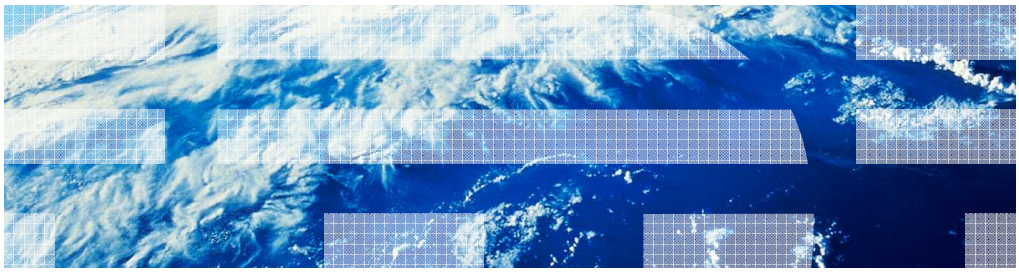
WebSphere Partner Gateway V6.2

Configuring JMS for receivers and destinations



This presentation focuses on the configuration steps necessary to use the JMS transport with WebSphere Partner Gateway 6.2. You will see how to create queue connection factories and queue definitions for receivers and destinations, using both WebSphere Application Server and JMSAdmin. At the end, you will see some steps showing the configuration setup necessary to run test transactions using the JMS protocol definitions configured for the WebSphere Partner Gateway elements.

Table of contents

- How to use the JMS transport with WebSphere Partner Gateway
- Common task: Create the queues
- Configuring JMS using WebSphere Application Server
- Configuring JMS using JMSAdmin

In this presentation you will be starting with specific information about how to use the JMS transport within WebSphere Partner Gateway.

Next, the queues creation, which is a common task, whether you want to proceed with configuring JMS using the WebSphere Application Server interface of JMSAdmin.

Then you will see the specifics of the different methods that you can use to create the JMS Queue Connection Factory - or QCF - and the queue definitions for the WebSphere Partner Gateway elements.

The WebSphere Partner Gateway configuration will follow and, as a final step for each method, you will see how to run a test transaction.

How to use the JMS transport with WebSphere Partner Gateway

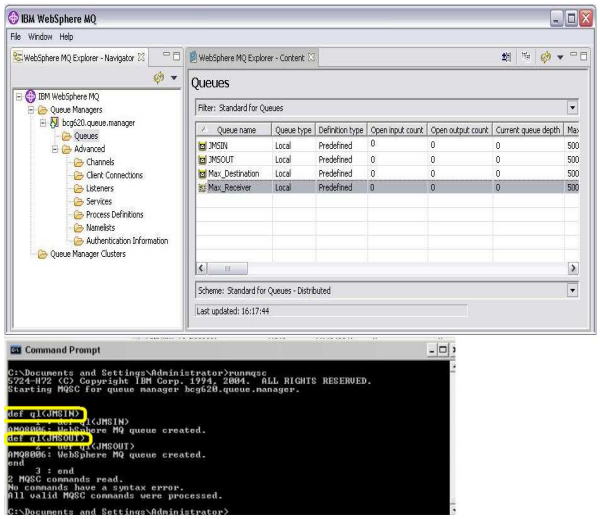
- WebSphere Partner Gateway allows the utilization of the JMS transport, using a WebSphere MQ provider, which can be configured using either WebSphere Application Server or WebSphere MQ to create:
  - 1 - Queue Connection Factory
  - 2 - JMS Queue definition
  - WebSphere Application Server: Use the Integrated Solutions Console
  - WebSphere MQ: Use JMSAdmin
- Install WebSphere MQ Server or Client on the machine hosting the WebSphere Partner Gateway component (DocMgr or Receiver) and Set WebSphere environment variable "MQ_INSTALL_ROOT"

You can perform the JMS integration using the WebSphere Message Queue product. In order to do that, you need to perform some configuration tasks, like creating QCFs and queue definitions, which can be done either using the WebSphere Application Server administrative console or JMSAdmin.

As shown at the bottom of this slide, you need access to the WebSphere MQ jars, which can be accomplished by setting the "MQ_INSTALL_ROOT" environment variable.

Common task: Create the queues

- The physical queues can be created using either tool:
  - MQ Explorer
  - runmqsc



The options for creating the physical queues in WebSphere MQ include using either the runmqsc command or the MQ Explorer graphical tool. Both methods are valid and reach the same result.

