An Oracle White Paper
June 2011

WebSphere MQ – Oracle Enterprise Gateway Integration Guide

Disclaimer

The following is intended to outline our general product direction. It is intended for information purposes only, and may not be incorporated into any contract. It is not a commitment to deliver any material, code, or functionality, and should not be relied upon in making purchasing decisions. The development, release, and timing of any features or functionality described for Oracle's products remains at the sole discretion of Oracle.

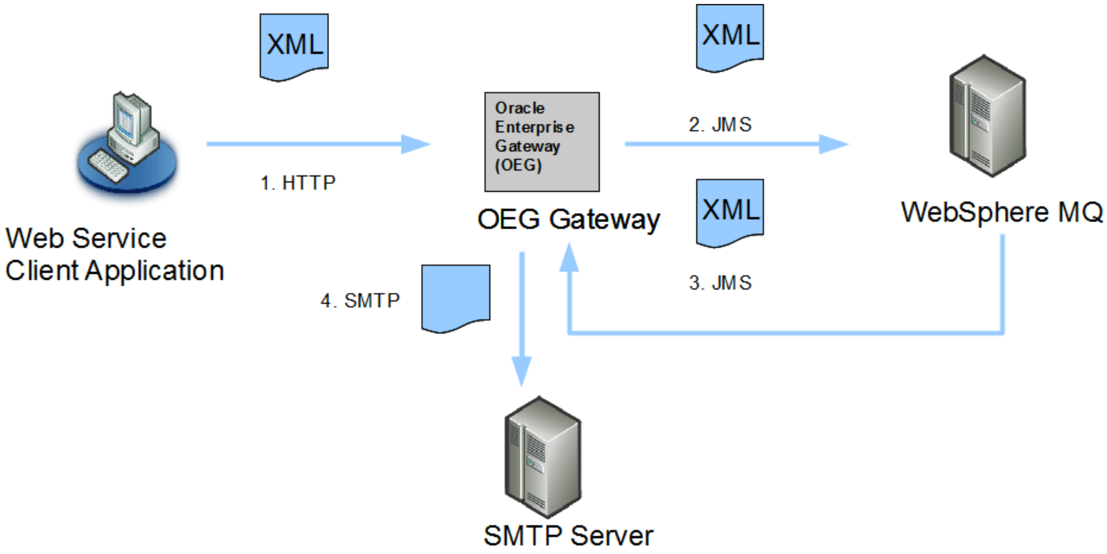
1. Introduction

1.1. Purpose

This document describes how to configure the Gateway to perform protocol translation. This will be demonstrated by the following:

1. The Gateway will listen for messages on a HTTP interface. Messages read from this interface will be placed on a message queue.
2. The Gateway will listen for messages on a message queue. Messages read from the queue will be sent to an email account via SMTP.

The message flow is as follows:



This guide applies to software products, from version 6.0 upwards.
In this guide the message queuing system that will be used is **IBM WebSphere MQ**.

1.2. JMS Architecture

The Gateway utilises JMS (Java Message Service) for sending and receiving messages from messaging systems. JMS API which was developed by Sun defines a common set of interfaces and associated semantics that allow the Gateway to communicate with various messaging applications in a standard way.

Messaging system products (IBM WebSphere MQ, JBossMQ, SonicMQ, Fiorano, and OpenJMS) provide implementations of JMS which can be plugged into the Gateway.

The Gateway has been designed to allow 3rd party JMS providers to be "plugged in". To plug in new JMS providers, you must install the JMS provider on the Gateway machine.

The messaging system vendors can provide an implementation of the JMS provider which is normally in the form of jar files and configuration settings to be entered in the OEG Policy Studio.

1.3. Prerequisites

1. IBM WebSphere MQ is available from http://www.ibm.com
2. OEG GatewaySoftware available from http://www.oracle.com
3. Java Virtual Machine. WebSphere MQ will run on any platform where there is a suitable Java 2 runtime environment

1.4. Setup Used for this Guide:

- WebSphere installed on Windows to D:\IBM\WebSphere MQ
- OEG Gateway 6.0 software installed to C:\OEG\gateway-6.0.0 on Windows
- JMS and JNDI subfolder and .bindings file created in D:\JMS\JNDI on Windows

1.5. Configuration Steps

1. Download and install WebSphere MQ
2. Configure WebSphere MQ
3. Install OEG Gateway

4. Configure Gateway to send messages to WebSphere MQ
5. Configure Gateway to listen for messages from WebSphere MQ queue
6. Test Setup

2. Setting up the WebSphere MQ environment

2.1. Download WebSphere MQ

WebSphere MQ is available from http://www.ibm.com
The version (at time of writing) was IBM WebSphere V7.0

2.2. Installation of WebSphere MQ

Once WebSphere MQ has been downloaded install it to a desired location
Please refer to WebSphere documentation for more information on installing WebSphere MQ.
For the purpose of this guide default options were used except the installation directory. A Windows platform was used.

2.3. Configure JMS Administered objects

Every connection factory or destination used by the OEG server must be registered in a JNDI directory with a unique name.
In this particular case a Sun File System context JNDI provider will be used. JNDI data will be stored in the file system directory in the .bindings file.
Using tools from the WebSphere MQ installation a .bindings file needs to be created. To do this the JMSAdmin tool will be used which is found in the following location of the WebSphere MQ installation:
/WebSphereMQ_Install_Dir/Java/bin

If no queues exist:

Create the .bindings file:

1. Create a new directory (for this guide D:\JMS was used)
2. Create a subfolder called 'JNDI' in the folder created above
3. Copy the JMSAdmin.config file into the newly created 'JMS' directory JMSAdmin.config is found in the /WebSphereMQ_Install_Dir/Java/bin directory

4. Edit the JMSAdmin.config so that it instructs JMSAdmin to create .bindings file in the "JNDI" subfolder of the current directory, do this by editing the contents of JMSAdmin.config to read the following:
   INITIAL_CONTEXT_FACTORY=com.sun.jndi.fscontext.RefFSContextFactory
   PROVIDER_URL=file:./JNDI

The JMSAdmin tool needs an additional file to tell it which JMS queue to configure.

1. Create a file called JNDIclient.scp in the 'JMS' folder,
2. Edit this file with a text editor. JMSAdmin will be instructed to create "JMSDEMOCF" connection factory with connection to "Insert_Host_Name_Here" port 1414 and queue called "JMSDEMOQueue" that maps to "QM_websphere_mq.QL" queue on the WebSphere MQ.
3. Enter the following text in the file JNDIClient.scp:

```
#-----------------------------------------------------------------
# Connection Factory for Client mode
# Delete the Connection Factory if it exists
DELETE  CF(JMSDEMOCF)

# Define the Connection Factory
DEFINE  CF(JMSDEMOCF) +
        SYNCPOINTALLGETS(YES) +
        TRAN(client) +
        HOST(Insert_Host_Name_Here) CHAN(SYSTEM.DEF.SVRCONN)
PORT(1414) +
        QMGR( )

#-----------------------------------------------------------------
# Queue Object
# Delete the Queue if it exists
DELETE  Q(JMSDEMOQueue)

# Define the Queue
DEFINE  Q(JMSDEMOQueue) QUEUE(QM_websphere_mq.QL)
End
```

4. Run the JMSAdmin tool with the following arguments:

JMSAdmin.bat < JNDIClient.scp.

