

2nd December 2012

WebSphere MQ Publish/Subscribe and JMS

Message-oriented middleware decouples applications communicating by using two messaging models: point-to-point messaging and publish/subscribe messaging.

Lets see here how publish/subscribe messaging is used for communication between two JMS applications making use of the administrative features of publish/subscribe provided in WebSphere MQ v7.

What is publish/subscribe?

Publish/subscribe messaging allows you to decouple the provider of information, from the consumers of that information. **The sending application and receiving application do not need to know anything about each other for the information to be sent and received.**

Components of publish/subscribe

Publisher - This component generates messages (publications) under a certain subject (topic) and sends the publications to a WMQ queue manager. A publisher knows nothing about the recipient(s) of the information.

Subscriber - This component registers interest in a certain topic(s) with a WMQ queue manager. It also specifies its subscription queue name; that is, the queue where it wants to receive the messages related to this topic. The subscriber knows nothing about the information publishers(s).

Publications Router - This component passes messages from publishers to topic subscribers. Generally this is a Queue Manager.

What is Topic and how is it different from Queue?

A topic is a character string that describes the subject of the information that is published in a publish/subscribe message. A queue is always associated with a specific address and can be used only in one-to-one communication. A topic has no logical address associated with and can be subscribed or published by many and hence can be said used in one-to-many or many-to-one communication. In WebSphere MQ you can create a Topic in same way as a queue and has a default system object called `SYSTEM.DEFAULT.TOPIC`.

From WebSphere MQ v7 Queue Manager is used as the publications router and we can define administrative objects like Topics and Subscriptions through WebSphere MQ Explorer. Prior to this a Broker component was used for this.

Also WebSphere MQ Explorer is used to create JMS Administered objects required by the applications to communicate with each other. WebSphere MQ Explorer is used to create and configure connection factories and destinations as administered objects in a JNDI namespace. An application can then retrieve the administered objects from the JNDI namespace. Connection factories and destinations are starting points in the flow of logic of a JMS application. An application uses a ConnectionFactory object to create a connection to a messaging server, and uses a Queue or Topic object as a target (destination) to send messages to or a source (destination) from which to receive messages.

IBM WebSphere MQ classes for Java Message Service (also referred to as WebSphere MQ JMS) is a set of Java classes that enables the JMS applications to access WebSphere MQ systems. The JMS application is written to use only references to the interfaces in javax.jms package. All vendor-specific information (IBM WebSphere MQ) is encapsulated in implementations of the following JMS administered objects:

ConnectionFactory

QueueConnectionFactory

TopicConnectionFactory

Queue

Topic

The Java classes of the IBM WebSphere MQ JMS provider communicate with the queue manager through Connection factories using one of the following transport options:

1. As an IBM WebSphere MQ client using TCP/IP
2. In bindings mode, connecting directly to IBM WebSphere MQ

A publish/subscribe scenario

A news broadcasting company A sends various channels of news to its clients C and D through email and sms. For e.g take sports news. In point-to-point messaging this is achieved by company A put messages in email and sms queues defined for companies C and D. The destination needs to be known by company A to do this. But in publish/subscribe messaging company A publishes the sports news messages to a component called Topic. The interested clients C and D can subscribe to this topic and provide their subscription queue names where they like to receive the messages. This allows full decoupling of applications involved in messaging. Company A does not know about their Sports News channel subscribers C and D and vice versa. Now if the company A like to create a new news channel for entertainment all it has to do is define a new topic in the intermediate queue manager used for messaging.

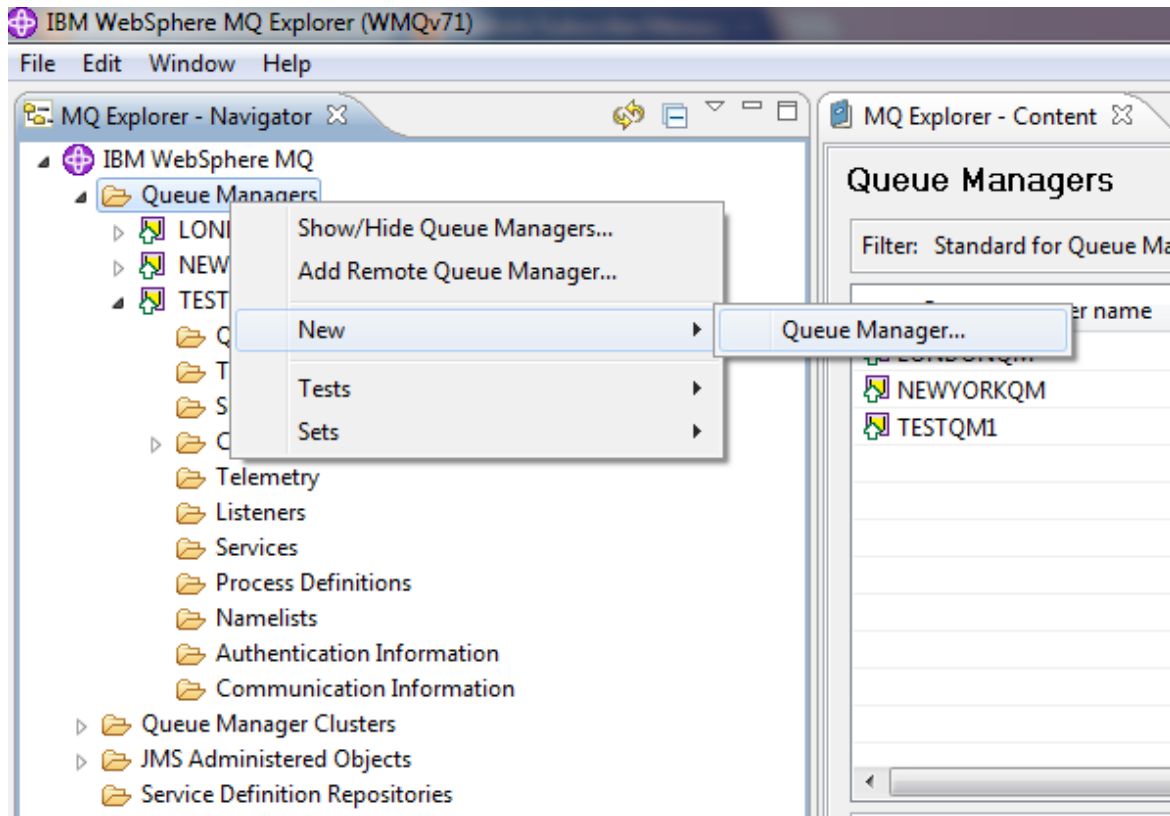
How to setup publish/subscribe using WebSphere MQ JMS and WebSphere MQ Explorer v7

1. Create a queue manager.
2. Create a news topic.
3. Create subscription queues.
4. Create subscriptions.
5. Create Java Messaging Services (JMS)-administered objects.
6. Create a Connection Factory
7. Create Destinations
8. Write publisher and subscriber classes. [Applications]