Configuring JMS using WebSphere Application Server

- Use WebSphere Application Server Integrated Solutions Console to create a JMS communication bridge between WebSphere Partner Gateway and WebSphere MQ:
  - 1 - Create JMS QCF (Queue Connection Factory)
  - 2 - Create JMS Queue definition
  - 3 - Configure JMS destination on WebSphere Partner Gateway
  - 4 - Configure JMS receiver on WebSphere Partner Gateway

This slide covers configuration of JMS using the WebSphere Application Server administrative console. The administration console is the tool to use for the first two tasks on the list shown in this slide, that is, create the JMS QCF, and create the JMS queue definition. The last two tasks, configuring the WebSphere Partner Gateway JMS receiver and destination, are performed in the WebSphere Partner Gateway console.

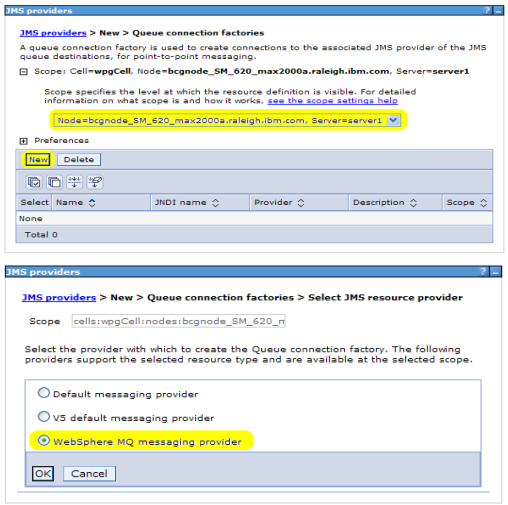
Create the JMS QCF (1 of 2)

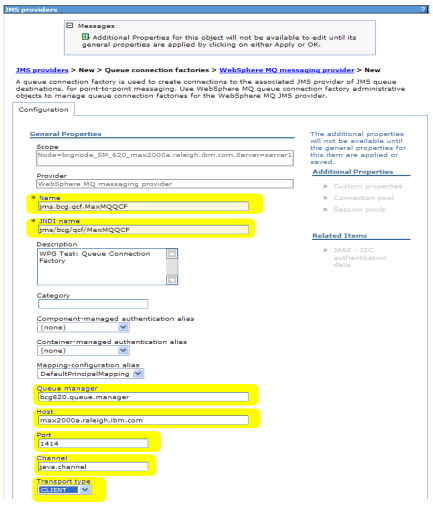
- In WebSphere Application Server Integrated Solutions Console select:
  - Resources->JMS->Queue connection factory
    - A - Simple Mode: From the "scope" pull-down, select the "server1" scope
    - B - Distributed Mode: From the “scope” pull-down, select scopes:
      - For “Destinations”: The "DocMgr" server/node
      - For "Receivers“: The "Receiver" server/node
  - Click "New" and select “WebSphere MQ messaging provider“. Next click "OK“ and configure the required fields as shown in the next slide. Then save it to the master configuration.

In order to create the JMS Queue Connection Factory, you need to select "Resource>JMS>Queue Connection Factory" from the WebSphere Application Server administrative console.

Then, depending on the WebSphere Partner Gateway installed "mode", select the appropriate scope. Select "WebSphere MQ messaging provider" in the next panel and proceed on to configuring the required fields. Finally, save the changes to the master configuration.

Create the JMS QCF (2 of 2)





This slide shows screen captures of what has been mentioned earlier. The top left panel shows the selection to be made in the WebSphere Application Server administrative console and the scope needed, depending on whether WebSphere Partner Gateway is installed in Simple or Distributed mode. The bottom left panel shows the selection for WebSphere MQ as the JMS resource provider. The rightmost panel shows the attribute values to be entered.