Below is an example of the successful running of the JMSAdmin tool on Windows:

```
---------------------------------------------------------------------------------------------

D:\JMS>dir
 Volume in drive D has no label.
 Volume Serial Number is 9C54-4566

 Directory of D:\JMS

19/08/2008  12:00       <DIR>          .
19/08/2008  12:00       <DIR>          ..
19/08/2008  12:00                   94 JMSAdmin.config
19/08/2008  12:02       <DIR>          JNDI
19/08/2008  12:02                  613 JNDIClient.scp
               2 File(s)            707 bytes
               3 Dir(s)  12,119,982,080 bytes free


D:\JMS>"D:\IBM\WebSphere MQ\Java\bin\JMSAdmin.bat" < JNDIClient.scp

5724-H72, 5655-L82, 5724-L26 (c) Copyright IBM Corp. 2002,2005. All
Rights Reserved.
Starting Websphere MQ classes for Java(tm) Message Service
Administration

InitCtx> InitCtx> InitCtx> InitCtx> Binding non-administerable or not
found

InitCtx> InitCtx> InitCtx>
InitCtx> InitCtx> InitCtx> InitCtx> InitCtx> InitCtx> Binding non-
administerable or not found

InitCtx> InitCtx> InitCtx>
InitCtx> InitCtx>
Stopping Websphere MQ classes for Java(tm) Message Service
Administration

D:\JMS>

---------------------------------------------------------------------
```

5. Upon completion a .bindings file is created in JNDI subdirectory of the directory created in the first step (i.e. D:\JMS). These file needs to be copied to a host running the OEG server where the Gateway is installed.

If queues exist already:

Create the .bindings file:

1. Create a new directory (for this guide D:\JMS was used)
2. Create a subfolder called 'JNDI' in the folder created above
3. Copy the JMSAdmin.config file into the newly created 'JMS' directory JMSAdmin.config is found in the /WebSphereMQ_Install_Dir/Java/bin directory
4. Edit the JMSAdmin.config so that it instructs JMSAdmin to create .bindings file in the "JNDI" subfolder of the current directory, do this by editing the contents of JMSAdmin.config to read the following:
   INITIAL_CONTEXT_FACTORY=com.sun.jndi.fscontext.RefFSContextFactory
   PROVIDER_URL=file:./JNDI

The JMSAdmin tool needs an additional file to tell it which JMS queue to configure.

1. Create a file called JNDIclient.scp in the 'JMS' folder,
2. Edit this file with a text editor. JMSAdmin will be instructed to create "JMSDEMOCF" connection factory with connection to "Insert_Host_Name_Here" port 1414 and queue called "JMSDEMOQueue" that maps to "QM_websphere_mq.QL" queue on the WebSphere MQ.
3. Enter the following text in the file JNDIClient.scp

```
#----------------------------------------------
```

```
# Connection Factory for Client mode
# Delete the Connection Factory if it exists
DELETE  CF(JMSDEMOCF)

# Define the Connection Factory
DEFINE  CF(JMSDEMOCF) +
        SYNCPOINTALLGETS(YES) +
        TRAN(client) +
        HOST(Insert_Host_Name_Here) CHAN(SYSTEM.DEF.SVRCONN)
PORT(1414) +
        QMGR( )

#-----------------------------------------------------------------
```

4. Run the JMSAdmin tool with the following arguments:

   JMSAdmin.bat < JNDIClient.scp.

After the .bindings file has been created successfully do the following:

5. After you create the .bindings file , run (using JMSAdmin):

   alter q(JMSQueueName) targclient(MQ)

   for each queue (out, response, logging, etc). Where "JMSQueueName" is the name of the existing queue.

   This will modify the .bindings file with the information of the existing queues.

6. This step sets up all the correct parameters for WebSphere MQ for the existing queues.

2.4. Configuring WebSphere MQ

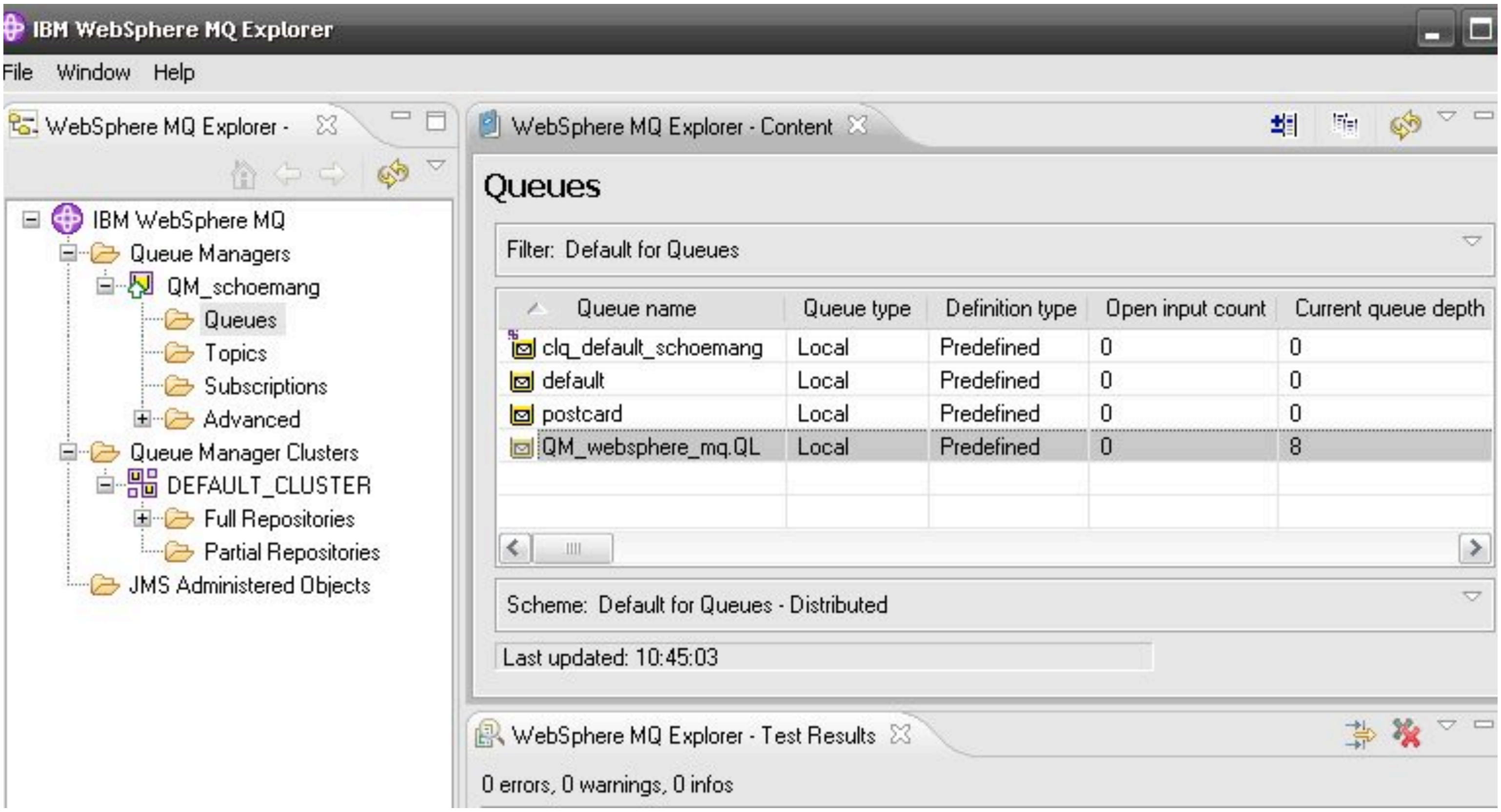
A Queue needs to be configured in WebSphere MQ. WebSphere MQ Explorer will be used to create the required queues.

1. Open "WebSphere MQ Explorer", this can be found at Start menu, and navigate to All Programs => IBM WebSphere MQ => WebSphere MQ Explorer.

2. Create a Queue underneath Queue Manager. To do this, expand the "Queue Manager" and right mouse click on the "Queues" node and select "New" => "Local Queue".
3. Enter the name of the new queue; in this case the queue is called "QM_websphere_mq.QL".

PLEASE NOTE: Use existing "Queue Manager" that has been created during the installation. In this walkthrough the "Queue Manager" has been called “QM_schoemang”.

This is WebSphere MQ Explorer showing the configured queues.



