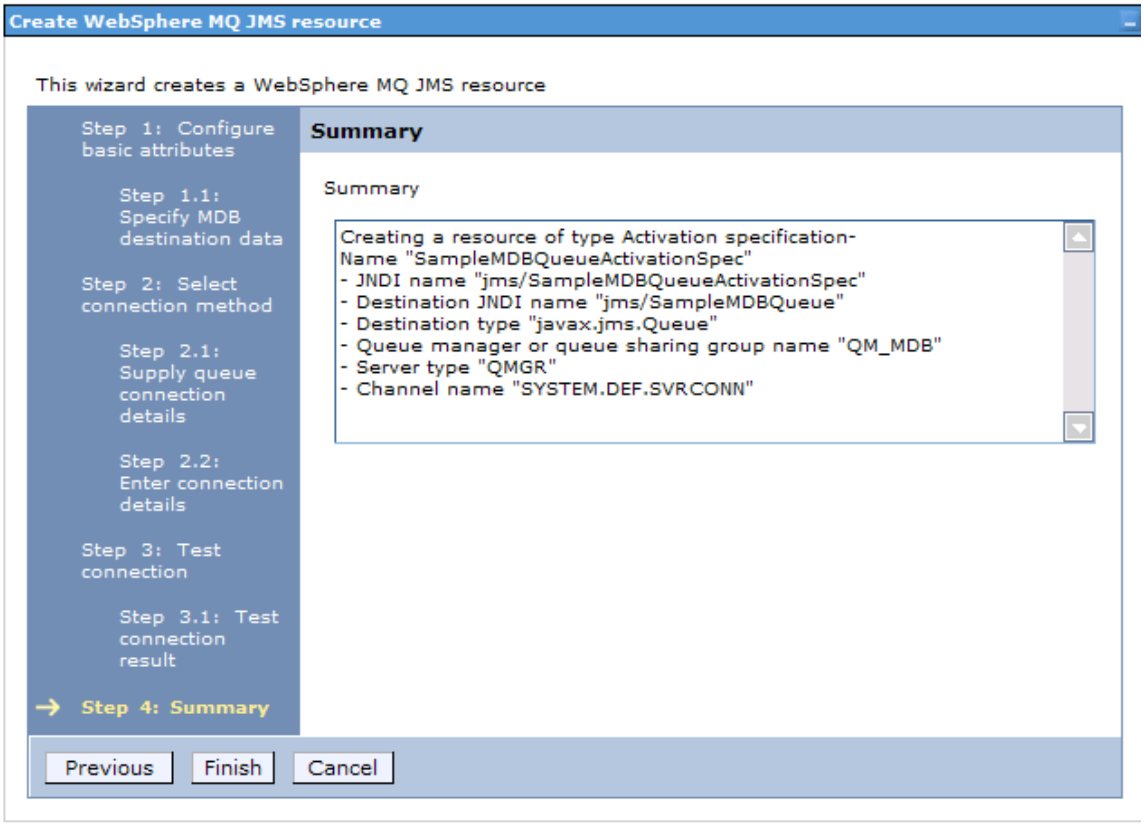


In “Step 3: Test Connection” click on the button “Test connection”.

In “Step 3.1: Test connection result”, you should see a successful report: “A connection was successfully made to WebSphere MQ”

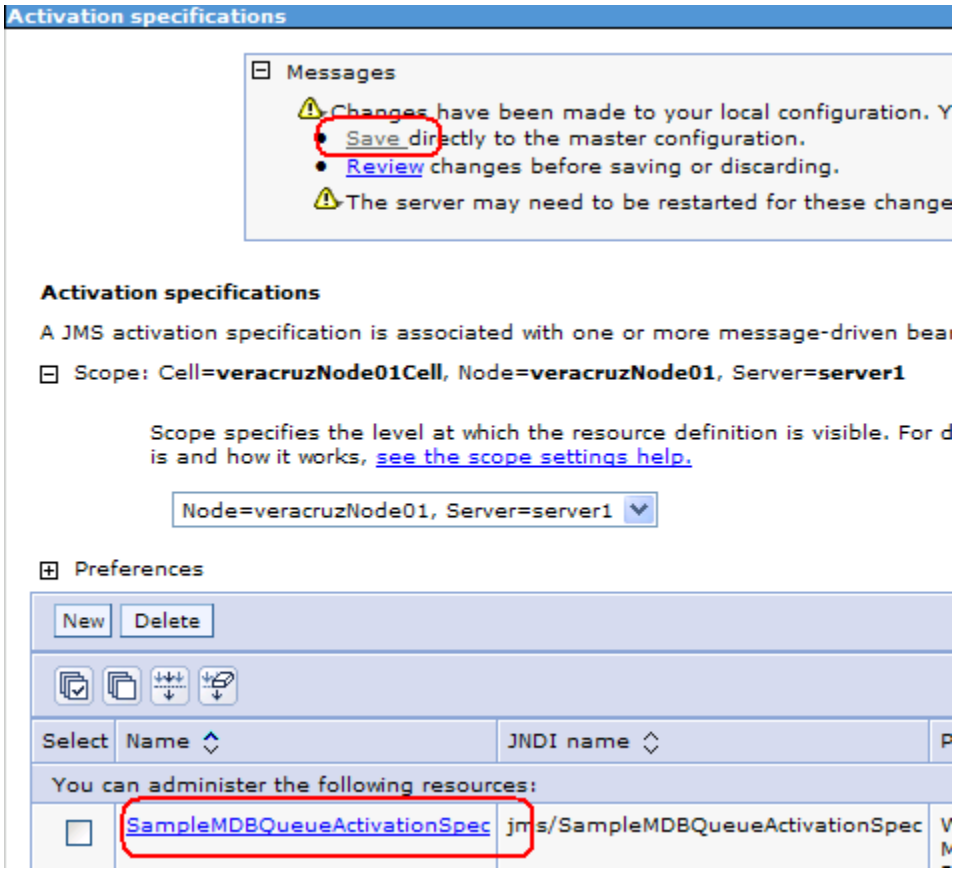
+++ Note: If you do NOT have a successful result at this point, then STOP! Fix any problems before you proceed.

You should see the summary for the object to be created:



Click on Finish.

Once the Activation Specification is created, it will be added to the list:



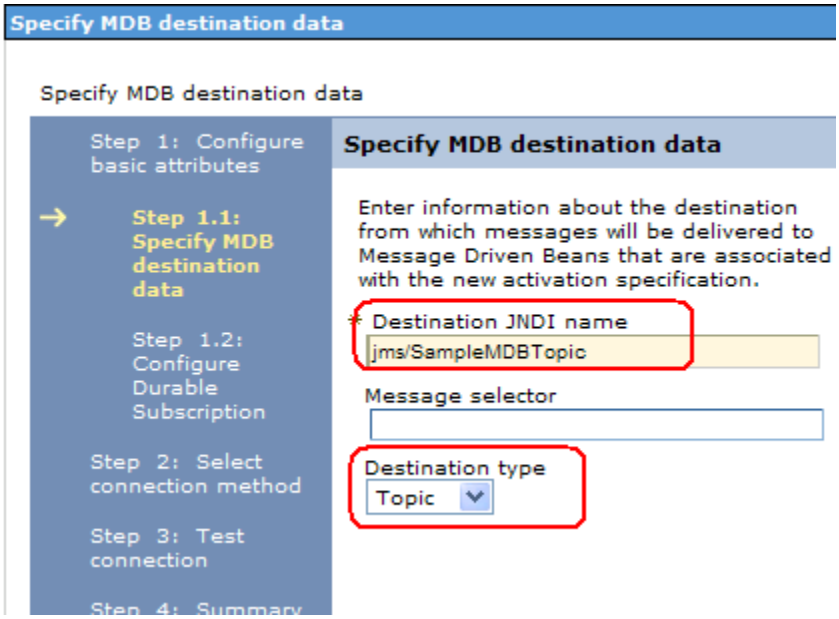
Click on Save.