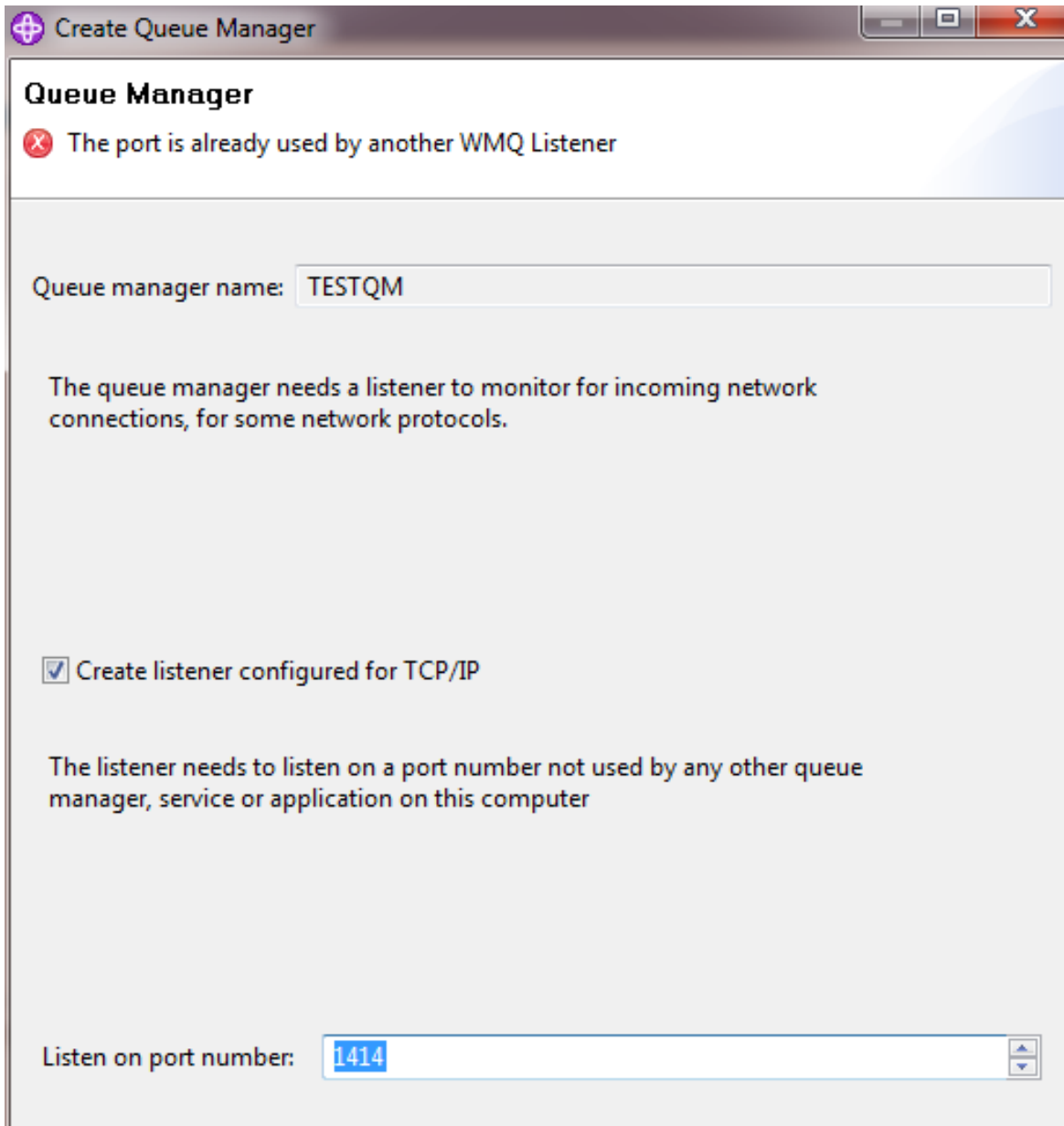
9. Run the classes.

1. Create a Queue manager

In MQ Explorer's Navigator view, right-click Queue Managers and then select New > Queue Manager. This starts the "Create Queue Manager" wizard. On Step 1, enter TestQM1 as the queue manager name and then click Next until you reach the screen to enter port number. Make sure the port number you enter is not used by other Queue Manager. Finally click Finish.



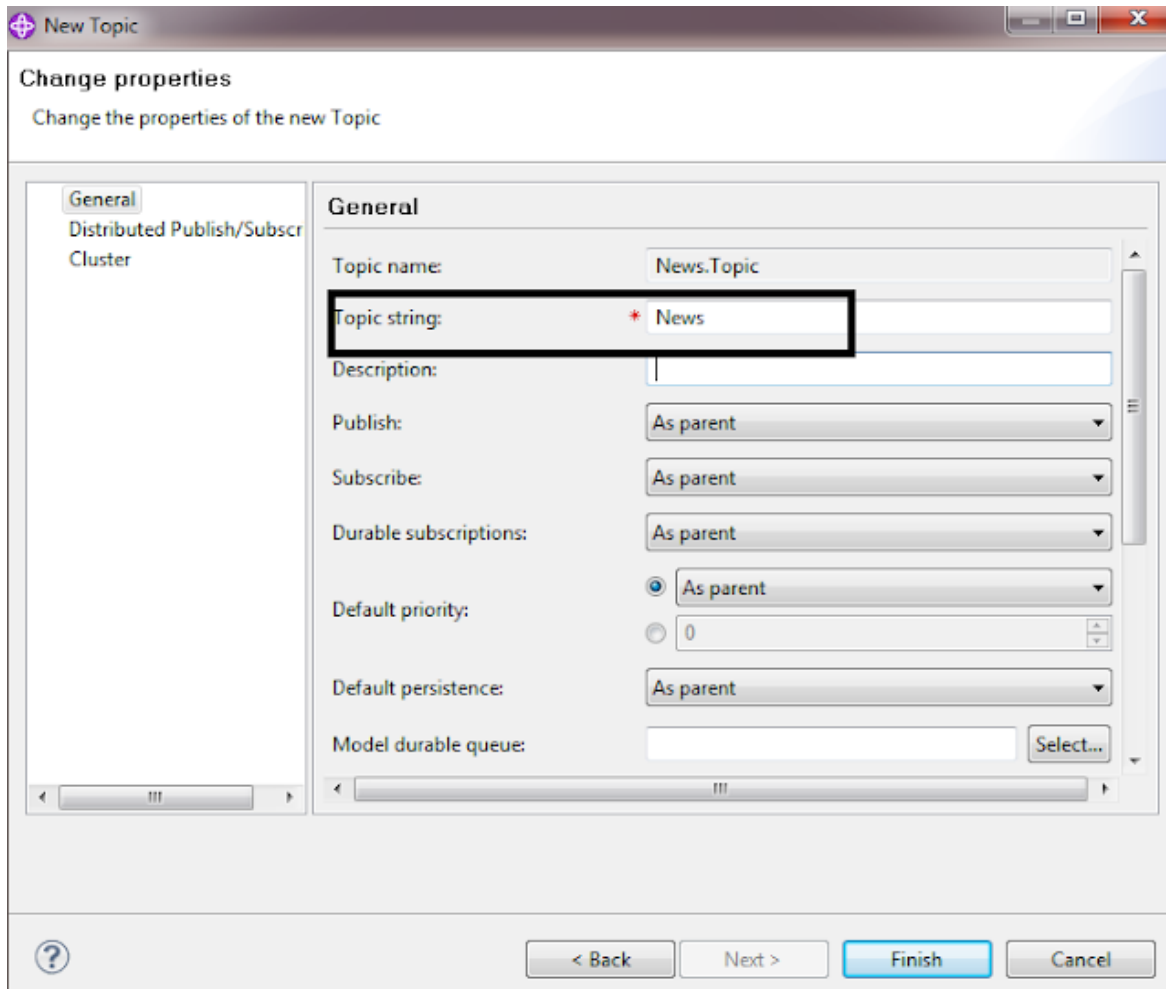
[http://1.bp.blogspot.com/-1WHTTY18I20/UAUy_xY8JTI/AAAAAAAAAC10/RYR4dK3eo7o/s1600/pbsb01.png]