Queues

Filter: Default for Queues

Queue name	Queue type	Definition type	Open input count	Current queue depth
clq_default_schoemang	Local	Predefined	0	0
default	Local	Predefined	0	0
postcard	Local	Predefined	0	0
QM_websphere_mq.QL	Local	Predefined	0	8

Scheme: Default for Queues - Distributed

Last updated: 10:45:03

0 errors, 0 warnings, 0 infos

3. Setting up the OEG environment

3.1. Download OEG Gateway software

OEG provides copies of Gateway software to partners, customers, and evaluators.

3.2. Load WebSphere MQ Provider files onto the OEG Gateway

WebSphere MQ provides a particular JMS provider that the Gateway will use to connect to WebSphere MQ. The JMS provider takes the form of Java archive files (i.e. JAR files). Once WebSphere is installed it is a simple matter to copy the necessary JAR files from the WebSphere MQ installation and drop the JMS provider JAR files onto the OEG Gateway.
The .bindings file created in section 2.3 would also need to be copied to the Gateway host.

3.2.1 Instructions for Software install

1. Browse to /WebSphere_Install_Dir/lib directory
2. Copy the following jar files:
   - fscontext.jar
   - providerutil.jar
   - jta.jar
   - connector.jar
   - com.ibm.mq.jar
   - com.ibm.mqjms.jar
   - dhbcore.jar
3. Browse to the /OEG_Install_Dir/ext/lib directory and copy files into /lib directory

4. Configuring the Gateway to place messages on WebSphere MQ Queue

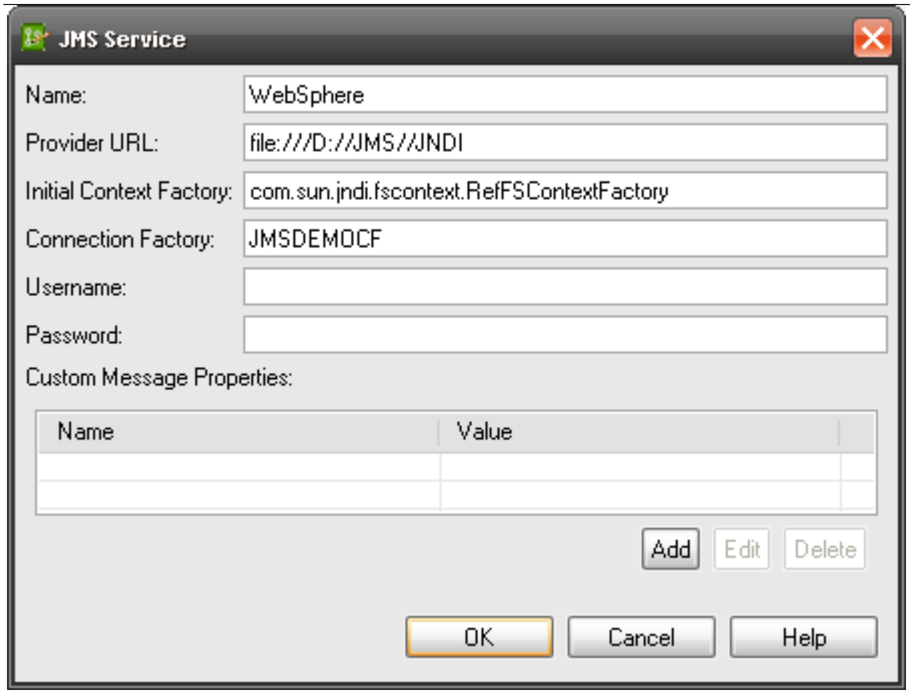
The gateway will be configured to place messages it receives on a queue (i.e. destination) named “JMSDEMOQueue” in WebSphere MQ.

4.1. Creating a JMS Session:

1. In Policy Studio with the **External Connections** panel selected right click on the **JMS Services** and **Add a JMS Service** and configure the following fields:
   - Name: WebSphere MQ
   - Provider URL: file:///D://JMS//JNDI:, where D:///JMS//JNDI points to the location of the .bindings file located on the Gateway installed on Windows in this case (D:///JMS//JNDI is pointing to a path on the demo machine used for this guide which is d:\JMS\JNDI)
     If the .bindings file was uploaded to an Appliance then the Provider URL would be configured like this:
     Provider URL: file:///opt/OEG/OEG_product_Dir/ext/lib
   - Initial Context Factory:
     com.sun.jndi.fscontext.RefFSContextFactory
   - Connection Factory: JMSDEMOCF
   - Username: leave blank (or ask MQ service administrator)
   - Password: leave blank
2. Click OK
3. With the **Services** panel selected right click on the “OEG Gateway” **Process.**
4. Select **Messaging System -> Add JMS Session**
5. Set JMS Service to “Websphere MQ” just configured above.
6. Choose not to allow duplicates.
7. Click **OK**

PLEASE NOTE: The provider URL will differ depending on where the .bindings file is located.
It should be located in a folder of ANY name on the Appliance or host where the Gateway is running.

The JMS Session configuration for Windows:

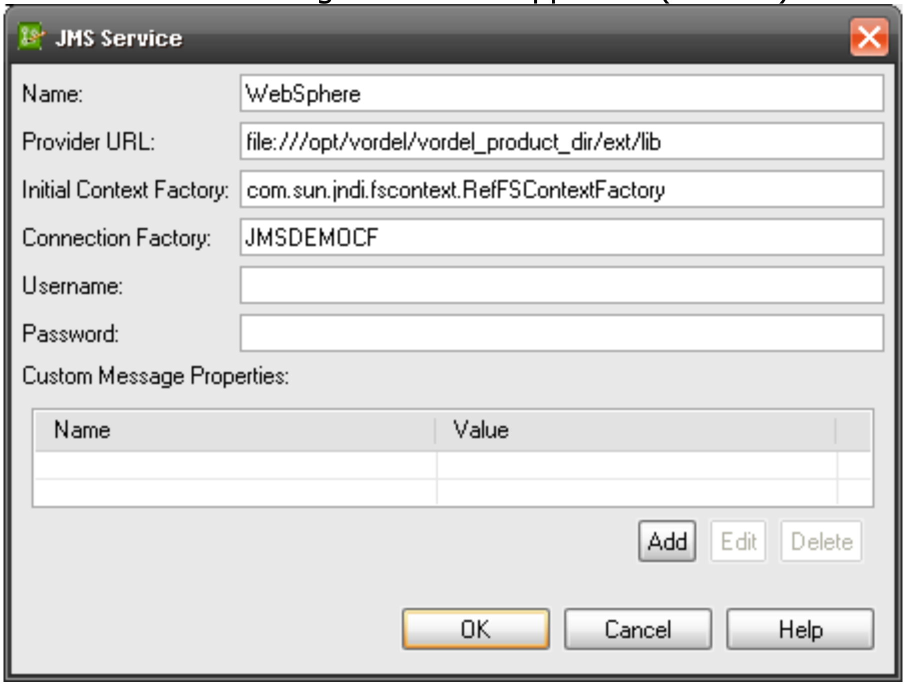


Name:	WebSphere
Provider URL:	file:///D://JMS//JNDI
Initial Context Factory:	com.sun.jndi.fscontext.RefFSContextFactory
Connection Factory:	JMSDEMOCF
Username:	
Password:	

Custom Message Properties:

Name	Value

The JMS Session configured for the Appliance (VX4000):



Name:	WebSphere
Provider URL:	file:///opt/vordel/vordel_product_dir/ext/lib
Initial Context Factory:	com.sun.jndi.fscontext.RefFSContextFactory
Connection Factory:	JMSDEMOCF
Username:	
Password:	

Custom Message Properties:

Name	Value

4.2. Create a “Route to WebSphere” Policy

Create a small test policy to route messages on to the WebSphere queue by completing the following steps:

1. Open the OEG Policy Studio with the **Policies** panel selected.
2. Right click on **Policies** and select **Add Policy**.
3. Enter the new Policy name “Route to WebSphere” and click **OK.**
4. Open Services panel and right click on **Default Services** in the expanded OEG Gateway process.
5. Create a new relative path on the Gateway Process called “/ToWebSphere”.
6. Map the /ToWebSphere path to the policy called “Route to WebSphere. This means that when a message is received by the Gateway on the path “/ToWebSphere”, it will be passed to the “Route to WebSphere” policy, which will then process the message.