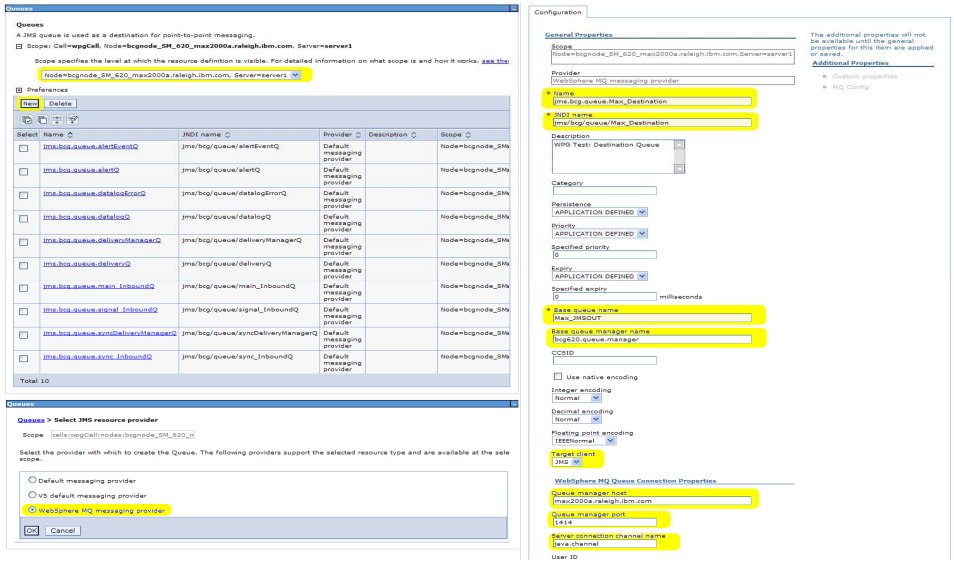
Create the JMS queues

- In WebSphere Application Server Integrated Solutions Console select:
  - Resources->JMS->Queues
    - A - Simple Mode: From the "scope" pull-down, select the "server1" scope
    - B - Distributed Mode: From the “scope” pull-down, select scopes:
      - For “Destinations”: The "DocMgr" server/node
      - For "Receivers“: The "Receiver" server/node
  - Click the "New" button and select "WebSphere MQ messaging provider“. Next click "OK“ and configure the required fields as shown in the next two slides. Then save it to the master configuration.

The next step is to create the JMS queue definitions. Select the "Resource>JMS>Queues” and, same as you saw earlier, select the scope, depending on the WebSphere Partner Gateway mode.

Then click "New“, select WebSphere MQ as the messaging provider, and then configure the required fields, which you will see in detail on the next slide. After entering the value, save the changes to the master configuration.

Create the JMS destination queue



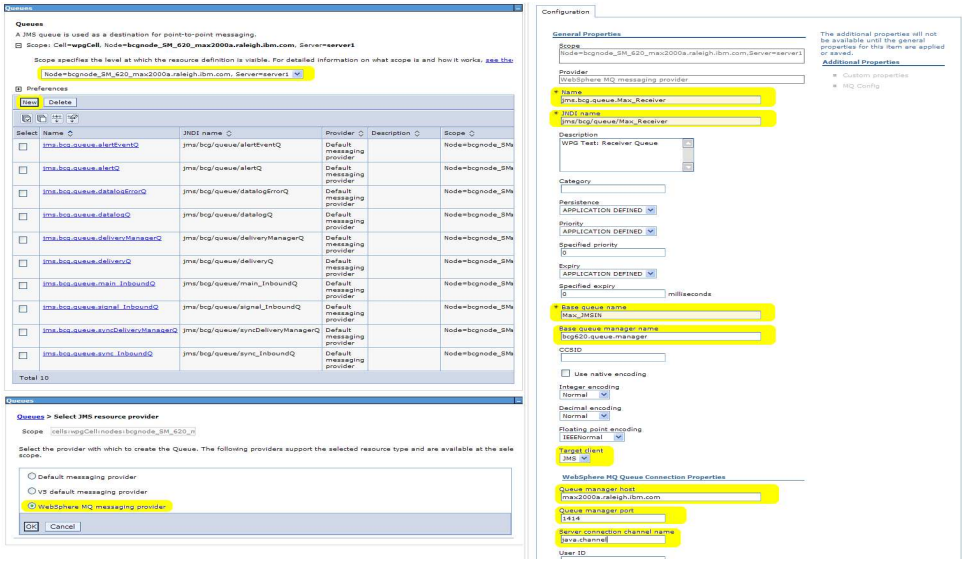
Here are the graphical details for the JMS destination queue, shown by screen captures:

The top left panel shows the selection to be made in the WebSphere Application Server administrative console and the scope needed depending on whether WebSphere Partner Gateway is installed in Simple or Distributed mode.

The bottom left panel shows the selection for WebSphere MQ as the messaging provider.