Repeat the steps to create another Activation Specification, but this time for the topic.

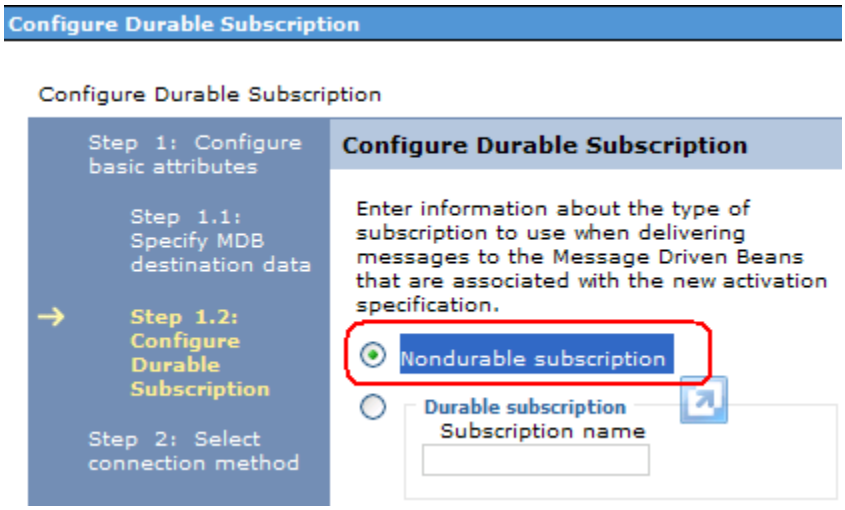
Activation Specification for a Topic:

Name:	SampleMDBTopicActivationSpec
JNDI Name:	jms/SampleMDBTopicActivationSpec
Destination JNDI name:	jms/SampleMDBTopic
Destination Type:	Topic
Type of subscription:	Nondurable subscription
Queue manager:	QM_MDB
Transport :	Bindings, then client
Hostname:	localhost
Port:	1420
Server connection channel:	SYSTEM.DEF.SVRCONN

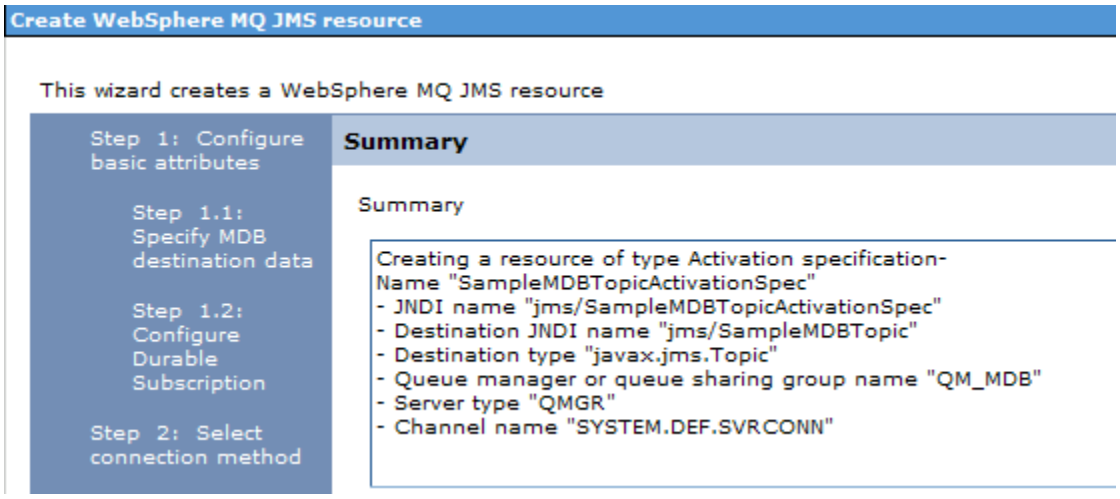
For simplicity, only the screenshots relevant for a topic be shown.



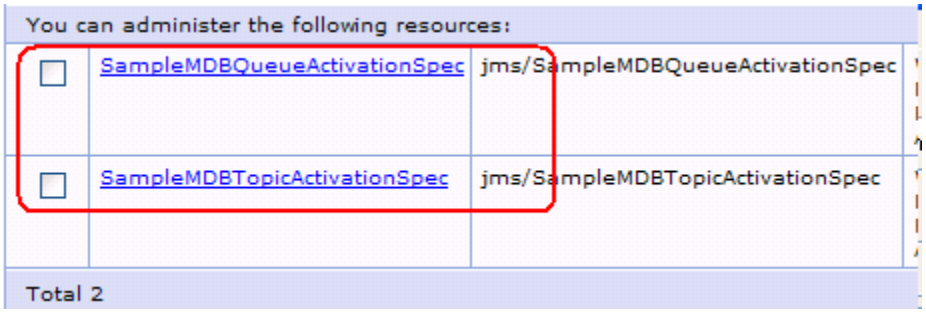
Accept the default of “Nondurable subscription”



… continue with the rest of the screens.

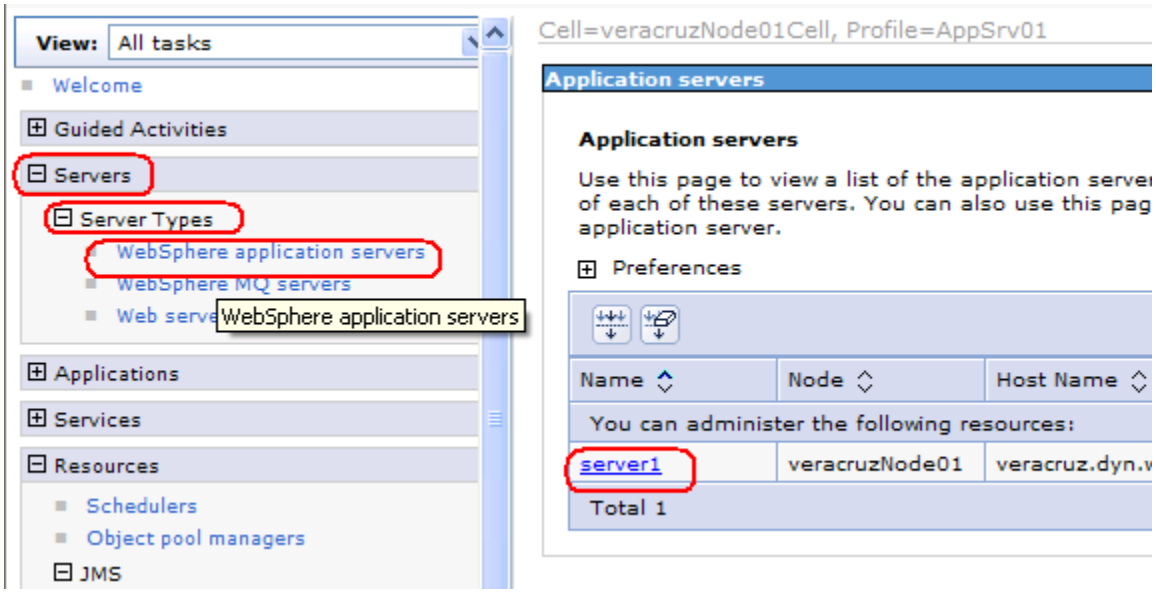


Click on Finish then Save.
You should see now 2 activation specifications:



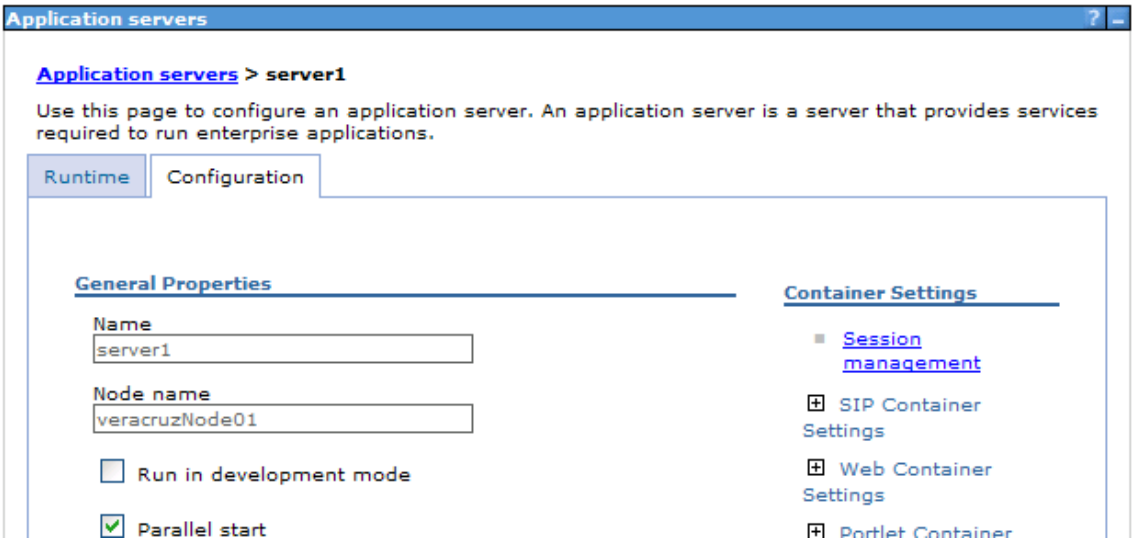
- Create a Listener Port in WebSphere Application Server.

A WebSphere Application Server Listener Port is not a JMS administrative object and thus, it is not stored in the JNDI directory service. Rather, it is an object under the WebSphere Application Server “server”.
From the left panel, select:
Servers > Server Types > WebSphere application servers



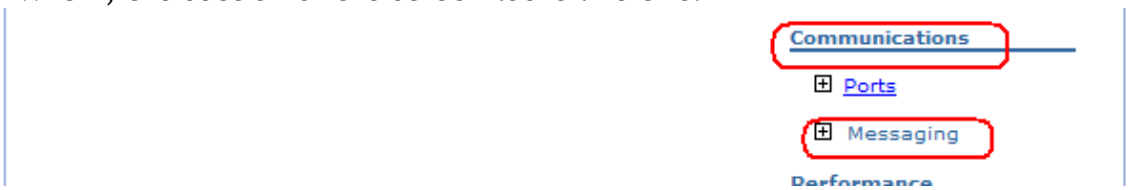
Click on “server1”.

The Configuration tab will display the main information on the server:



You need to scroll down to reach the section “Communications”

In WAS 7, the section of the screen looks like this:



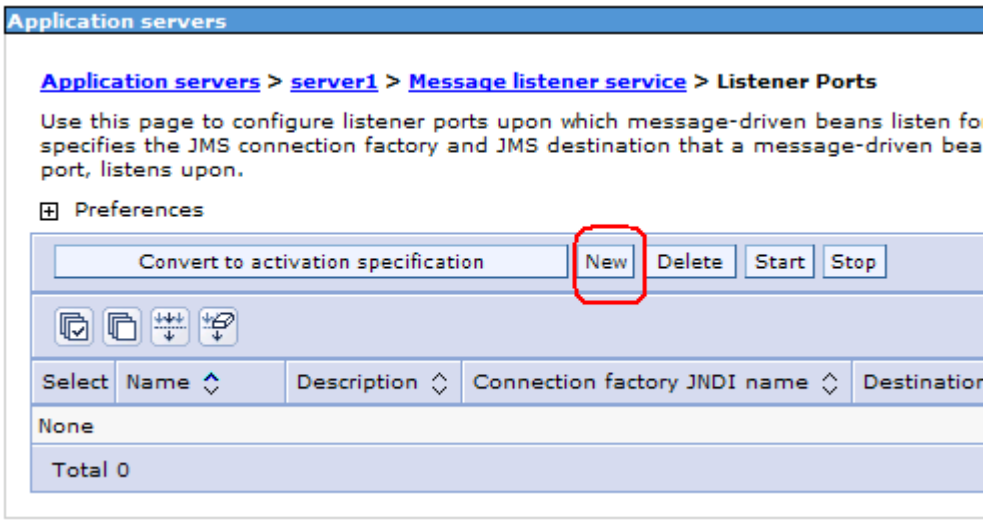
In WAS 8.x, the section of the screen looks like this:

Click on “Messaging”

Click on “Message listener service”

In the “Message listener service” specify: Listener Ports

From the window for “Listener Ports” click on New.



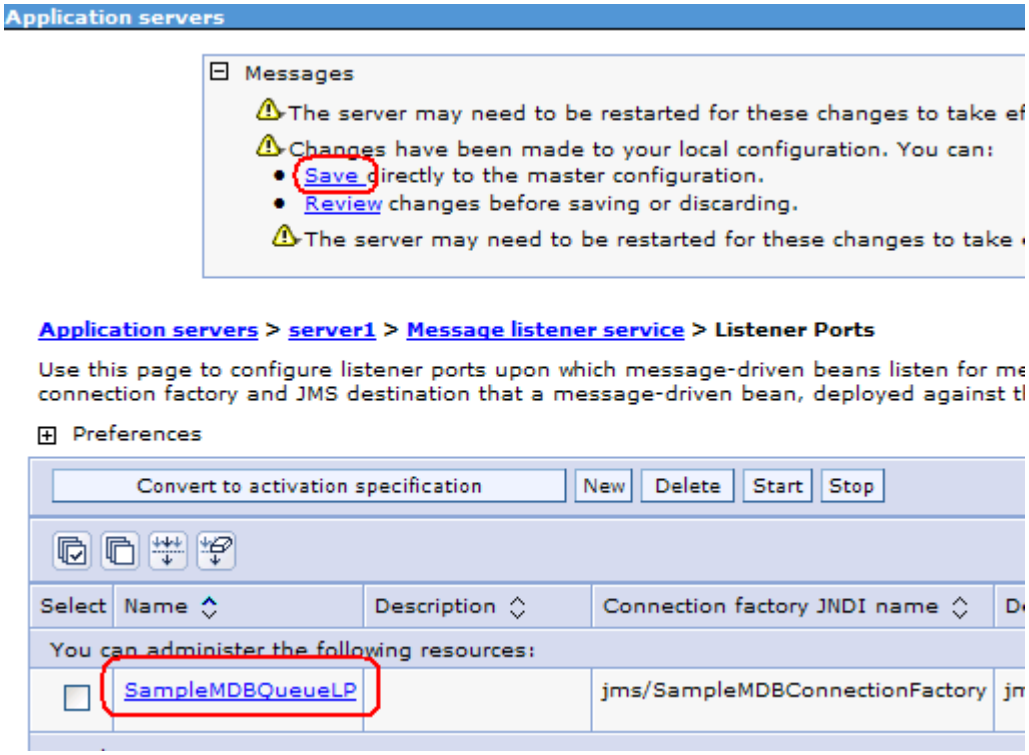
In the Configuration page, specify the following for the ListenerPort:

Name	SampleMDBQueueLP
Connection factory JNDI name	jms/SampleMDBConnectionFactory
Destination JNDI name	jms/SampleMDBQueue

Accept the default for Initial State: Started

Click OK

Once the Listener Port is created, you will see it in the list.
Click on Save.



Click on Save.

Create another Listener Port, but this time for the Topic.

Name	SampleMDBTopicLP
Initial State:	Stopped
Connection factory JNDI name	jms/SampleMDBConnectionFactory
Destination JNDI name	jms/SampleMDBTopic

Note:
To simplify our testing, ensure that this Listener Port is not running when the server is restarted. The "Initial State" for this LP should be: Stopped
We want to test first the Listener Port for the Queue and we want to avoid collisions. Later on we will discuss a scenario when the Topics will be used and we will stop the Listener Port for the Queue and start the one for the Topic

Click OK and click Save.