Configuring the Messaging System Filter

When a policy that routes to a JMS provider (such as WebSphere MQ) is created, the policy must contain a Messaging System filter, which can be found under the Routing category of filters in the Policy Studio. To configure this filter, complete the following steps:

NOTE: Make sure that the JMS Service has been configured already (see section 4.1) as the available JMS service will need to be refered to when configuring the Messaging System Filter

1. Go back to the “Route to Websphere” filter and drag a **Messaging System** filter from the **Routing** group located in the pallette on the right of the Policy Studio.
2. Under the **Request** tab select the **JMS Service** that has been configured above (titled “WebSphere MQ”) from the JMS Session dropdown.
3. Set the Destination to “JMSDEMOQueue” (This is the queue refered to in the .bindings file created in section 2.3)
4. The Message Type should be specified. Change this to “Use content.body attribute to create a message in the format specified in the ‘SOAP OVER Java Message Service’".
5. All other settings may be left at default.
6. Click on the Response tab and select ‘No Response’
7. Click on Ok

Note on Message Type:
Here is an explanation of how the various serializations (from OEG message to JMS message work)

- Use content.body attribute to create a message in the format specified in the "SOAP over Java Messaging Service" proposal:
If this option is selected, messages will be formatted according to the SOAP over JMS proposal. This is the default option since, in most cases; it is the message body that is to be routed to the messaging system. This will result in a ByteMessage being sent to the queue/topic and JMS a property will contain the Content-Type (i.e. text/xml)

- Create a MapMessage from the java.lang.Map in the attribute named below:
Select this option to create a javax.jms.MapMessage from the OEG message attribute named below that consists of name-value pairs.

- Create a ByteMessage from the attribute named below:
Select this option to create a javax.jms.ByteMessage from the OEG message attribute named below.

- Create an ObjectMessage from the java.lang.Serializable in the attribute named below:
This option can be selected in order to create a javax.jms.ObjectMessage from the OEG message attribute named below.

- Create a TextMessage from the attribute named below:
A javax.jms.TextMessage can be created from the message attribute named below by selecting this option from the dropdown.

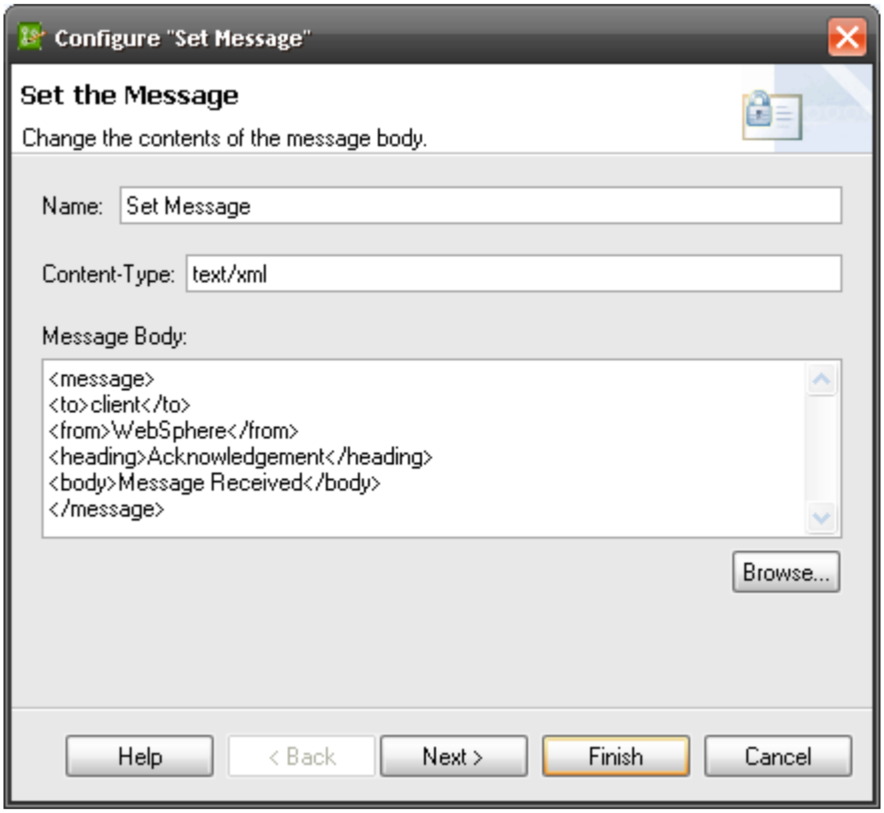
To complete the test policy create the following flow:

1. Messaging System Filter: Located in the "Routing" group. This filter should be configured as described above. This is a mandatory filter in the policy.
2. Set Message Filter: Located in the "Conversion" group. Used to set the content of an XML response message that can be returned to the client to acknowledge that the message has been placed on

the WebSphere queue. This step is not mandatory, but generates an acknowledgement of the message placed on the queue.

- Drag the Set Message filter onto the policy canvas and add message content as below.

Example of the configuration of the Set Message filter:



Name: Set Message

Content-Type: text/xml

Message Body:

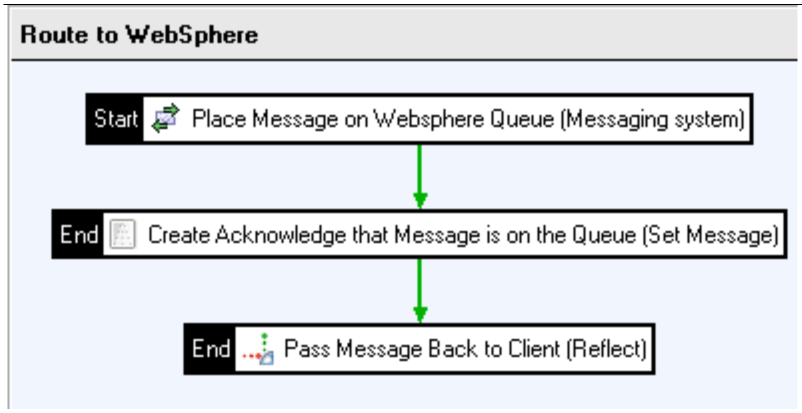
```
<message>
<to>client</to>
<from>WebSphere</from>
<heading>Acknowledgement</heading>
<body>Message Received</body>
</message>
```

3. Reflect Filter: Routes the customized response back to the client if necessary. The Reflect filter can be found under the Utility category of filters.

   Drag the Reflect filter onto the policy canvas.

   Link the 3 filters together with success arrows as below

   Once configured, the policy will appear as follows:



Route to WebSphere

Start
Place Message on Websphere Queue (Messaging system)
End
Create Acknowledge that Message is on the Queue (Set Message)
End
Pass Message Back to Client (Reflect)

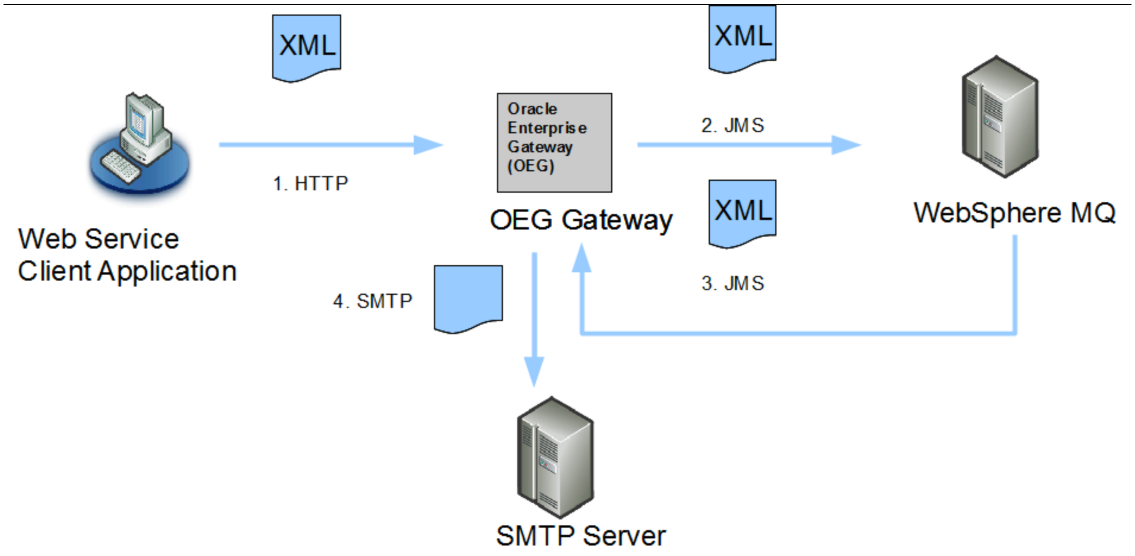
4.3. Ensure policies are updated on the Gateway