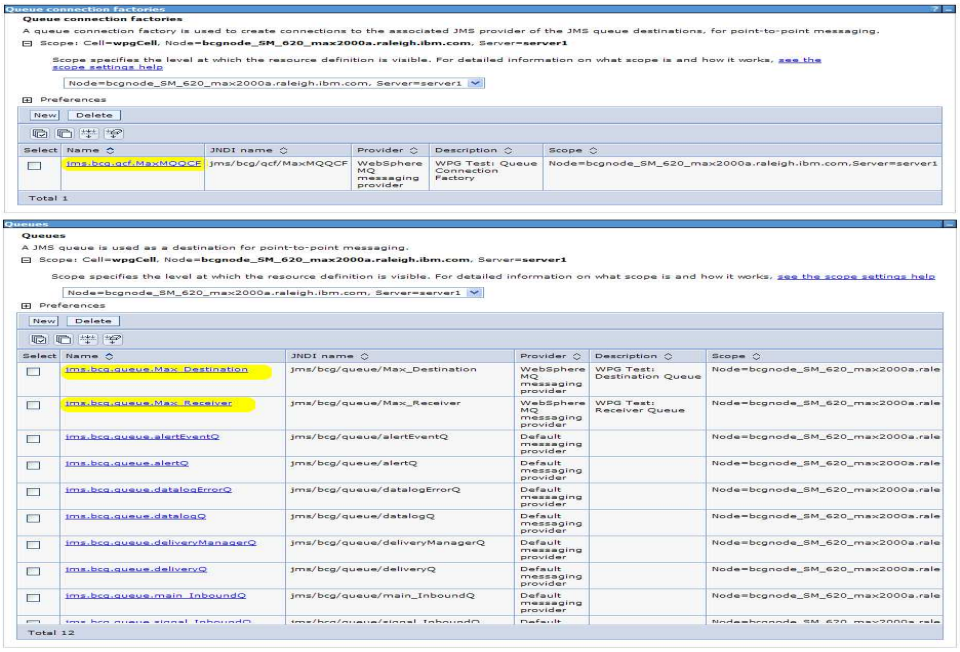
And the rightmost panel shows the attribute values to be entered, such as the destination queue name, JNDI name, base queue name, base queue manager, target client, queue manager host name and port, and the channel name.

Create the JMS receiver queue



The considerations in the previous slide for the destination are also applicable for the receiver, as shown here.

Review the new QCF and queues



Here is a summary of what the configuration made in the previous slides looks like. The top panel shows the newly created QCF, and the bottom panel shows receiver and destination queue definitions.

JMS WebSphere Application Server configuration completion

- Once the QCF and queues configuration are completed
  - Restart the affected WebSphere Partner Gateway servers

When the JMS configuration in WebSphere Application Server is completed, and if your WebSphere Partner Gateway is installed in Distributed mode, then restart the Deployment Manager.

JMS configuration in WebSphere Partner Gateway

- Use the WebSphere Partner Gateway Console to create:
  - JMS receiver
  - JMS destination
- Use the “Queue Connection Factory” and “Queue” objects information, previously created in WebSphere Application Server

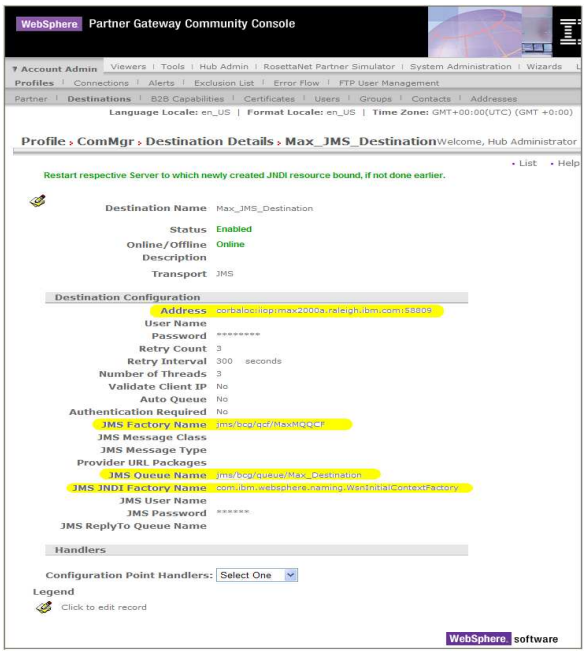
Now that you are finished with the WebSphere Application Server side, you need to complete the JMS configuration in WebSphere Partner Gateway. This task implies creating a JMS receiver and a JMS destination. Use the QCF and queue objects information that were created in the WebSphere Application Server administrative console, as illustrated in the previous slides.

Create a JMS destination (1 of 2)

- Configure the JMS Destination using the following values:
  - Address: corbaloc:iiop:< localhost >:<bootstrapportnumber>
  - JMS Factory Name: jms/bcg/qcf/MaxMQQCF
  - JMS Queue Name: jms/bcg/queue/Max_Destination
  - JMS JNDI Factory Name: com.ibm.websphere.naming.WsnInitialContextFactory

To create a JMS Destination, you need to provide values for the fields shown here. The address is composed of the "corbaloc:iiop" protocol, the host name of the machine, and the WebSphere Application Server bootstrap port number. Then provide the JMS QCF and Queue, as defined in WebSphere Application Server. Finally, provide the JNDI factory name used by WebSphere Application Server.