You should see 2 Listener Ports.
None of them should BE RUNNING at this point.

Select	Name	Description	Connection factory JNDI name	Destination JNDI name	Sta
You can administer the following resources:					
☐	SampleMDBQueueLP		jms/SampleMDBConnectionFactory	jms/SampleMDBQueue	
☐	SampleMDBTopicLP		jms/SampleMDBConnectionFactory	jms/SampleMDBTopic	

Good! We have finished adding all the JMS administration objects in the JNDI controlled by WebSphere Application Server, as well as the Listener Port.

Note:
Any additions/deletions to the JNDI directory service of the App Server require a reboot of the server: thus, logout from the Administrative Console.

Then login as root and stop and restart the server:

```
# stopServer.sh server1

# startServer.sh server1
```

Look at the SystemOut.log and SystemErr.log files to see if there are any warnings or error related to the JMS objects or the listener ports that we defined. The default location for these logs is:

/opt/IBM/WebSphere/AppServer/profiles/AppSrv01/logs/server1

Even though we are not going to use the listener ports right away, let's verify that the Listener Port for a Queue (SampleMDBQueueLP) is shown as running.

From the WebSphere Application Server Administrative Console, follow the steps mentioned earlier to access the screen for the Listener Ports for server1. In this example, one listener port is started (SampleMDBQueueLP) and the other is stopped (SampleMDBTopicLP).

Select	Name	Description	Connection factory JNDI name	Destination JNDI name	Status
You can administer the following resources:					
☐	SampleMDBQueueLP		jms/SampleMDBConnectionFactory	jms/SampleMDBQueue	▶
☐	SampleMDBTopicLP		jms/SampleMDBConnectionFactory	jms/SampleMDBTopic	✖
Total 2					

Note:
In case that SampleMDBQueueLP is not running, then look at the WebSphere Application Server logs (mentioned above) for errors. One possible situation is that the MQ queue manager for which the Listener Port is trying to connect ("QM_MDB" in this example) is not running, in which case, start the queue manager and restart the WebSphere Application Server Listener Port (select the check box and then click on Start from the screen mentioned above).

Once you have verified that there are no errors and that the Listener Port SampleMDBQueueLP is running, proceed to deploy and test the MDB.

++
+++ Chapter 3: WebSphere Application Server V7 deployment and testing of MDB
++

The purpose of this chapter is to show the steps to deploy an EAR file that has a simple MDB and to test the MDB: we will put a message into the queue and the MDB should receive the message and write the message contents into the SystemOut.log:

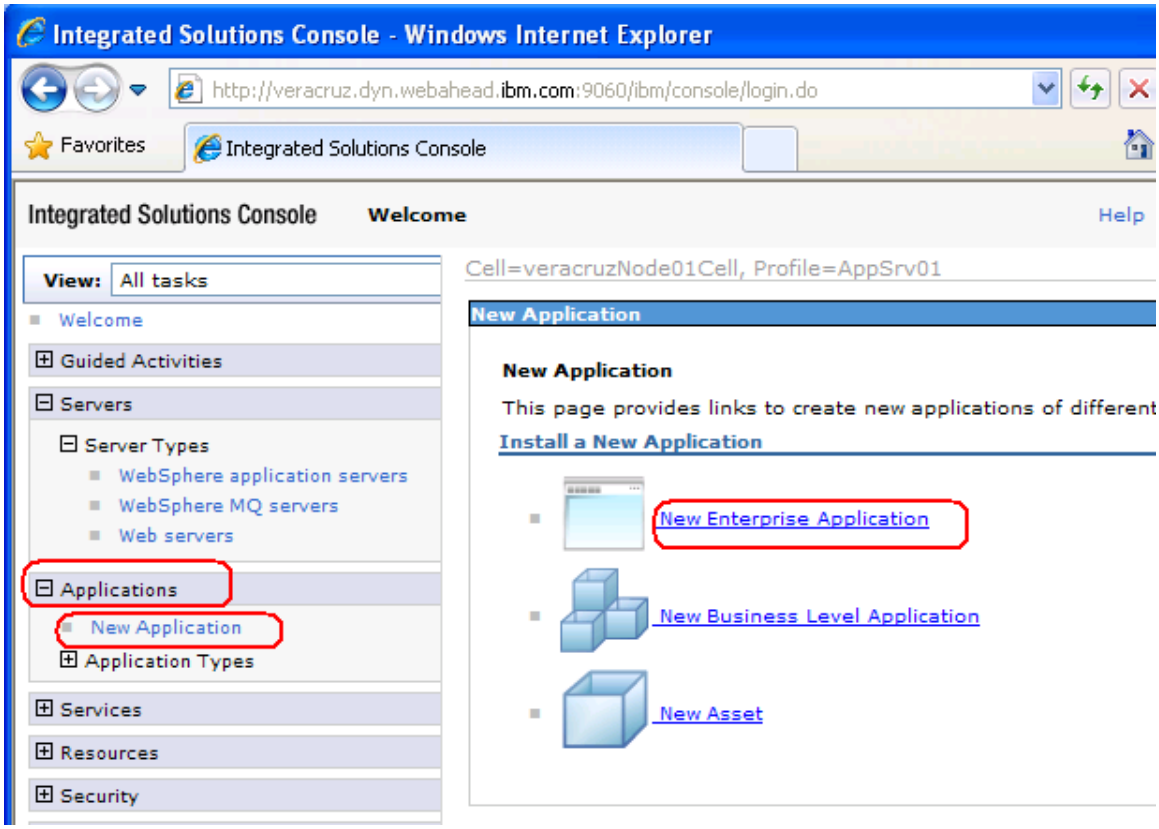
```
+++ SAMPLE MDB: Text Message => TESTING MDB
```

From the WebSphere Application Server Administrative Console, in the left panel, select:

Applications > New Application

Then from the center window “New Application” select:

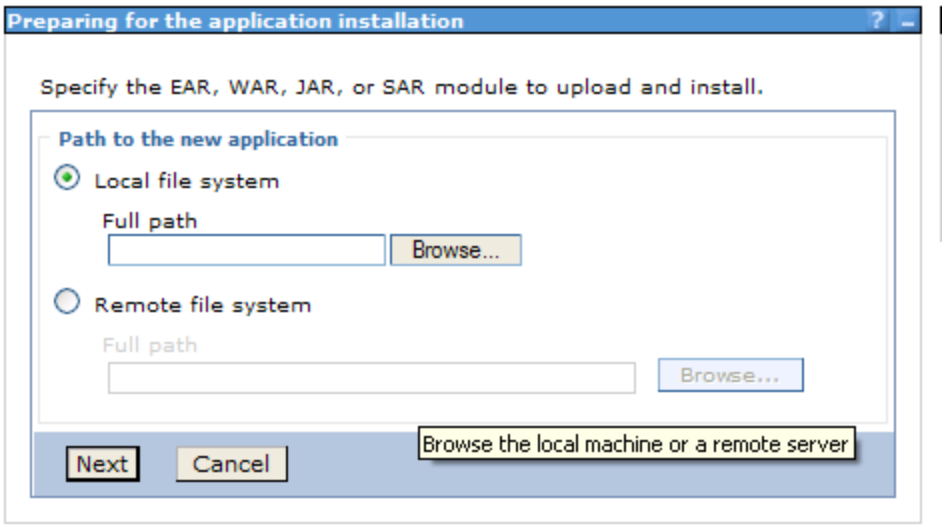
New Enterprise Application



You will see a dialog “Preparing for the application installation”.
Specify the location where you have downloaded the ear file:

SampleMDBEJB.ear

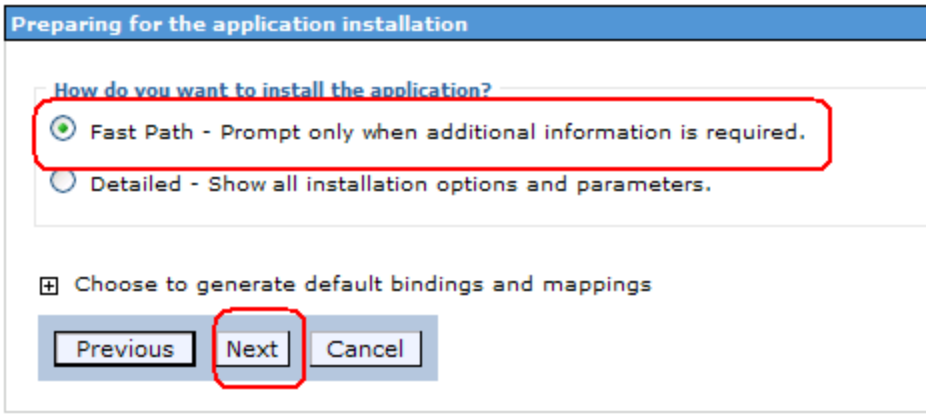
Click Next.