- Open the Policy Studio.
- Click on Settings.
- Select **Deploy** to ensure that the changes made are propagated to the live Gateway. Click **Yes** to deploy changes to the server.

4.4. Test the configuration to place message on WebSphere Queue

OEG Service Explorer will be used as the client to test the integration. The entire transaction will be tested from the client, through the Gateway, and on to a WebSphere queue.

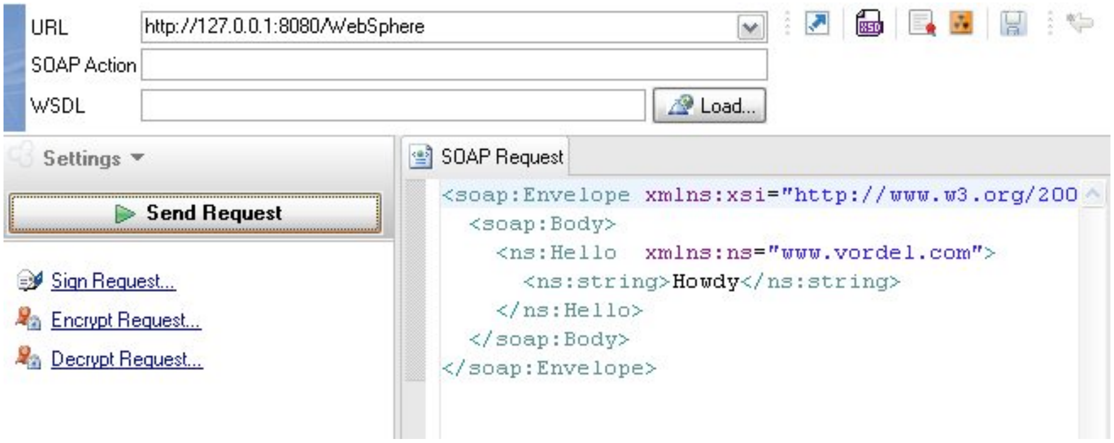
The following diagram shows the solution architecture:



4.5. Load a sample message into OEG Service Explorer

Load a sample XML message into OEG Service Explorer. Ensure that the URL field in OEG Service Explorer points to the Gateway and in particular to the "WebSphere" path on the Gateway.

The screenshot below shows a sample SOAP Request loaded in the OEG Service Explorer:



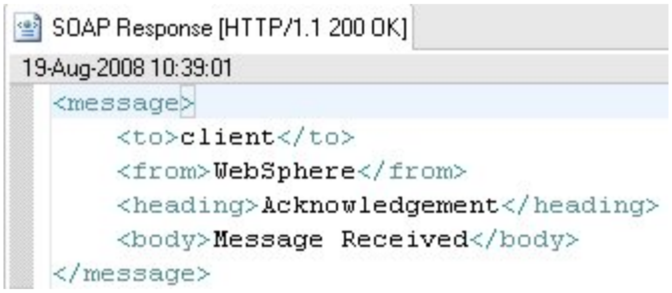
URL http://127.0.0.1:8080/WebSphere
SOAP Action
WSDL

```
<soap:Envelope xmlns:xsi="http://www.w3.org/200
  <soap:Body>
    <ns:Hello  xmlns:ns="www.vordel.com">
      <ns:string>Howdy</ns:string>
    </ns:Hello>
  </soap:Body>
</soap:Envelope>
```

4.6. Send Message and check WebSphere MQ queue

By sending messages using OEG Service Explorer, the Gateway will route the messages to the WebSphere MQ queue, in this case "QM_websphere_mq.QL".
Once test messages have been sent, open the WebSphere MQ Explorer. It should show a total number of messages that have been delivered to "QM_websphere_mq.QL".

Having sent the SOAP request, the response will be displayed in the SOAP Response panel, as displayed in the screenshot below:

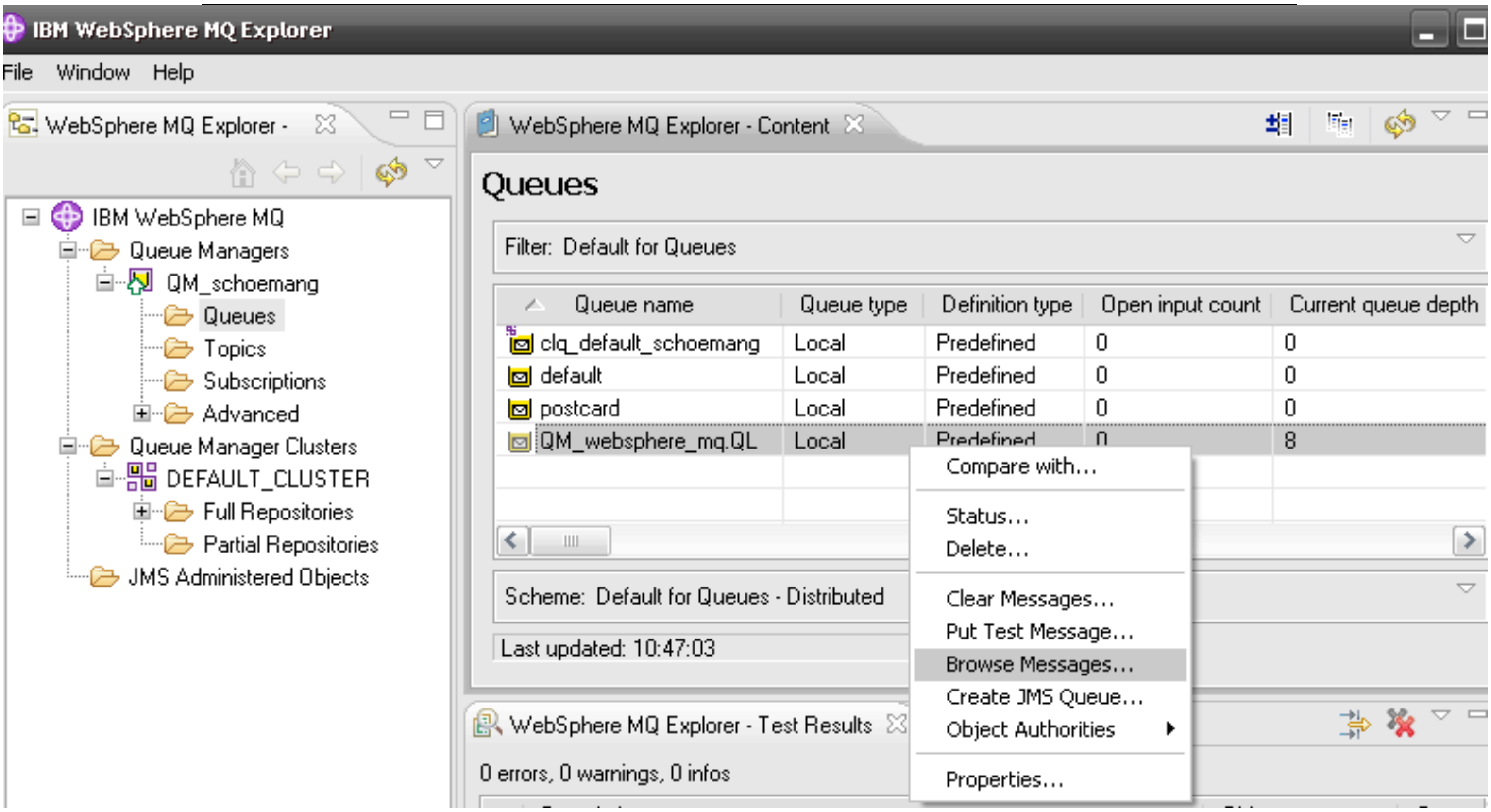


SOAP Response [HTTP/1.1 200 OK]