Create a JMS destination (2 of 2)



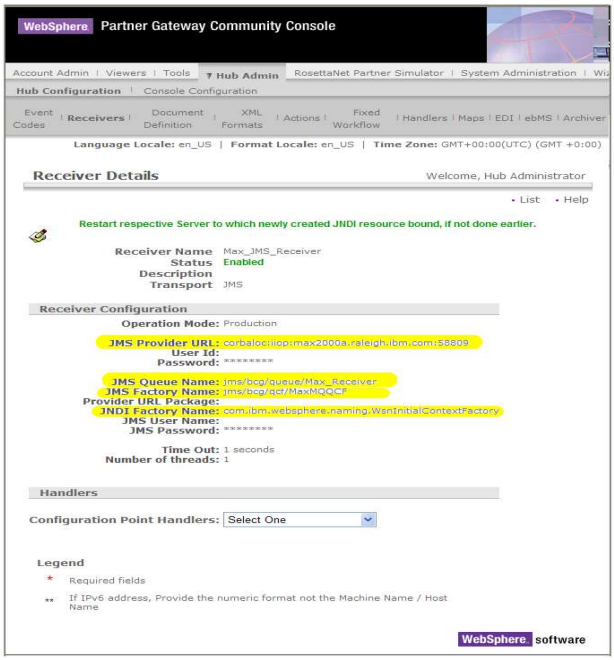
Here are the WebSphere Partner Gateway console screen captures with sample destination configuration values. Marked in yellow are the fields discussed in the previous slide. Note that after you save these values, the destination component needs to be restarted for the values to take effect.

Create a JMS receiver (1 of 2)

- Configure the JMS Receiver using the following values:
  - Address: corbaloc:iiop:<localhost>:<bootstrapportnumber>
  - JMS Factory Name: jms/bcg/qcf/MaxMQQCF
  - JMS Queue Name: jms/bcg/queue/Max_Receiver
  - JMS JNDI Factory Name: com.ibm.websphere.naming.WsnInitialContextFactory

The same considerations made for the destination are repeated in this slide for the receiver. Of course the queue name here is different since it needs to reference the specific queue that was created for the receiver.

Create a JMS receiver (2 of 2)



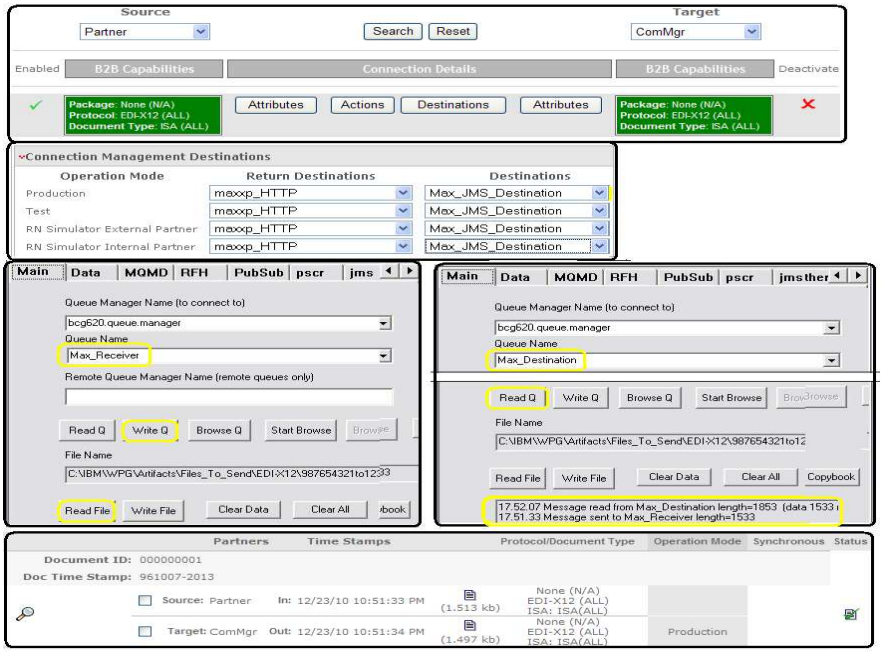
This is the screen capture with the graphical image of the values entered in the WebSphere Partner Gateway console to define the JMS Receiver.