Because we have defined already the Listener Port, the JNDI objects and specified in the EAR Project in RAD which objects we were going to use, then accept the default on how to install the application:

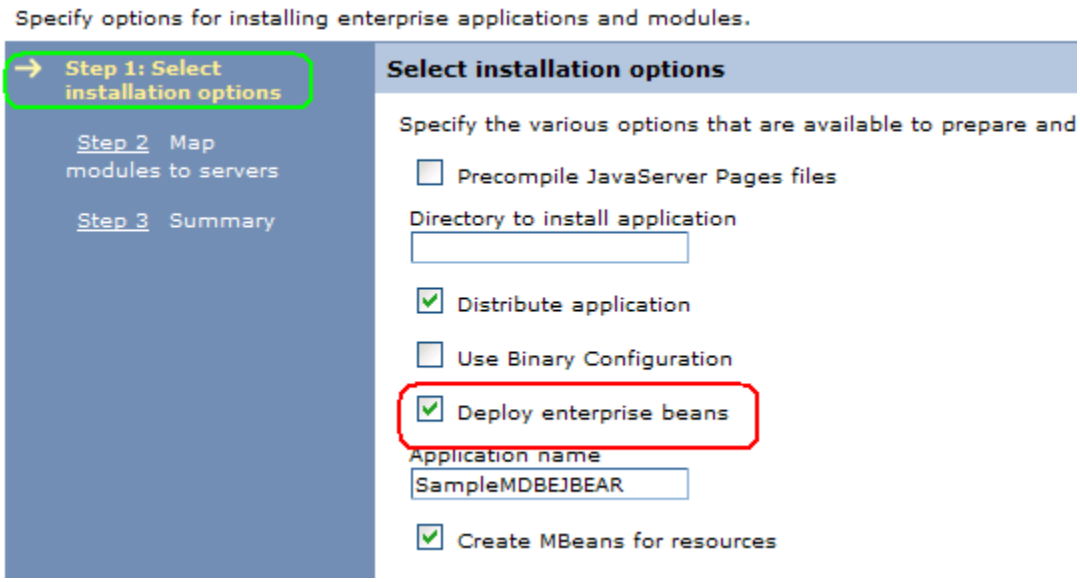
Fast Path – Prompt only when additional information is required

Click Next

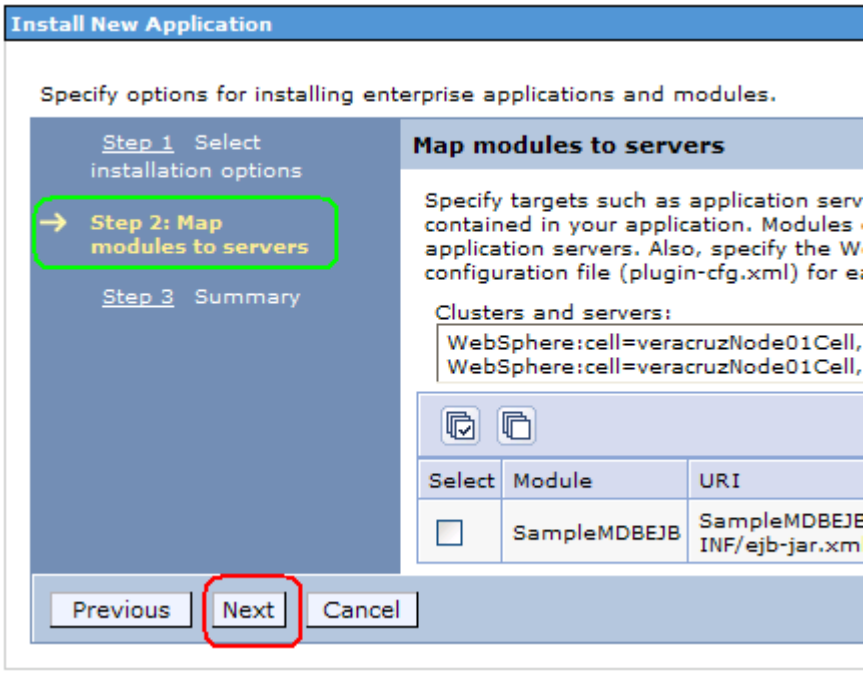


In “Step 1: Select installation options” ensure to select the checkbox:
Deploy enterprise beans
Click Next

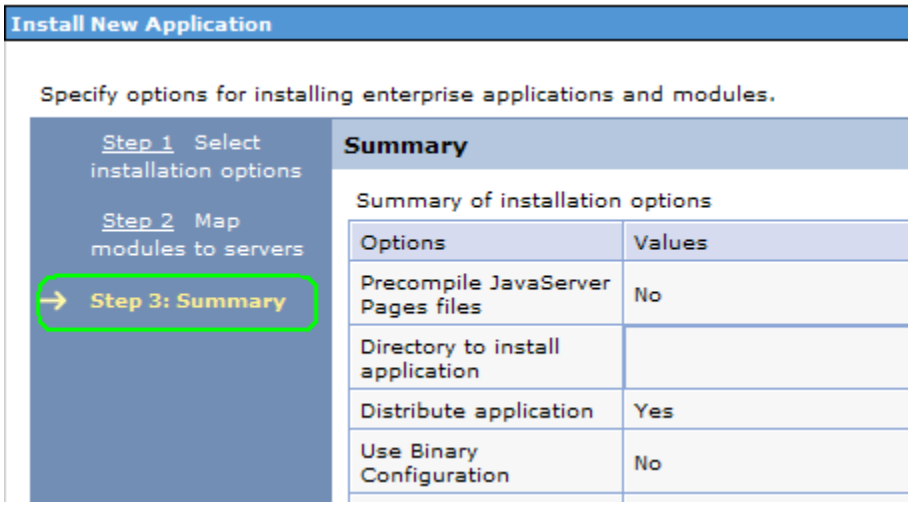
Note: For more information on this checkbox, consult:
http://www.ibm.com/support/knowledgecenter/SSEQTP_7.0.0/com.ibm.websphere.base.doc/info/aes/ae/urun_rapp_installoptions.html
WebSphere Application Server V7
Reference > Settings
> Select installation options settings
> Deploy enterprise beans