19-Aug-2008 10:39:01

```
<message>
    <to>client</to>
    <from>WebSphere</from>
    <heading>Acknowledgement</heading>
    <body>Message Received</body>
</message>
```

The screenshot below shows the WebSphere MQ Explorer Console showing the messages on "QM_websphere_mq.QL". 8 messages have been placed on this queue:



Filter: Default for Queues

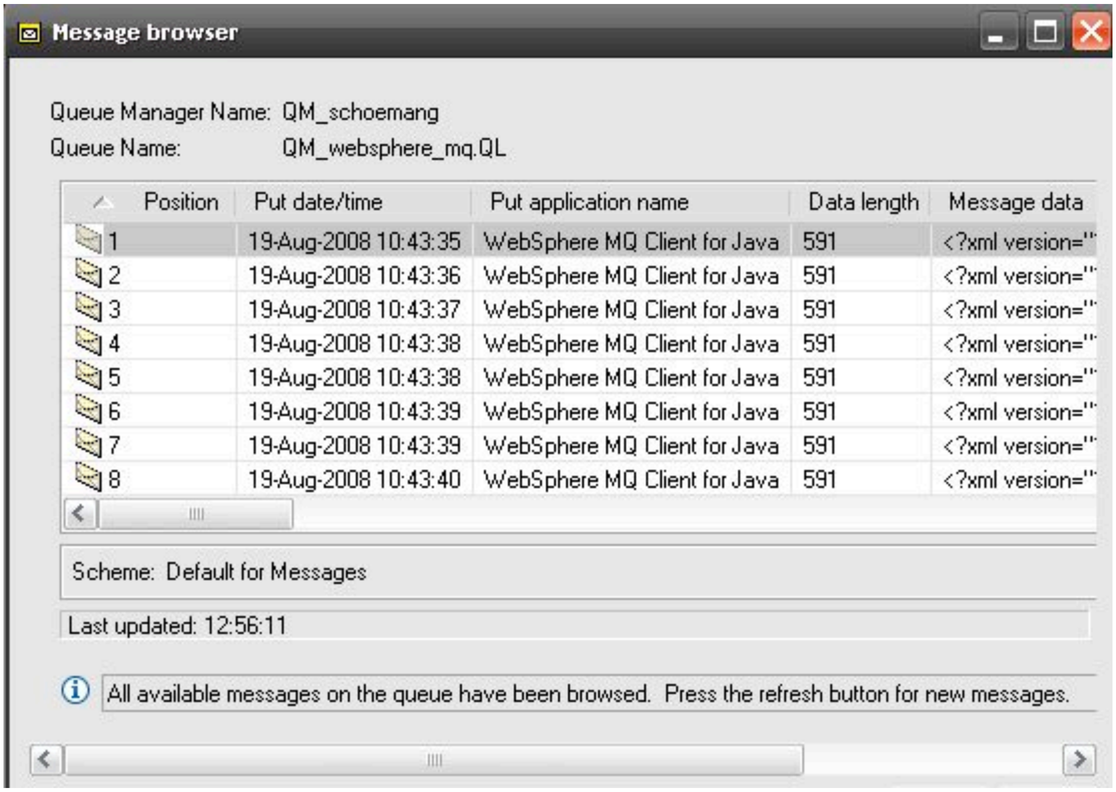
Queue name	Queue type	Definition type	Open input count	Current queue depth
clq_default_schoemang	Local	Predefined	0	0
default	Local	Predefined	0	0
postcard	Local	Predefined	0	0
QM_websphere_mq.QL	Local	Predefined	0	8

Scheme: Default for Queues - Distributed

Last updated: 10:47:03

0 errors, 0 warnings, 0 infos

Below is a detailed view of the messages received:



Queue Manager Name: QM_schoemang
Queue Name: QM_websphere_mq.QL

Position	Put date/time	Put application name	Data length	Message data
1	19-Aug-2008 10:43:35	WebSphere MQ Client for Java	591	<?xml version="
2	19-Aug-2008 10:43:36	WebSphere MQ Client for Java	591	<?xml version="
3	19-Aug-2008 10:43:37	WebSphere MQ Client for Java	591	<?xml version="
4	19-Aug-2008 10:43:38	WebSphere MQ Client for Java	591	<?xml version="
5	19-Aug-2008 10:43:38	WebSphere MQ Client for Java	591	<?xml version="
6	19-Aug-2008 10:43:39	WebSphere MQ Client for Java	591	<?xml version="
7	19-Aug-2008 10:43:39	WebSphere MQ Client for Java	591	<?xml version="
8	19-Aug-2008 10:43:40	WebSphere MQ Client for Java	591	<?xml version="

Scheme: Default for Messages

Last updated: 12:56:11

All available messages on the queue have been browsed. Press the refresh button for new messages.

5. Configuring the Gateway to read from an WebSphere MQ queue

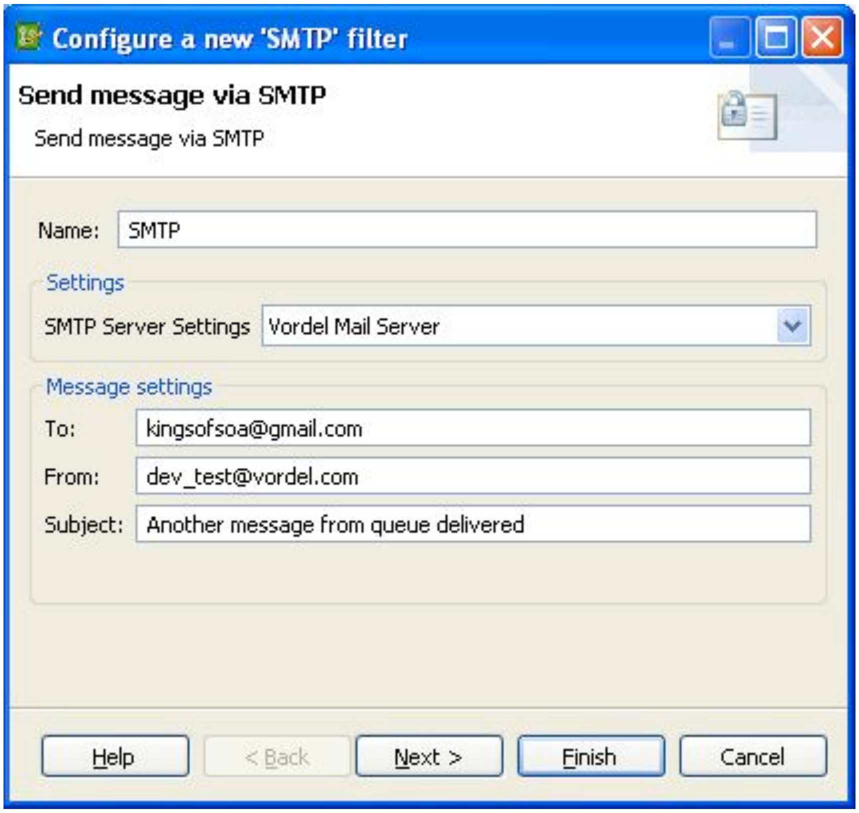
The Gateway will be configured to read the messages from "QM_websphere_mq.QL"

5.1. Create Policy that will be invoked with message read from WebSphere MQ queue

1. To create the second policy that the JMS consumer will point to:
2. In Policy Studio go to the **External Connections** panel and right click on **SMTP Servers**
3. Select **Add a SMTP Server** and configure the filter with credentials and settings of a mail server and account that can be used to send emails to. (e.g. below)
4. With the **Policies** panel selected right click on **Policies**
5. Click **Add Policy** and create a new policy titled "Read from WebSphere Queue".
6. Click on the newly created Policy.