[http://2.bp.blogspot.com/-S9gqZ4hxguA/UAUzWEHUaJI/AAAAAAAC18/eKm_pbR5Ry8/s1600/pbsb02.png]

2. Create a news topic.

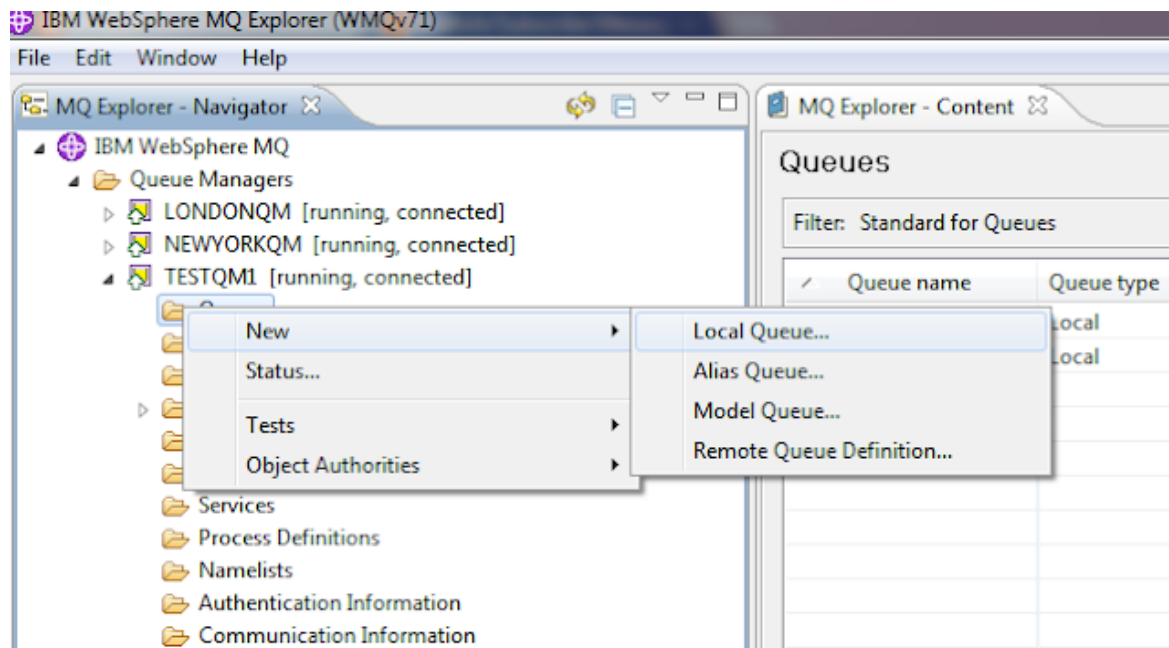
Under TestQM1, right-click Topics and then select New > Topic to start the "New Topic" wizard. Type News.Topic as the topic name and click Next. On the topic properties page, type News in the "Topic string" field and click Finish.



[<http://1.bp.blogspot.com/-2XOz0lYalTY/UAU1pDm73jI/AAAAAAAAAC2E/dM3IFucEi50/s1600/pbsb1.png>]

3. Create subscription queues.

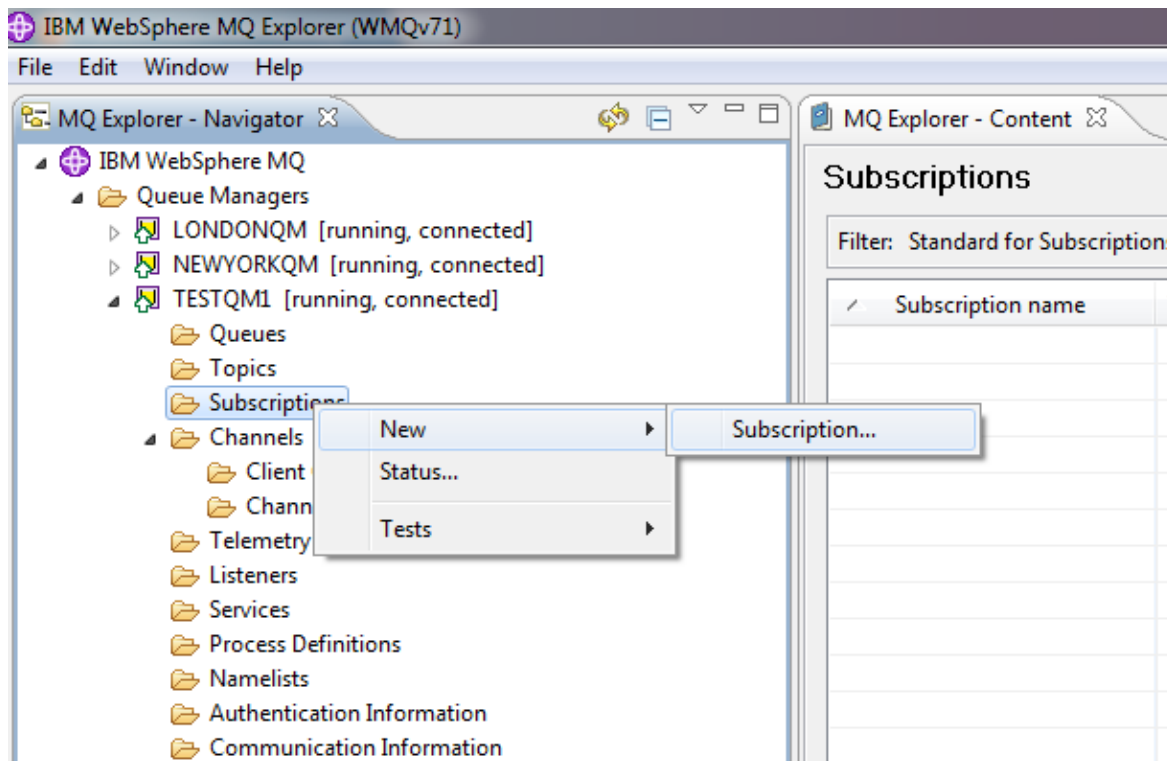
Under TestQM1, right-click Queues then select New > Local Queue to start the "New Local Queue" wizard. In the name field, type Email.Queue and click Finish. Repeat the same for SMS.Queue.