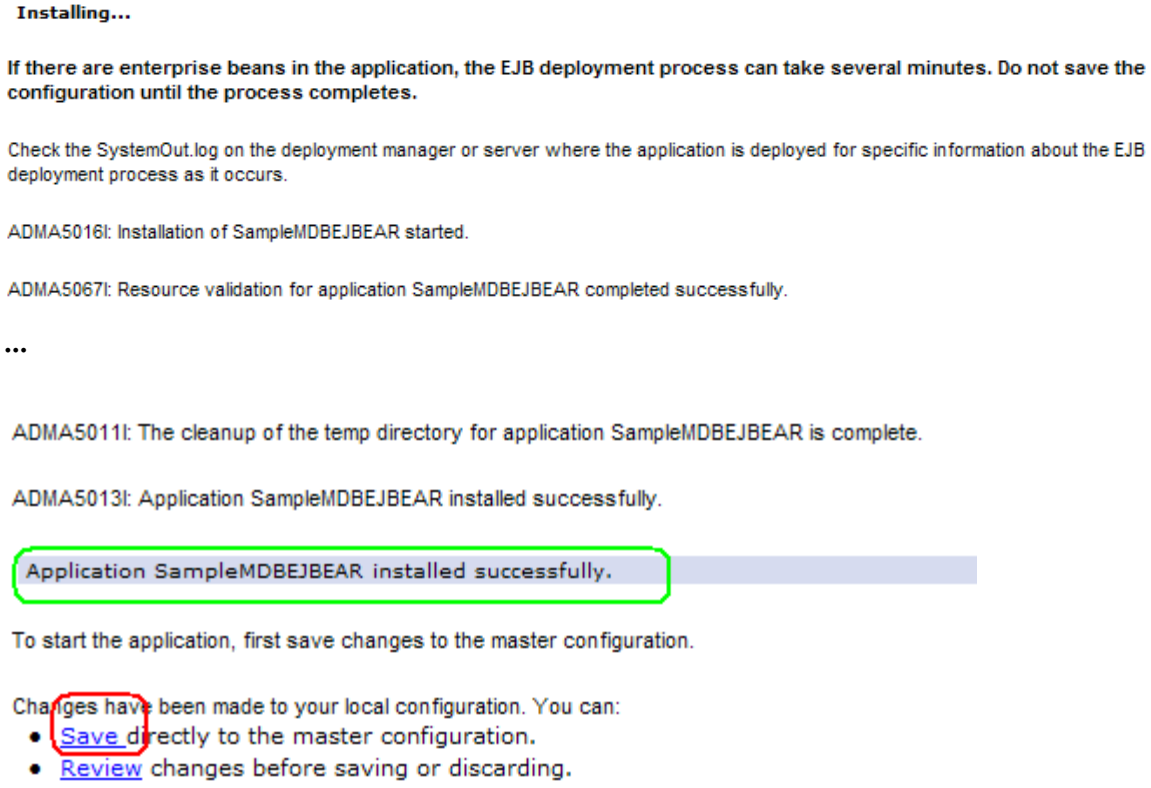
In “Step 2: Map modules to servers” just click Next.



In “Step 3: Summary” just click Finish.



You will see the activity of the deployment:



Ensure that you see the successful messages:

```
ADMA5013I: Application SampleMDBEJBEAR installed successfully
Application SampleMDBEJBEAR installed successfully
```

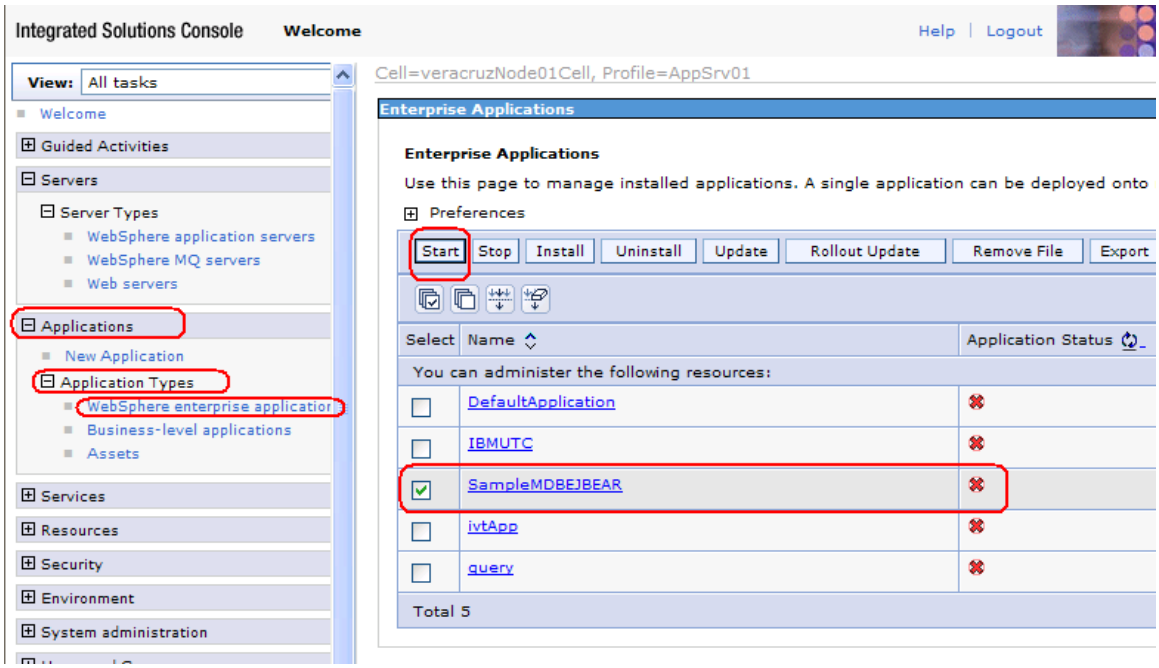
Then click “Save” to save the master configuration.

This deployment process does NOT start the application. You need to start manually the application:

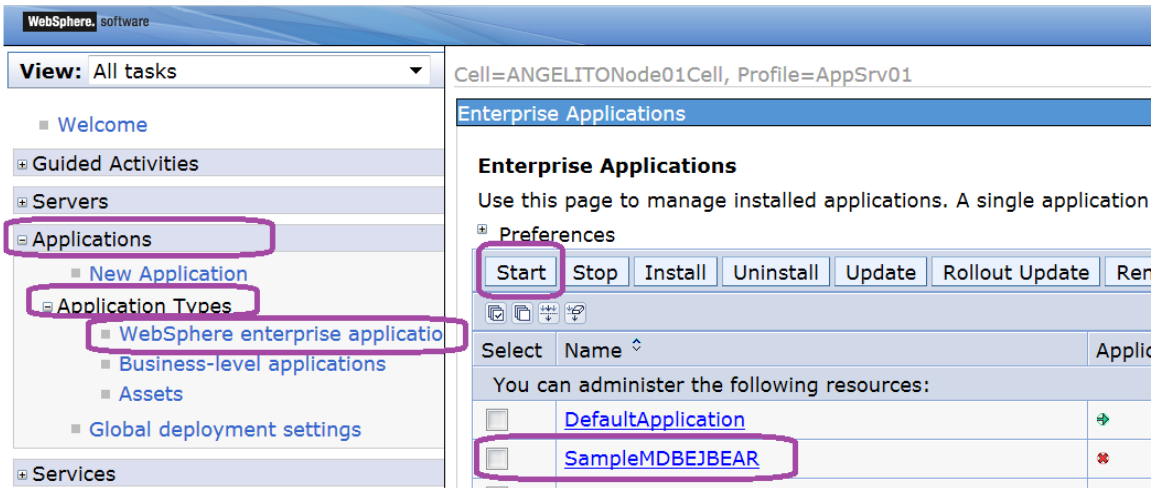
From the console, in the left panel, select:
Applications > Application Types > WebSphere enterprise applications

From the “Enterprise Applications” window, select “SampleMDBEJBEAR” and click on the box to the left of the name. Then click “Start”.

In MQ V7:



In WAS 8.x:



Once the application is started, notice that the icon with a red-cross (Stopped) is changed to a green arrow (Started):

Let's test the MDB. We will need 2 command prompt windows:

+ Window 1: One for watching the recent entries in the WebSphere Application Server SystemOut.log. We can watch for the output from the MDB that indicates that a message was received.

+ Window 2: The other for entering an MQ command that places one message into the queue that is monitored by the WebSphere Application Server Listener Port and which passes the message to the MDB.