Run test using corbaloc:iiop provider (1 of 2)

- Create a Connection between two TPs:
  - None,EDI-X12,ISA->None,EDI-X12,ISA
  - Select the “corbaloc:iiop” JMS Destination
- Use RHUtil to submit the EDI file in the Receiver’s JMS queue, that is: <Read File> and then <Write Queue>
- Use RHUtil again to check that the file has been dropped into the Destination JMS queue, that is: <ReadQ>
- Check the transaction in the document viewer

Now that you have completed the definitions in both WebSphere Application Server and WebSphere Partner Gateway consoles, you are ready to run a test. In order to do that, you need to create a connection between two trading partners. In this example, a “None,EDI-X12,ISA” channel has been set up for both directions. Select the JMS defined Destination for that connection. Then, submit the EDI file in the receiver's JMS queue, using the RFHUtil tool. Next you can check the destination queue to verify that the file has actually been delivered. The WebSphere Partner Gateway console document viewer can also be used to monitor the successful completion of the transaction.

Run test using corbaloc:iiop provider (2 of 2)



This slide shows a summary of the test that was run. In the top image is shown the None,EDI-X12,ISA connection. The second image shows the JMS Destination selection. The Third image shows the RFHUtil tool used to submit the EDI file to the receiver queue – show in the left image - and the EDI file delivered to the destination queue - shown in the right image. The bottom picture shows the WebSphere Partner Gateway console Viewer reporting the transaction successful completion. This slide concludes the JMS setup when using WebSphere Application Server as the JNDI provider.

Configuring JMS using JMSAdmin

- Use WebSphere Application Server as your JNDI provider for these benefits:
  - No dependence on Sun provider jars
  - Can be managed from WebSphere Application Server administrative console instead of a separate tool
  - In a clustered environment, connecting to a bootstrap server (as in WebSphere Application Server provider) can take advantage of failover
  - If using the .bindings file, then you need to secure this directory
- Use JMSAdmin to create a file containing the information necessary for WebSphere Partner Gateway to communicate with WebSphere MQ:
  - 1 - Create a .bindings file
  - 2 - Configure JMS Destination on WebSphere Partner Gateway
  - 3 - Configure JMS Receiver on WebSphere Partner Gateway

Starting with this slide, you will see the process of using JMSAdmin to create a .bindings file, which contains all the necessary information to communicate with WebSphere MQ. Once the .bindings file is available, you will reference its location in the file system to configure the WebSphere Partner Gateway receiver and destination.

Create a .bindings file using JMSAdmin (1 of 3)

- Update WMQ\Java\bin\JMSAdmin.config as follows:
  - INITIAL_CONTEXT_FACTORY = com.sun.jndi.fscontext.RefFSContextFactory
  - PROVIDER_URL = file:/C:/WPG/JMS

Before starting the JMSAdmin tool, you need to update a couple of attribute values in its configuration file. Provide the value for the "INITIAL_CONTEXT_FACTORY" attribute, and assign a path for the "PROVIDER_URL" attribute, which holds the file system location where the .bindings file is written. After setting the values for these two attributes, save the configuration file and start the JMSAdmin tool.

Create a .bindings file using JMSAdmin (2 of 3)

- Start WMQ\Java\bin\JMSAdmin and enter the following commands:
  - define ctx(jms)
  - change ctx(jms)
  - define qcf(WBICHub) tran(CLIENT) host(max2000a.raleigh.ibm.com) port(1414) chan(java.channel) qmgr(bcg620.queue.manager)
  - define q(inQ) queue(JMSIN) qmgr(bcg620.queue.manager)
  - define q(outQ) queue(JMSOUT) qmgr(bcg620.queue.manager)
  - InitCtx/jms> end
- The .bindings file is created in C:\WPG\JMS\jms folder

When the JMSAdmin tool starts, it is in command-line interactive mode, and prompts you for the JMS context. In that context, define the necessary attributes, such as the queue connection factory, the transport being used, the host name where the Queue Manager resides and the port it listens to, the channel type, and the Queue Manager name. Then two queues need to be defined - one to be used by the receiver and one by the destination. When all these are done, you can exit the JMSAdmin application and find the .bindings file created in the file system path specified in the JMSAdmin configuration file.

Create a .bindings file using JMSAdmin (3 of 3)

This slide shows a screen capture of the JMSAdmin interactive mode used to create the .bindings file for your lab.

JMS Configuration in WebSphere Partner Gateway

- Use the WebSphere Partner Gateway console to create:
  - JMS receiver
  - JMS destination
- Reference the .binding file directory location for the information necessary to communicate with WebSphere MQ

Now that the .bindings file is created, you can proceed with configuring your WebSphere Partner Gateway objects: a JMS receiver, and a JMS destination. In the configuration panels of these two objects, you will provide the location of the .bindings file, which holds all the necessary information for the WebSphere Partner Gateway objects to communicate with the WebSphere MQ Queue Manager.