[<http://3.bp.blogspot.com/-Q-aP-f4KUpw/UAU3trhx-ul/AAAAAAAAAC2M/WYCvhtnDL6g/s1600/pbsb03.png>]

4. Create subscriptions.

Under TestQM1, right-click Subscriptions and then select New > Subscription to start the "New Subscription" wizard. In the name field, type EmailSubscription and click Next. In the "Change properties" page, press the Select button next to the "Topic name" and then select News.Topic from the displayed topics list. Press OK. On the same page, type Email.Queue in the destination name field and click Finish.



[<http://2.bp.blogspot.com/-ZclAcbEpHEc/UAU9e56Qq-I/AAAAAAAAAC2Y/Ap8yJXIfJSI/s1600/pbsb2.png>]

New Subscription

Change properties

Change the properties of the new Subscription

General

Subscription name: EmailSubscription

Topic

Topic name: News.Topic Select...

Topic string:

Wildcard usage: Topic level wildcard

Scope: All

Destination

Destination class: Provided

Destination queue manager:

Destination name: * Email.Queue

Correlation identifier:

00000 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

00010 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

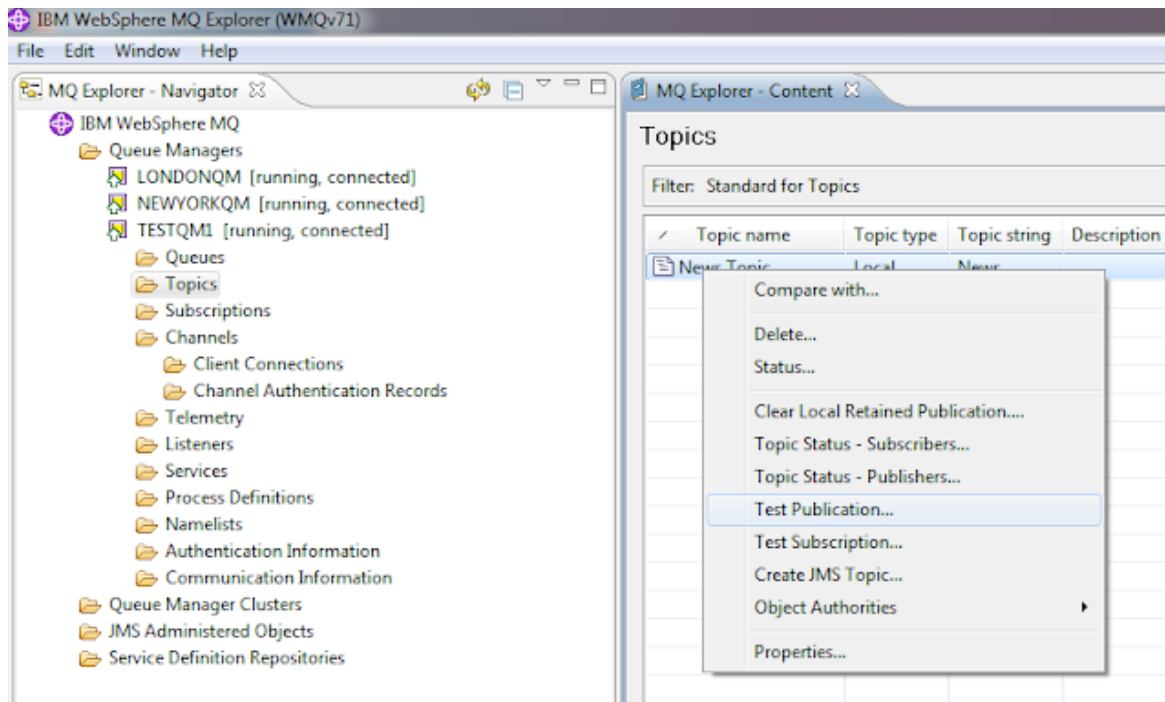
< Back Next > Finish Cancel

[\[http://1.bp.blogspot.com/-KZET4ws-KSk/UAVAAI98yII/AAAAAAAAAC2k/aOBv-2HAZuY/s1600/pbsb3.png\]](http://1.bp.blogspot.com/-KZET4ws-KSk/UAVAAI98yII/AAAAAAAAAC2k/aOBv-2HAZuY/s1600/pbsb3.png)

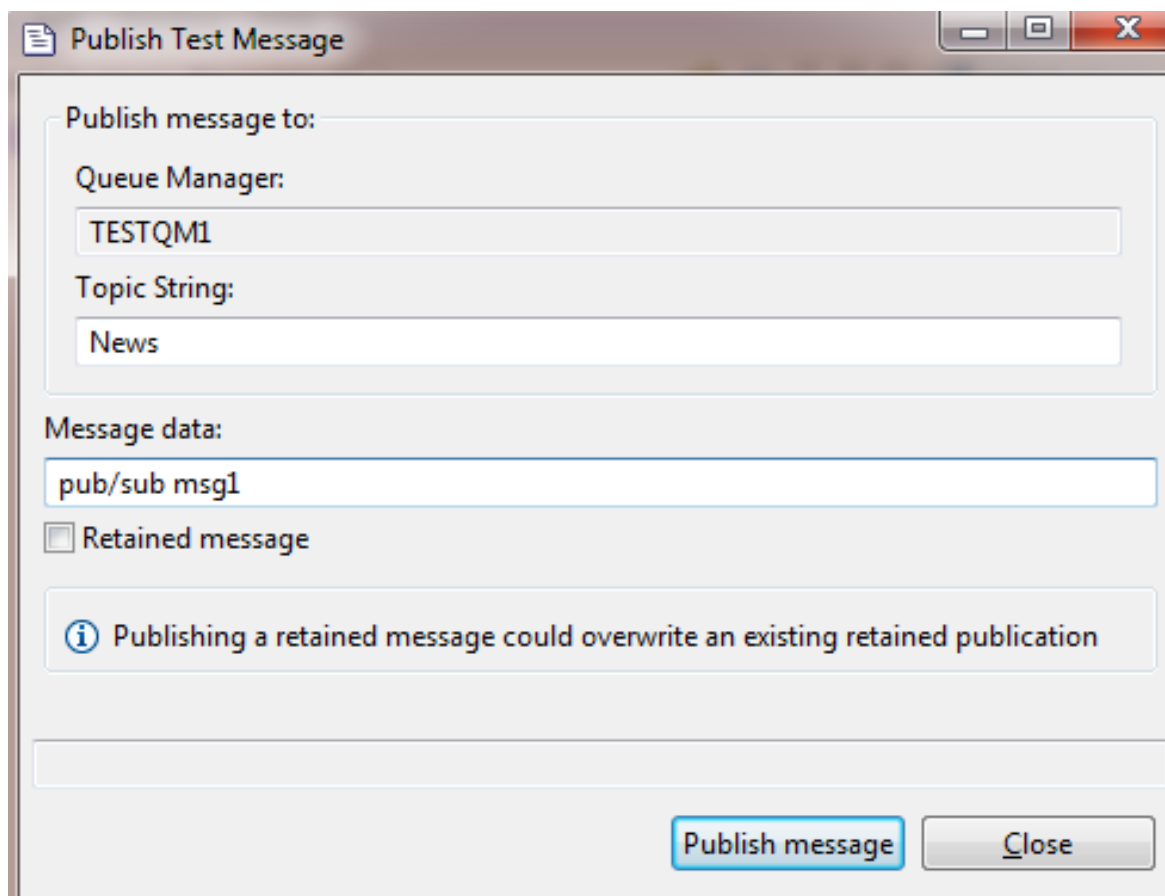
Repeat the above steps to create another subscription called SMSSubscription with the topic set to News.Topic and the destination name set to SMS.Queue.