Step A: From Window 1:

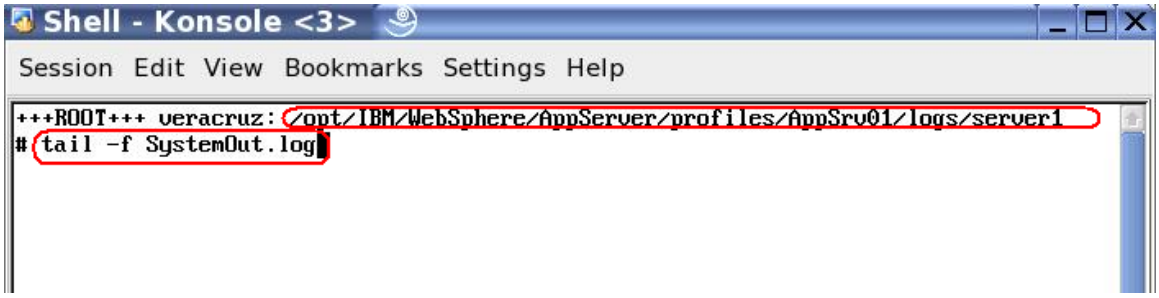
Change to the directory where the WebSphere Application Server server logs reside. In this case it is:

```
$ cd /opt/IBM/WebSphere/AppServer/profiles/AppSrv01/logs/server1
```

Issue the command to watch constantly the recent lines into the SystemOut.log file.

Unix:

```
$ tail -f SystemOut.log
```



Step B: From Window 2:

Login as user “mqm” (or another user who has access to MQ).

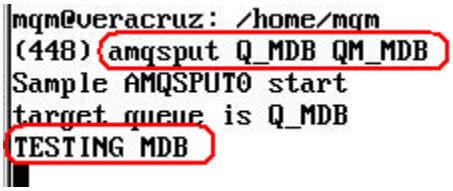
Enter the command to put a message into the queue Q_MDB from the queue manager QM_MDB.
This sample uses a bindings connection type and it needs to be run in the same host that has the queue manager.

```
$ amqsput Q_MDB QM_MDB
```

Enter a text that you could easily identify from the SystemOut.log, such as:

```
TESTING MDB
```

Example snapshot:

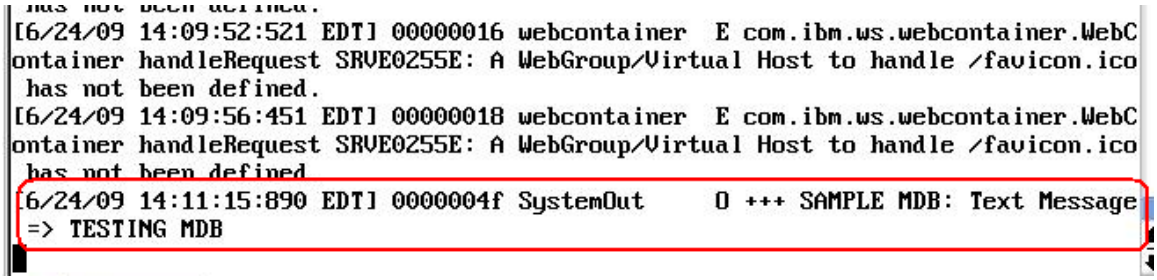


Press Enter to end amqsput.

Step C: From Window 1:

Notice the text at the bottom of the SystemOut.log file:

```
      +++ SAMPLE MDB: Text Message => TESTING MDB
```



Press Ctrl-C to end the “tail” command on the SystemOut.log.

This is the end of the main scenario.

++
+++ Chapter 4: Additional scenarios: Topics, Activation Specification
++

The purpose of this chapter is to explore additional related scenarios:
- How to use MQ Topics to exploit Publish/Subscribe from MQ.
- How to use a WebSphere Application Server Activation Specification, instead of a WebSphere Application Server Listener Port.

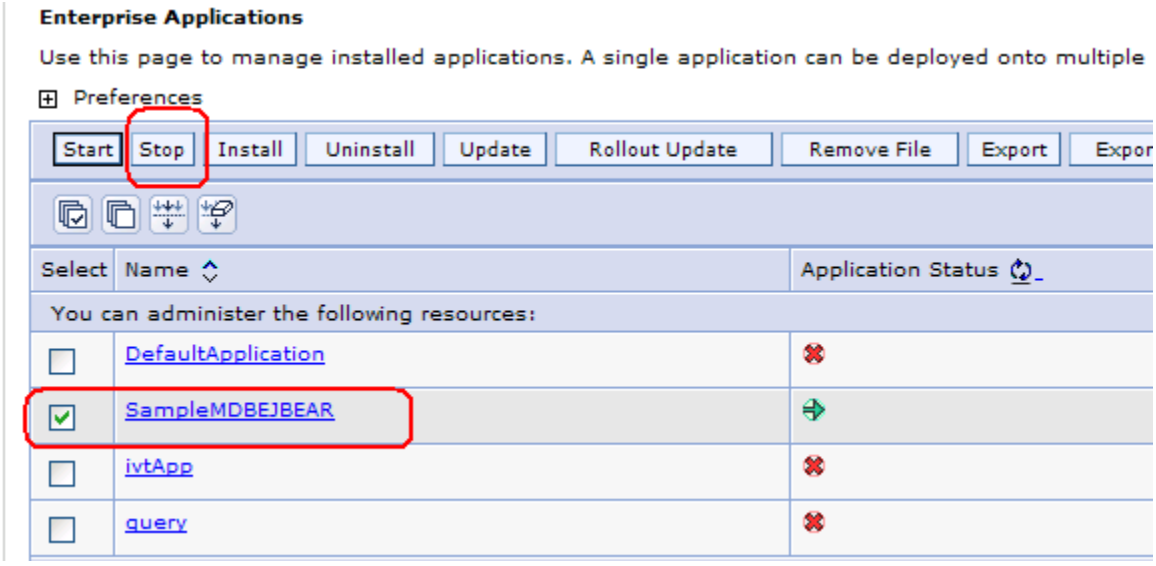
4.1: How to use MQ Topics to exploit Publish/Subscribe from MQ.

Assuming that you are continuing from the preceding chapter, it is necessary to stop the application and the WebSphere Application Server listener port.

Go to the screen:
Servers > Server Types > WebSphere application servers
Select “server1” and follow the instructions mentioned in Chapter 2 to go to:
Application servers > server1 > Message listener service > Listener ports
Stop “SampleMDBQueueLP” and start “SampleMDBTopicLP”.
The new status should look like this:

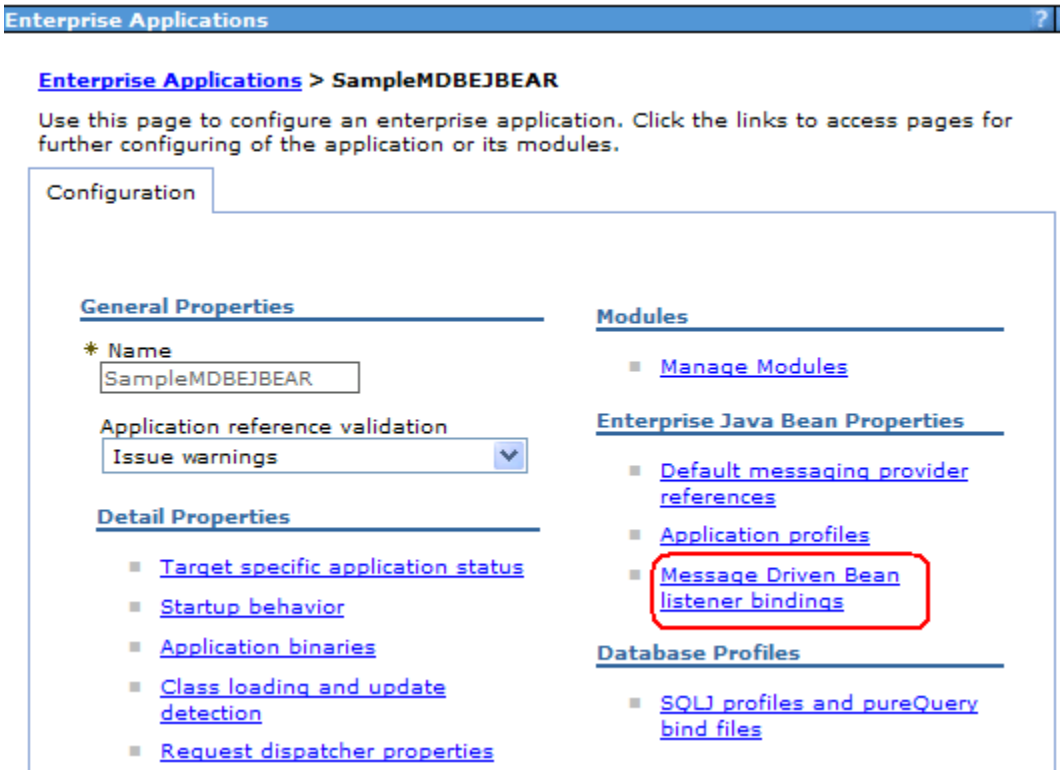
Select	Name	Description	Connection factory JNDI name	Destination JNDI name	Status
You can administer the following resources:					
☐	SampleMDBQueueLP		jms/SampleMDBConnectionFactory	jms/SampleMDBQueue	✖
☐	SampleMDBTopicLP		jms/SampleMDBConnectionFactory	jms/SampleMDBTopic	➔

Go to the screen:
Applications > Application Types > WebSphere enterprise applications
Select the sample application and click on Stop.

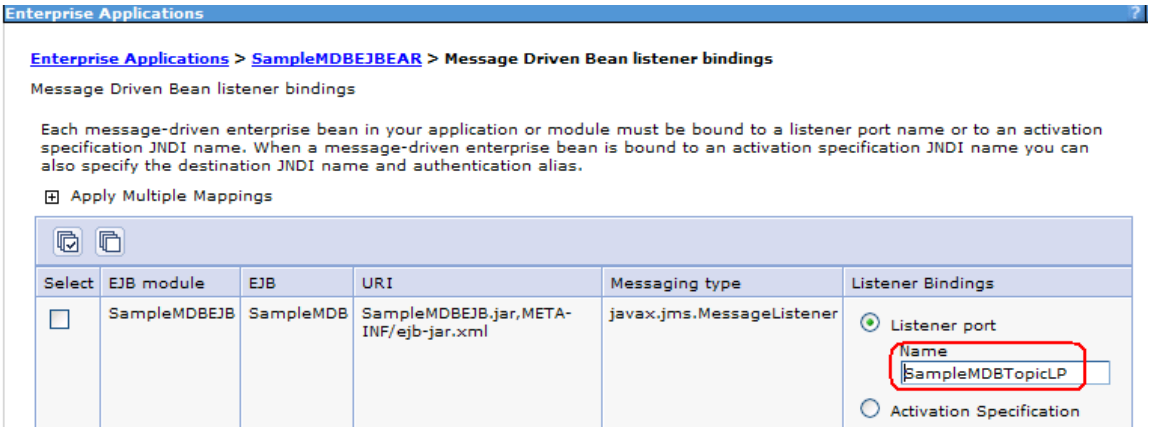


Once it is stopped, click on “SampleMDBEJBEAR”:

Click on “Message Driven Bean listener bindings”:



Under the Listener Bindings column (the right most)
Replace: SampleMDBQueueLP
For: SampleMDBTopicLP



Click on OK then click on Save.

Restart the application.

Perform a similar test of the MDB as described in Chapter 3: open 2 Windows:

a) In one window show the last entries of SystemOut.log by issuing:

```
tail -f SystemOut.log
```

b) In the other window, instead of using "amqsput" (which puts a message into a queue), you can use the following MQ sample that publishes a message into a topic string. In this case the message text is: `TESTING TOPIC`

```
$ amqspub sports QM_MDB
Sample AMQSPUBA start
target topic is sports
TESTING TOPIC

Sample AMQSPUBA end
```

c) Notice that in the window for SystemOut.log, the MDB will write the following:

```
+++ SAMPLE MDB: Text Message => TESTING TOPIC
```

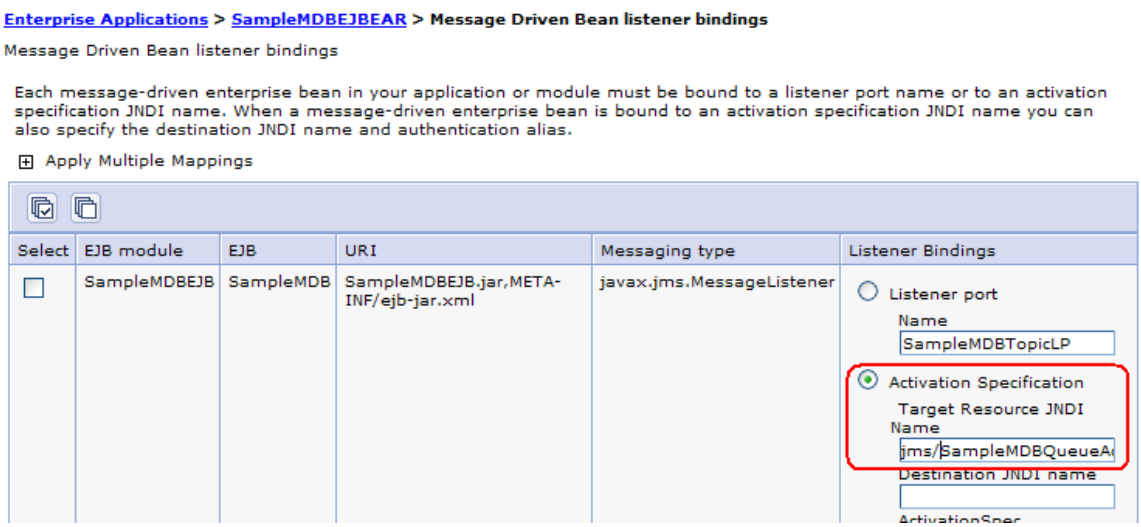
4.2 - How to use a WebSphere Application Server Activation Specification, instead of a WebSphere Application Server Listener Port.

4.2.1 – Using Queues

Stop the application. See previous section for details.
Actually, there is no need to stop the application in order to change the listener bindings.

Change the listener bindings for the MDB:
Go to the screen: Enterprise Applications > SampleMDBEJBEAR > Message Driven Bean listener bindings

Uncheck “Listener Port”.
Check “Activation Specification” and specify the Activation Spec for Queues.
Targer Resource JNDI Name: jms/SampleMDBQueueActivationSpec



Click on OK then click on Save.
Restart the application.

Repeat the testing:

a) Use “amqsput” to put a message in the queue Q_MDB:

```
$ amqsput Q_MDB QM_MDB
Sample AMQSPUT0 start
target queue is Q_MDB
TESTING QUEUE USING ACT SPEC
Sample AMQSPUT0 end
```

b) The bottom of the SystemOut.log will look like this:

```
+++ SAMPLE MDB: Text Message => TESTING QUEUE USING ACT SPEC
```

4.2.2 – Using Topics

Change the listener bindings for the MDB:

Check “Activation Specification” and specify the Activation Spec for Topics.
Targer Resource JNDI Name: jms/SampleMDBTopicActivationSpec

Repeat the testing:

a) Use “amqspub” to publish a message in the topic string “sports”:

```
$ amqspub sports QM_MDB
Sample AMQSPUBA start
target topic is sports
TESTING TOPIC USING ACT SPEC
Sample AMQSPUBA end
```

b) The bottom of the SystemOut.log will look like this:

```
+++ SAMPLE MDB: Text Message => TESTING TOPIC USING ACT SPEC
```

This is the end of the techdoc.