Create a JMS destination (1 of 2)

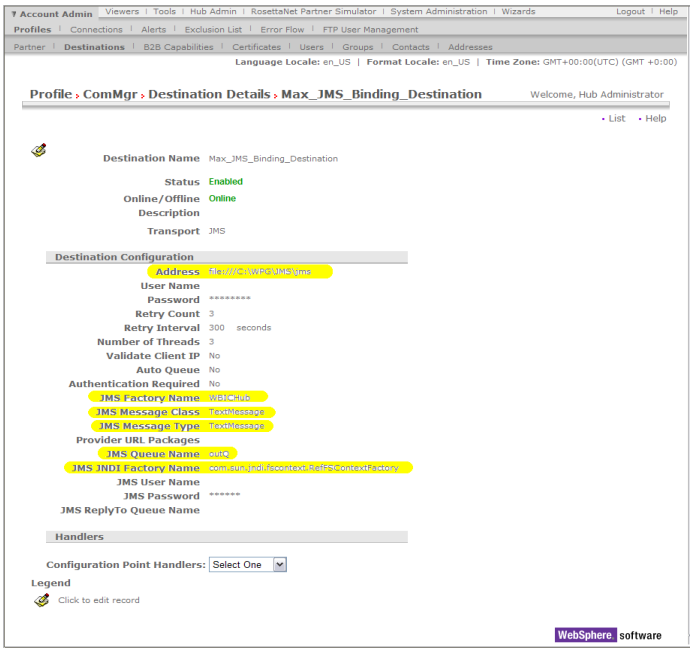
- Configure the JMS Destination using the following values:
  - Address: file:///C:/WPG/JMS/jms
  - JMS Factory Name: WBICHub
  - JMS Queue Name: outQ
  - JMS JNDI Factory Name: com.ibm.websphere.naming.WsnInitialContextFactory
  - JMS Message Class: TextMessage
  - JMS Message Type: TextMessage

Take a closer look at the specifics of the WebSphere Partner Gateway objects configuration, starting with the JMS Destination:

The Address attribute now points to the .bindings file location, whereas it was using a composite corbaloc:iiop address when the JMS configuration was performed in WebSphere Application Server.

Then you need to provide the values for the other attributes, such as the JMS factory name, the destination queue name, the JNDI factory name, and the required JMS message class and message type.

Create a JMS destination (2 of 2)



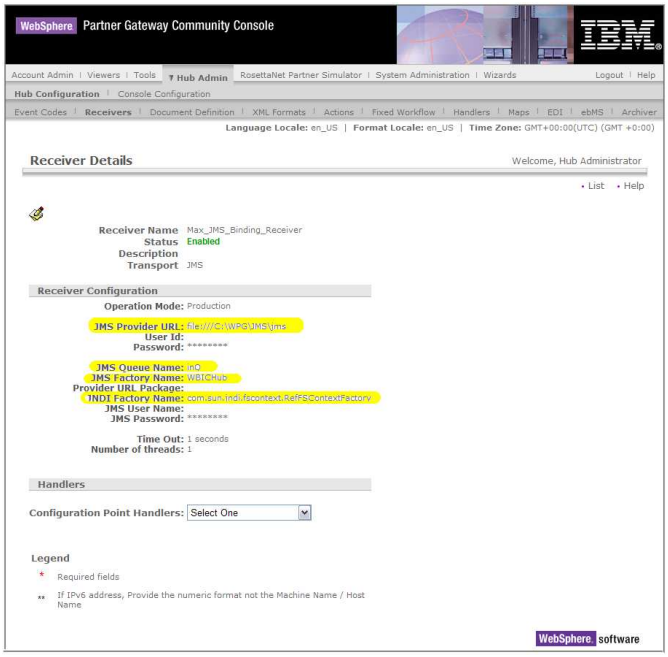
This is a WebSphere Partner Gateway Console screen capture showing the actual JMS Destination configuration panel.

Create a JMS receiver (1 of 2)

- Configure the JMS Receiver using the following values:
  - Address: file:///C:/WPG/JMS/jms
  - JMS Factory Name: WBICHub
  - JMS Queue Name: inQ
  - JMS JNDI Factory Name: com.ibm.websphere.naming.WsnInitialContextFactory
  - JMS Message Class: TextMessage
  - JMS Message Type: TextMessage

Just as you saw earlier for the destination, the same configuration considerations are valid for the receiver, which includes pretty much the same attributes. Those are the address attribute to point to the .bindings file location, the JMS factory name, the receiver queue name, the JNDI factory name, and the JMS message class and message type values.