Quick Test for the Publish/Subscribe setup Created.

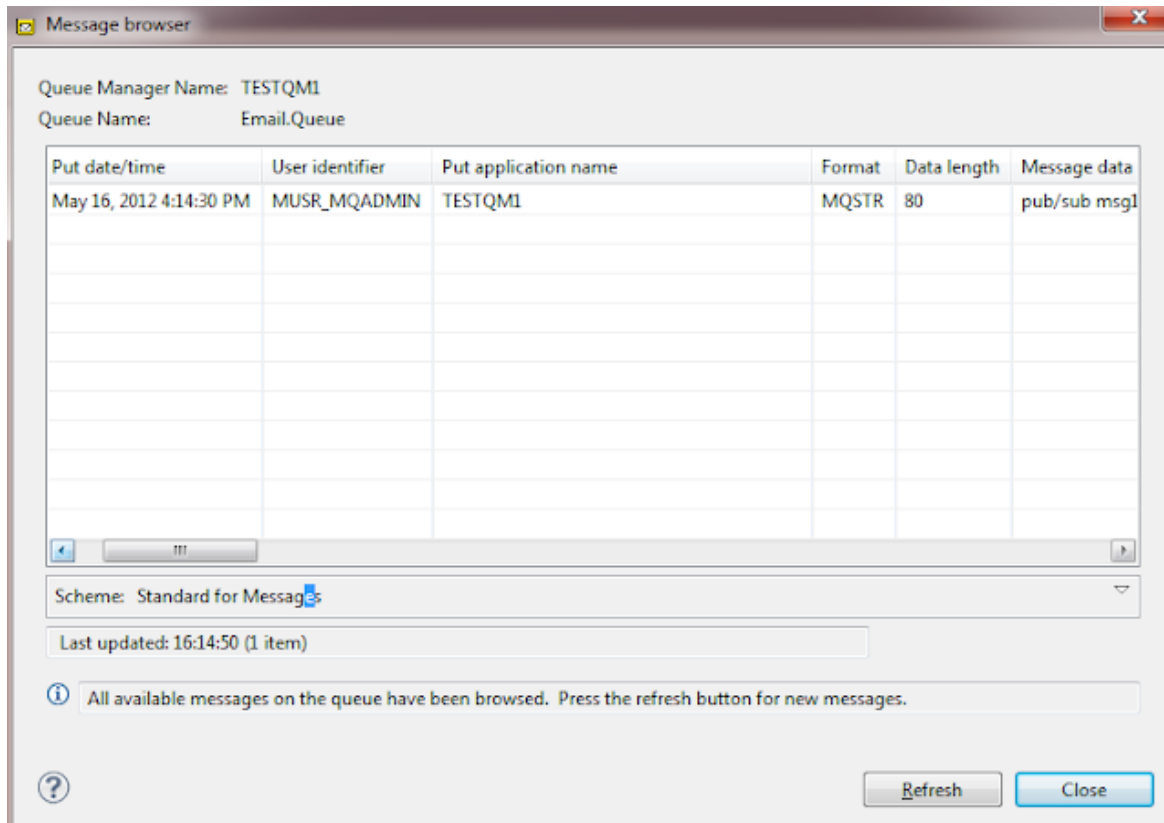
Under TestQM1, click Topics. In the right pane, right click News.Topic and select "Test Publication" to open the test window. In the message data field, type "pub/sub msg1" (or any other text), click the "Publish message" button, and then click the Close button. This should distribute the message to the subscriber queues Email.Queue and SMS.Queue. You can right click on Email.Queue and click Browse Messages... You will find the same message in the Message Data column. Right Click the Queues and Clear Messages... after testing.



[<http://3.bp.blogspot.com/-vFLPoG1SlyQ/UAWBPZ4wt2I/AAAAAAAAAC2w/l63U6qiplm8/s1600/pbsb4.png>]



[<http://4.bp.blogspot.com/-2VWRmMA0vQQ/UAWBTgDpLOI/AAAAAAAAAC24/5ktOrHwjYE/s1600/pbsb5.png>]

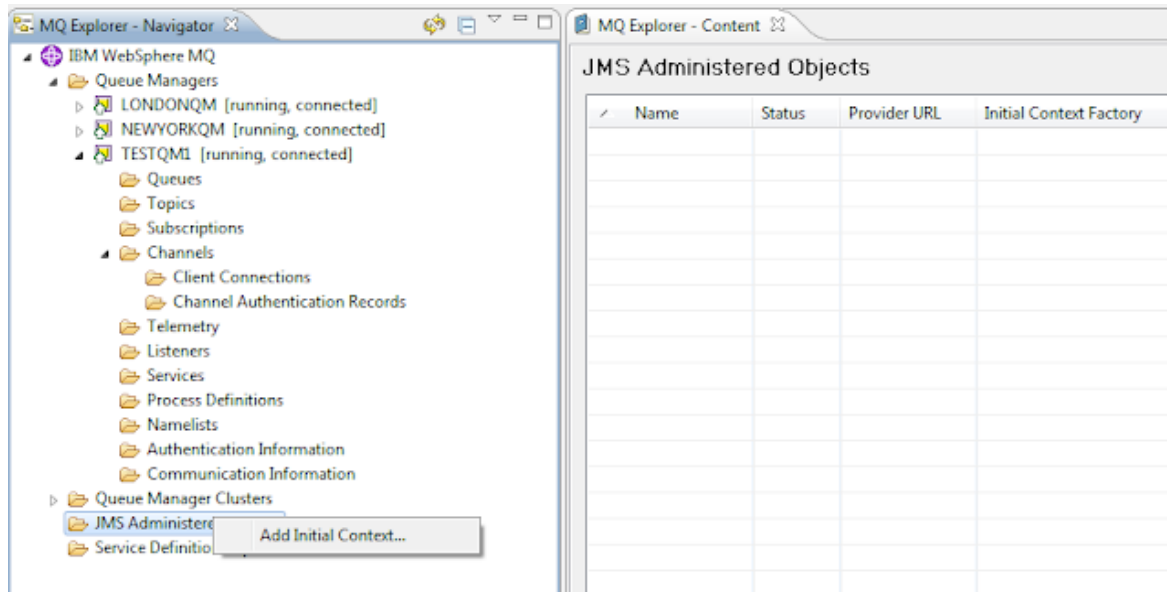


[<http://1.bp.blogspot.com/-uFfKWg-DCo/UAWBYFg8I7I/AAAAAAAAAC3A/wAzZvnS5D1s/s1600/pbsb6.png>]

5. Create Java Messaging Services (JMS)-administered objects.

To store JMS-administered objects, create a file-based Java Naming and Directory Interface (JNDI) namespace. Start by adding a new initial context. Right-click JMS-Administered Objects and then select Add Initial Context. Select "File System" for "Where is the JNDI located." For the Bindings directory, browse to C:\JNDI-Directory (This directory has to exist). Click Finish.