7. Drag an **SMTP** filter from the **Routing** group located in the palette on the right of the Policy Studio.
8. Configure the filter with newly created SMTP server setting and test message settings.

SMTP Filter settings used for test:



Name: SMTP

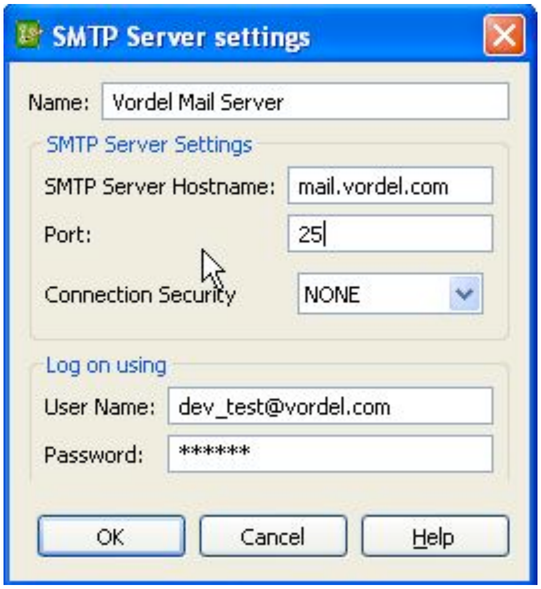
SMTP Server Settings: Vordel Mail Server

To: kingsofsoa@gmail.com

From: dev_test@vordel.com

Subject: Another message from queue delivered

SMTP Server Settings:



Name: Vordel Mail Server

SMTP Server Settings

SMTP Server Hostname: mail.vordel.com

Port: 25

Connection Security NONE

Log on using

User Name: dev_test@vordel.com

Password: ******

5.2. Creating a JMS Session:

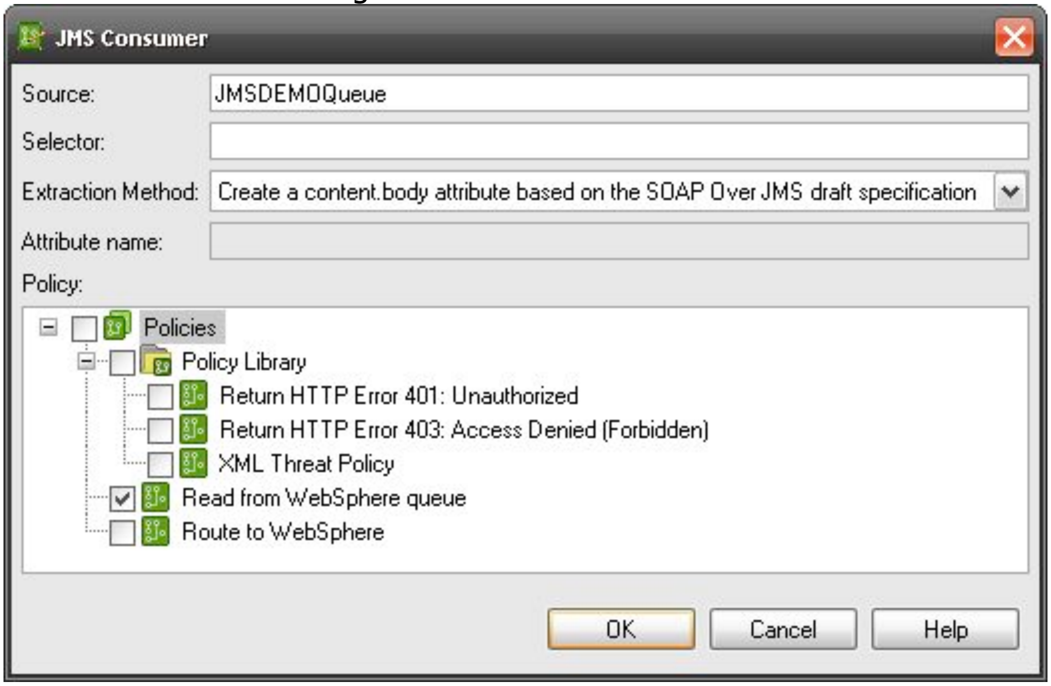
NOTE: If a JMS Session has already been created as per section 4.1, skip to number 8 below to add a JMS consumer to the existing JMS Session.

1. In Policy Studio with the **External Connections** panel selected right click on the **JMS Services** and **Add a JMS Service** and configure the following fields:
   - Name: WebSphere
   - Provider URL file:///D://JMS//JNDI:, where D:///JMS//JNDI points to the location of the .bindings file located on the Gateway machine (D:///JMS//JNDI in this case is pointing to a path on the demo machine for this guide which is d:\JMS\JNDI)
     If the .bindings file was uploaded to an Appliance then the Provider URL would be configured like this:
     Provider URL: file:///opt/OEG/OEG_product_Dir/ext/lib
   - Initial Context Factory:
     com.sun.jndi.fscontext.RefFSContextFactory
   - Connection Factory: JMSDEMOCF
   - Username: leave blank (or ask MQ service administrator)
   - Password: leave blank

2. Click **OK**
3. With the **Services** panel selected right click on the “OEG Gateway” **Process**
4. Select **Messaging System -> Add JMS Session**
5. Set JMS Service to “Websphere” just configured above.
6. Choose not to allow duplicates.
7. Click **OK**
8. Right click on the **JMS Session on service ‘Websphere’** and add a JMS Consumer and configure as follows:
   - Source: "JMSDEMOQueue" (as is configured in .bindings file created in section 2.3)
   - Extraction Method: For simplicity, select "Create a content body attribute based on the SOAP Over JMS draft specification”.
   - Point the JMS Consumer to the “Read from WebSphere Queue” policy.

PLEASE NOTE: The provider URL will differ depending on where the .bindings file is located.
It should be located in a folder of ANY name on the Appliance or host where the Gateway is running.

The JMS Consumer configuration window:



JMS Consumer

Source: JMSDEMOQueue

Selector:

Extraction Method: Create a content.body attribute based on the SOAP Over JMS draft specification

Attribute name:

Policy:

- □ Policies
  - □ Policy Library
    - □ Return HTTP Error 401: Unauthorized
    - □ Return HTTP Error 403: Access Denied (Forbidden)
    - □ XML Threat Policy
  - ☑ Read from WebSphere queue
  - □ Route to WebSphere

PLEASE NOTE: The provider URL will differ depending on where the .bindings file is located.
It should be located in a folder of ANY name on the Appliance or host where the Gateway is running.