Create a JMS receiver (2 of 2)



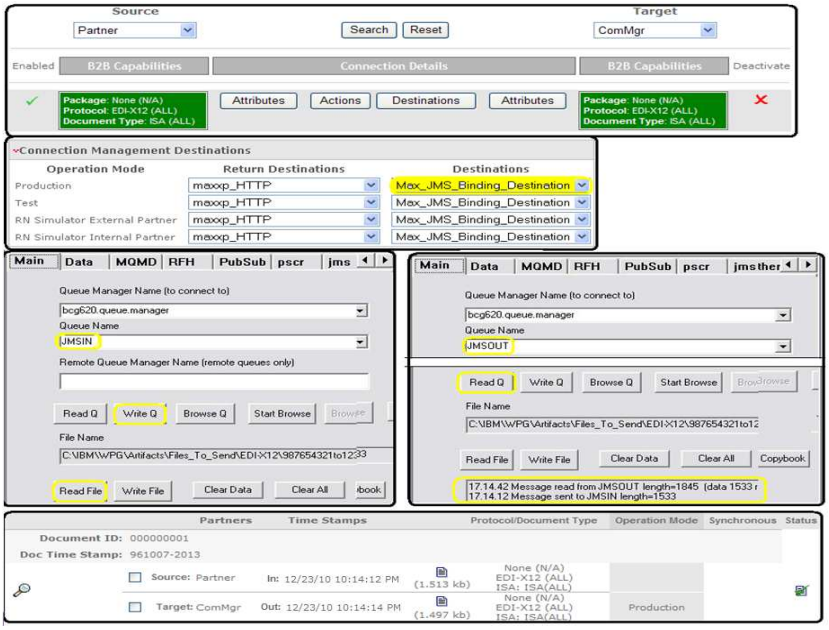
Here is the screen capture of the JMS receiver configuration panel from the WebSphere Partner Gateway console.

Run test using .bindings file provider (1 of 2)

- Create a Connection between two TPs:
  - None,EDI-X12,ISA->None,EDI-X12,ISA
  - Select the ".bindings file" JMS Destination
- Use RHUtil to submit the EDI file in the Receiver's JMS queue, I.E.: <Read File> and then <Write Queue>
- Use RHUtil again to check that the file has been dropped into the Destination JMS queue, that is: <ReadQ>
- Check the transaction in the document viewer

As the last item, you can run the same test for this configuration as the one you ran configuring JMS using WebSphere Application Server administrative console. You need to create a connection between two trading partners, in this example, a "NONE,EDI-X12,ISA" channel for both directions. Then select the destination configured to use the .bindings file. Next, submit the EDI file in the Receiver's JMS queue using the RFHUtil tool. Observe the transactions results and verify that the file has actually been delivered to the destination queue. The WebSphere Partner Gateway console document viewer can also be used to monitor the successful completion of the transaction.

Run test using .bindings file provider (2 of 2)



This slide contains a summary of the test that you ran:

In the top picture is shown the None,EDI-X12,ISA connection. The second picture shows the JMS Destination selection. The third picture shows the RFHUtil tool used to submit the EDI file to the receiver queue – in left image - and the EDI file delivered to the destination queue - shown in the right image. The bottom picture shows the WebSphere Partner Gateway console Viewer reporting the transaction successful completion.

Summary

- How to use the JMS transport with WebSphere Partner Gateway
- Creating the queues
- Configuring JMS using WebSphere Application Server
- Configuring JMS using JMSAdmin

In this presentation, you have seen how to use the JMS transport within WebSphere Partner Gateway, how to create the queues, and two ways to create the JMS queue connection factory and the queue definitions for the WebSphere Partner Gateway elements. Finally, you saw how to run a test transaction.

Feedback

Your feedback is valuable

You can help improve the quality of IBM Education Assistant content to better meet your needs by providing feedback.

- Did you find this module useful?
- Did it help you solve a problem or answer a question?
- Do you have suggestions for improvements?

Click to send email feedback:

mailto:iea@us.ibm.com?subject=Feedback_about_WPGV62_ConfiguringJMSReceiversDestination.ppt

This module is also available in PDF format at: ../WPGV62_ConfiguringJMSReceiversDestination.pdf

You can help improve the quality of IBM Education Assistant content by providing feedback.

Trademarks, disclaimer, and copyright information