Note::Make a note of Factory class "com.sun.jndi.fscontext.RefFSContextFactory" and Provider URL "file:/C:/JNDI-Directory/". This will be used in the Java code later.



[<http://1.bp.blogspot.com/-hPjZTgwSmiU/UAWeyiUTFel/AAAAAAAAAC3M/8yU-RIwsd3M/s1600/pbsb7.png>]

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

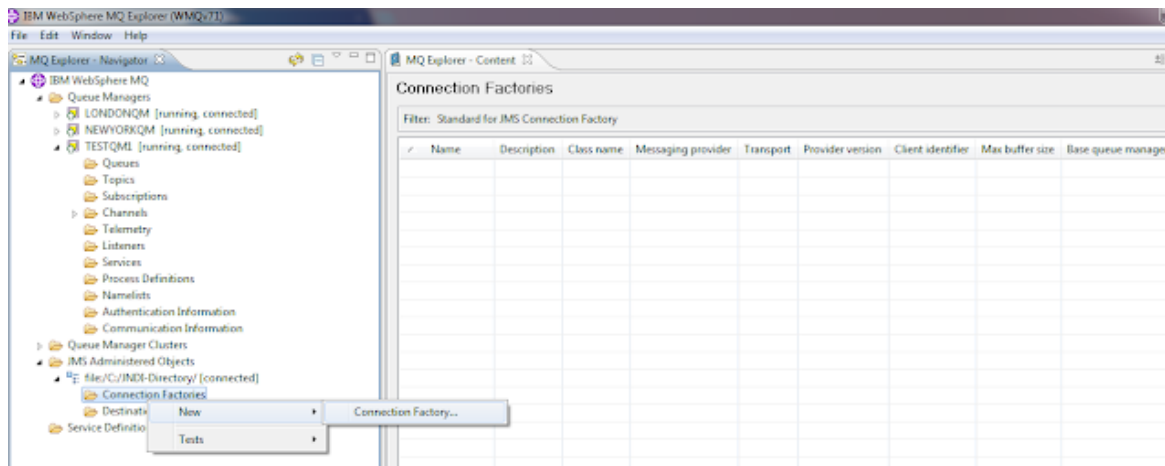
Bindings directory:

Provider URL:

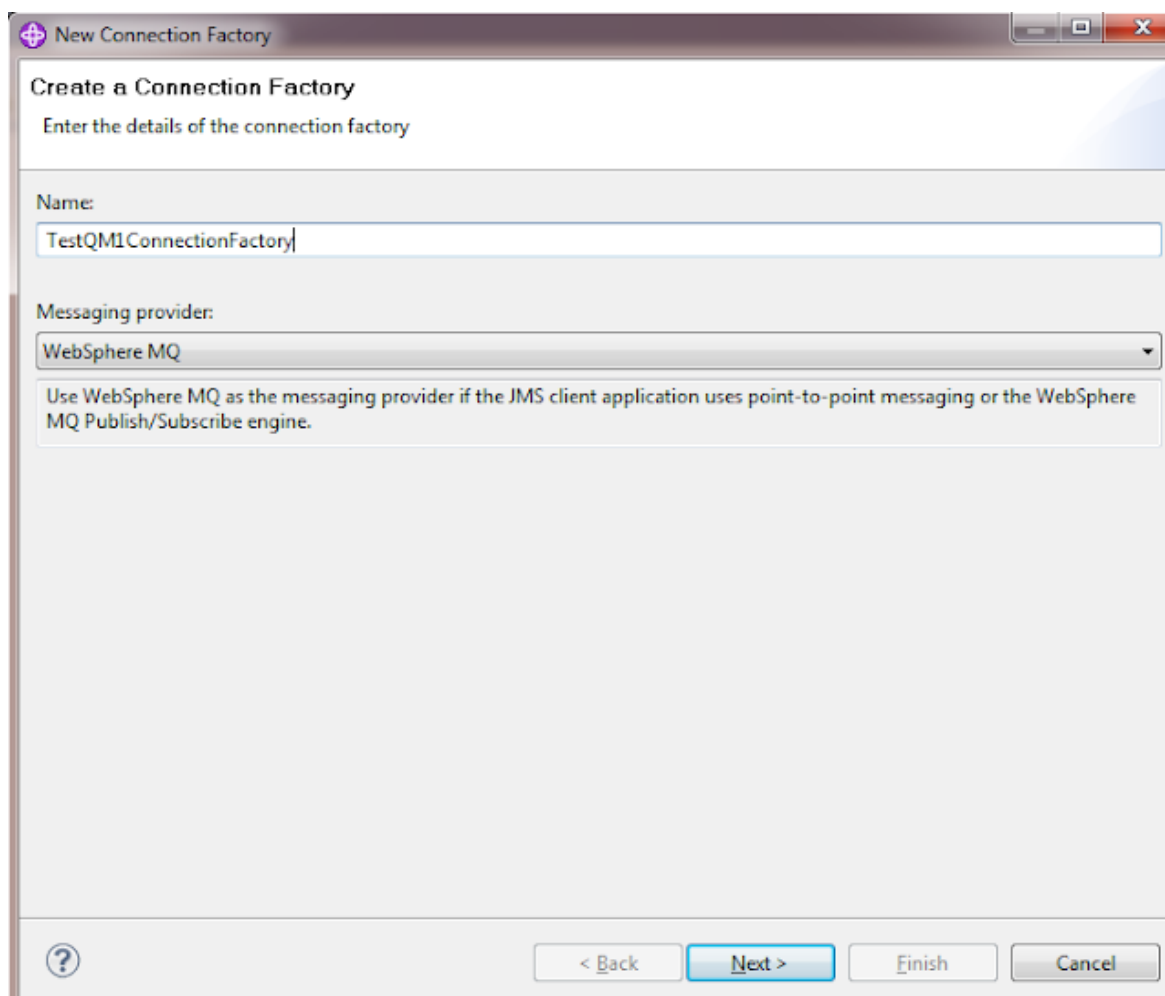
[<http://1.bp.blogspot.com/-r4ifAJwUmvE/UAWExvIMI/AAAAAAAAAC3U/pLab01MiY3w/s1600/pbsb8.png>]

6. Create a Connection Factory

Right-click Connection Factories and then select New → Connection Factory. On the first screen, type TestQM1ConnectionFactory in the name field and click Next. Click Next till last page. On the last page, select the connection tab (on the left) and then click the Select button to select TestQM1 as "Base queue manager". Click Finish.