5.3. Ensure policies are updated on the Gateway

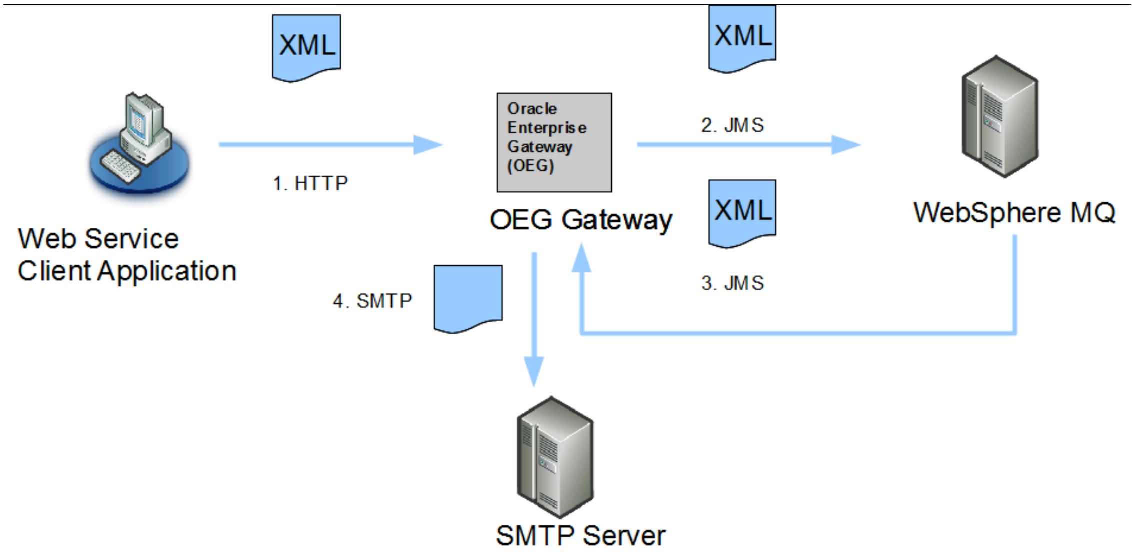
Complete the following steps to refresh the policies:

1. Open the Policy Studio.
2. Click on **Settings**.
3. Select **Deploy** to ensure that the changes made are propagated to the live Gateway. Click **Yes** to deploy changes to the server.

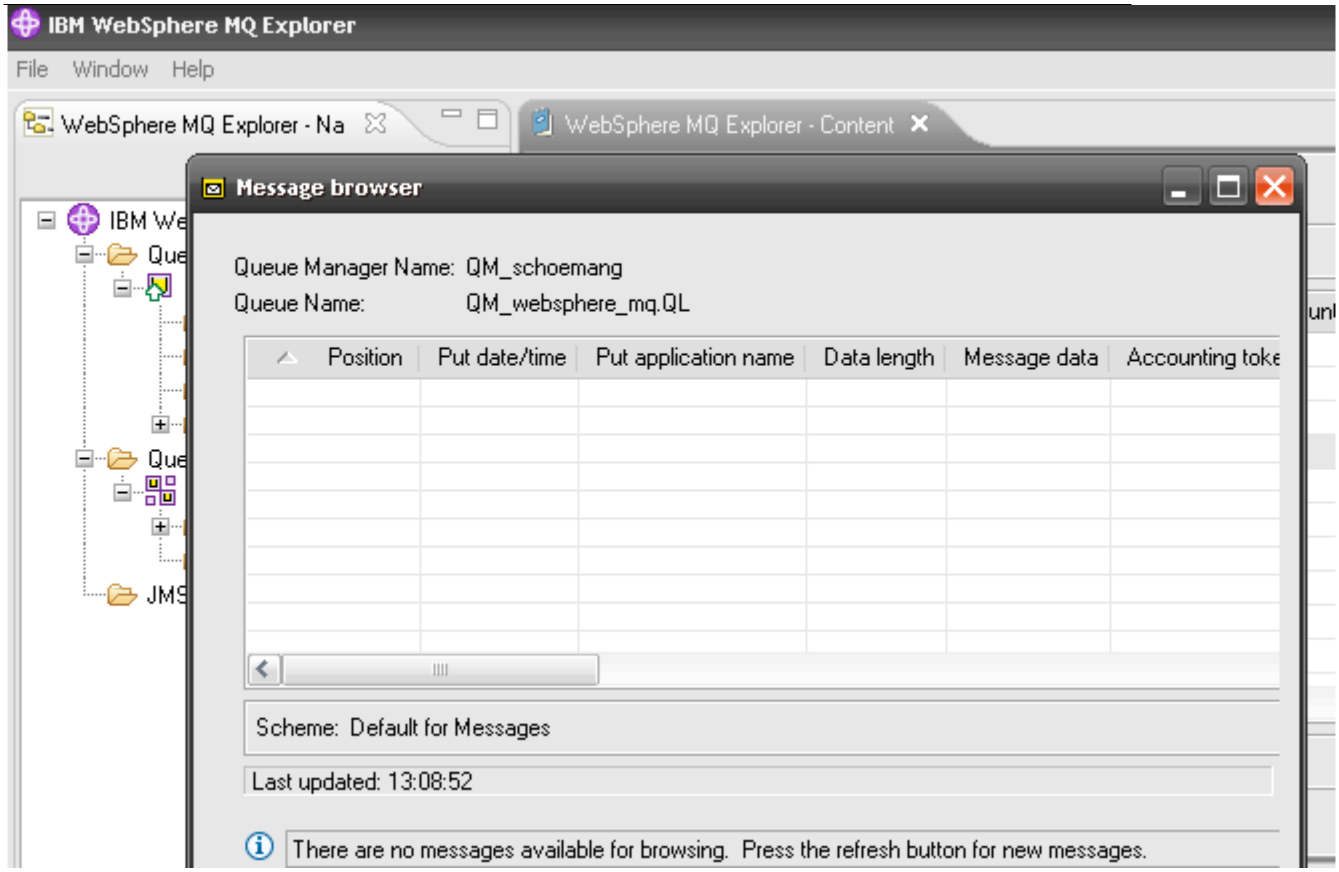
5.4. Testing to read messages from a queue

The Gateway has also been configured to let the JMS service consume the message on the queue and to forward it to a mail client via SMTP. By creating the JMS consumer and the policy that it pointed to (i.e. "Read from WebSphere Queue") that contains a SMTP filter, the messages have been read from "queue1" and sent to a mail client as configured in the SMTP filter.

The following diagram shows the flow of the message from the client through the Gateway to WebSphere MQ:



The following screenshot shows that all messages have now been "consumed" from the and forwarded via SMTP to the mail client:


IBM WebSphere MQ Explorer
File Window Help
WebSphere MQ Explorer - Na
WebSphere MQ Explorer - Content
Message browser
IBM We
Que
Que
JMS

Queue Manager Name: QM_schoemang
Queue Name: QM_websphere_mq.QL

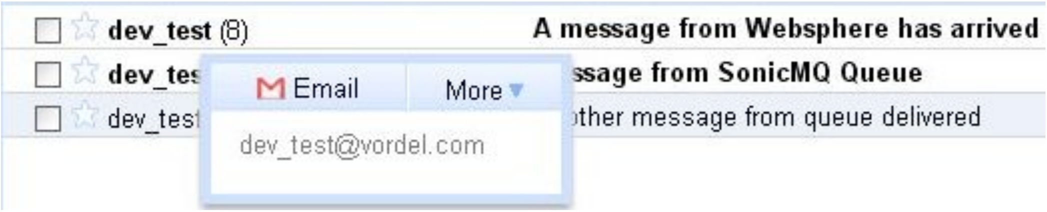
Position	Put date/time	Put application name	Data length	Message data	Accounting toke

Scheme: Default for Messages

Last updated: 13:08:52

There are no messages available for browsing. Press the refresh button for new messages.

The screenshot below shows the inbox of the email recipient that is configured in the SMPT filter that reads all messages off the queue and sends it over SMTP



□	dev_test (8)	A message from Websphere has arrived
□	dev_tes	ssage from SonicMQ Queue
□	dev_test	other message from queue delivered

6. Conclusion

This document is a simplistic demonstration on how to setup the connection from a OEG Gateway to the WebSphere MQ provider using the JMS Service and filter options in the Gateway.
This configuration can be part of a larger policy, including features such as XML threat detection and conditional routing, features which are out of the scope of this document but are covered in other documents which can be obtained from Oracle at http://www.oracle.com.

Oracle is committed to developing practices and products that help protect the environment

Oracle Enterprise Gateway
May 2011
Author:

Oracle Corporation
World Headquarters
500 Oracle Parkway
Redwood Shores, CA 94065
U.S.A.

Worldwide Inquiries:
Phone: +1.650.506.7000
Fax: +1.650.506.7200
oracle.com

SOFTWARE. HARDWARE. COMPLETE.