[<http://4.bp.blogspot.com/-g6WeKCC9jQo/UAZWwBoWRHI/AAAAAAAAAC34/uebrPePn89Y/s1600/pbsb9.png>]



[<http://1.bp.blogspot.com/-yHS2e1Vjoeo/UAZXVslyO8I/AAAAAAAAAC4A/fiLVWFnmV1M/s1600/pbsb10.png>]

New Connection Factory

Change properties

Change the properties of the new Connection Factory

General

Connection

Broker

Temporary queues

Temporary topics

Subscriber

Extended

Advanced tuning

General

Name:

Description:

Class name:

Messaging provider:

Transport:

Provider version: ☒ 7 ☐

Client identifier:

?

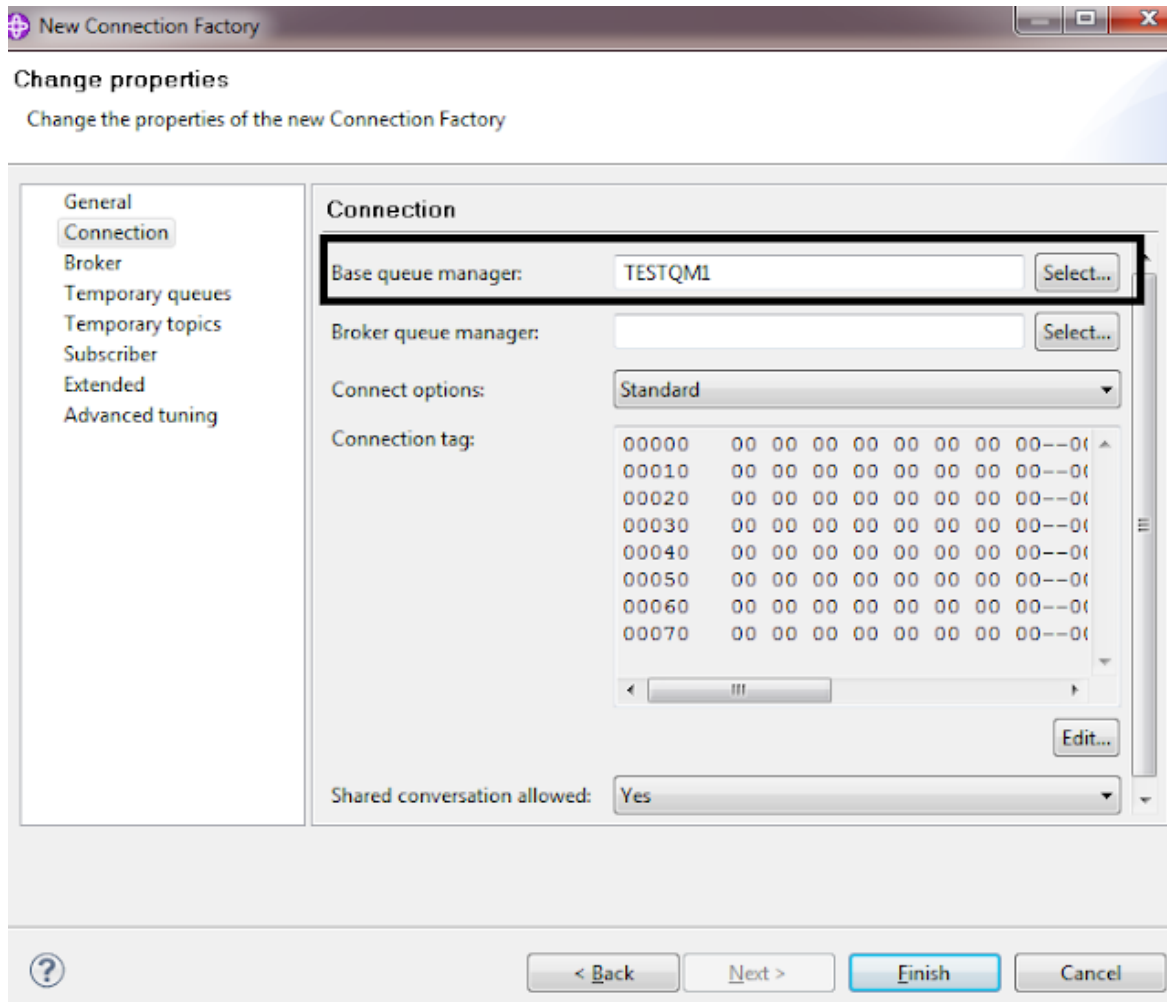
< Back

Next >

Finish

Cancel

[<http://4.bp.blogspot.com/-LTMpAzsf66Y/UAZXhaK2SvI/AAAAAAAAAC4I/xbLB8NzotKk/s1600/pbsb11.png>]



[<http://1.bp.blogspot.com/-8h4dr6U86zU/UAZXriEfx2I/AAAAAAAAAC4Q/UtOmzTPnBTA/s1600/pbsb12.png>]

7. Create Destinations

Right-click destinations and select New → Destination to start the "New Destination" wizard. In the first step, enter NewsTopic in the Name field and change the Type field to Topic. Click Next till last page. On the last page (Change properties), type News in the Topic field and then click Finish.

Right-click destinations and then select New → Destination to start the "new Destination" wizard. In the first step, type EmailQueue in the Name field and then click Next. Click Next on the second screen. On the last page, click the Select button next to the "Queue Manager" field and select TestQM1 from the list. Click the Select button next to the Queue field and select Email Queue from list and click Finish.

Repeat the above steps to create another Destination called SMSQueue for SMS.Queue.

New Destination

Create a Destination

Enter the details of the object you wish to create

Name:
NewsTopic

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:
Topic

Select this option if the JMS application uses publish/subscribe messaging. The destination will represent a topic.

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

☐ Start wizard to create a matching MQ Topic

? < Back Next > Finish Cancel

[http://4.bp.blogspot.com/-MDc7fQMMPN4/UAZge_QpBqI/AAAAAAAAAC4c/UuQg_LGR1Rc/s1600/pbsb13.png]

New Destination

Change properties

Change the properties of the new Destination

- General
- Message handling
- Broker
- Producers
- Consumers
- Extended

General

Name:

Description:

Class name:

Messaging provider:

Topic: *

?

< Back Next > **Finish** Cancel

[http://1.bp.blogspot.com/-7-pwfCL2qdM/UAZgj1UjqvI/AAAAAAAAAC4k/zFND_X-yDII/s1600/pbsb14.png]

New Destination

Create a Destination

Enter the details of the object you wish to create

Name:
EmailQueue

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

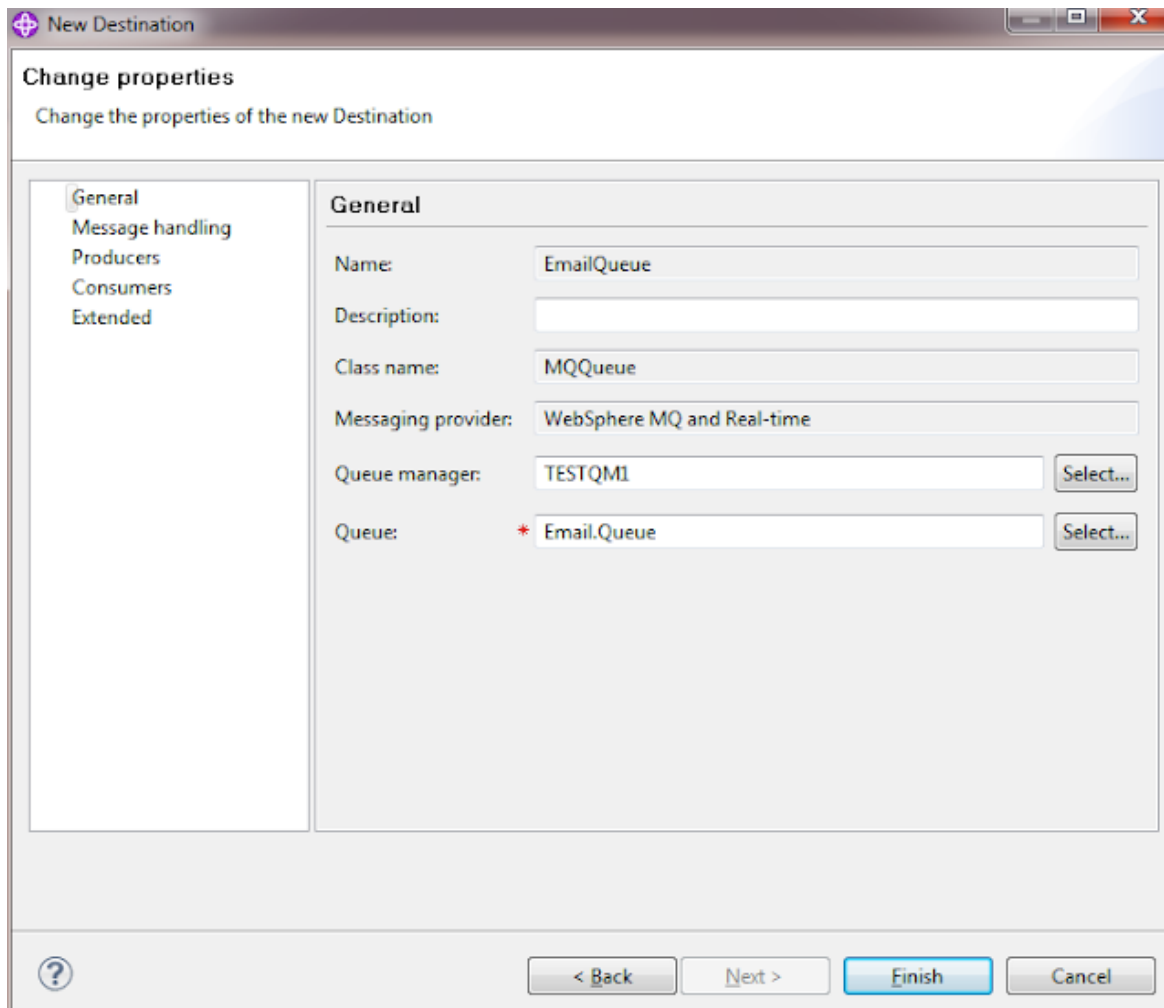
Select this option if the JMS application uses point-to-point messaging. The destination will represent a queue.

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

[<http://3.bp.blogspot.com/-MHwo-owEE10/UAZgoLAvVfI/AAAAAAAAAC4s/G8abH3AHvrw/s1600/pbsb15.png>]



[http://4.bp.blogspot.com/-gU1o7T_EpC0/UAZhP9nMIyI/AAAAAAAAAC40/QRYYLcQe8Sw/s1600/pbsb16.png]

The screenshot shows the 'MQ Explorer - Content' window with the 'Destinations' tab selected. The table below lists the destinations:

Name	Description	Class name	Messaging provider	Queue manager	Queue	Topic	Expiry	Persistence	Priority
EmailQueue		MQQueue	WebSphere MQ and Real-time	TESTQM1	Email.Queue		Application	Application	Application
NewsTopic		MQTopic	WebSphere MQ and Real-time			News	Application	Application	Application
SMSQueue		MQQueue	WebSphere MQ and Real-time	TESTQM1	SMS.Queue		Application	Application	Application

[<http://4.bp.blogspot.com/-KPKofVZod8/UAZh6zbrJI/AAAAAAAAAC48/UY8kvYGB0kl/s1600/pbsb17.png>]

8. Write publisher and subscriber classes.

Now all the configurations involving JMS Administered objects and MQ Explorer are complete we can write the JMS Applications.

The Applications will be simple.

Publisher Application write a message to a Topic.
Subscriber Application read messages from Subscription Queues.

Lets see what are the java classes [Applications] about. The Applications use JMS Interfaces [API] which treat queues and topics uniformly as destinations. You have a utility class and a Publisher class and two Subscriber classes.

JNDIUtil Class

This class includes generic methods which retrieve JMS objects through JNDI lookup. Get the InitialContext and assign it to a Context object. Generic methods in this class are used to retrieve references to the JMS objects already defined. `getObjectByName()` `getConnectionFactory()` `getDestination()`

NewsPublisher Class

This class publishes messages to News Topic
Get a ConnectionFactory through lookup and use the factory object to create a Connection

```
factory= jndiUtil.getConnectionFactory("TestQMConnectionFactory");  
connection = factory.createConnection();
```

Use the connection object to create a session

```
session = connection.createSession( transacted, Session.AUTO_ACKNOWLEDGE);
```

To publish messages retrieve a News destination object and create a MessageProducer and then send messages:

```
destination = jndiUtil.getDestination("NewsTopic");  
producer = session.createProducer(destination);  
TextMessage message = session.createTextMessage("No News is Good News!");  
producer.send(message);
```

EmailSubscriber and SMSSubscriber

These applications read messages from the Subscription queues. It uses a MessageConsumer to retrieve messages from queues:

```
consumer = session.createConsumer(destination);TextMessage iMsg =  
(TextMessage) consumer.receive(1000);
```

9. Run the classes

Now since the setup is complete you can run the Java Classes.

The JAR files required to compile and run the sample classes are automatically added to the CLASSPATH environment variable when you install WMQ. The required JARs are located in C:\Program Files\IBM\WebSphere MQ\Java\lib, including the JARs you need for JMS and JNDI. You just need to add the sample classes from the code to your CLASSPATH environment variable.

To publish messages, run the NewsPublisher:

```
java <packagename>.NewsPublisher
```


Run both subscribers after your run the NewsPublisher to retrieve messages. Each class should retrieve its own copy of the messages published by NewsPublisher:

```
java <packagename>.EmailSubscriber  
java <packagename>.SMSSubscriber
```

Credits: The complete credit goes to the original content and author in the below link. Source code is available for download.

<http://www.devx.com/Java/Article/41921> [<http://www.devx.com/Java/Article/41921>]

Posted 2nd December 2012 by Unknown

 3 View comments