



An Oracle White Paper
June 2011

SonicMQ - 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	5
Prerequisites	6
Configuration Steps	6
2. Setting up the SonicMQ environment	7
Download SonicMQ	7
Installation of SonicMQ	7
Start SonicMQ and configure JMS settings	7
3. Setting up the OEG environment	9
Download OEG Gateway software	9
Load SonicMQ Provider files onto the OEG Gateway	9
Preparing the jar files for upload:	9
4. Configuring the Gateway to read from a SonicMQ Queue	13
Creating a JMS Session:	13
Create a “Route to SonicMQ” Policy	14
Ensure policies are updated on the Gateway	17
Test the configuration to place message on SonicMQ Queue.....	17
Configure SonicMQ Test Client	18
Load a sample message into OEG Service Explorer.....	18
Send Message and check SonicMQ Queue	18
5. Configuring the Gateway to place messages on SonicMQ Queue... ..	20
Create Policy that will be invoked with message read from SonicMQ	20
Creating a JMS Session:	22
Ensure policies are updated on the Gateway	23
Testing to read messages from a queue	23
6. Conclusion	25

1. Introduction

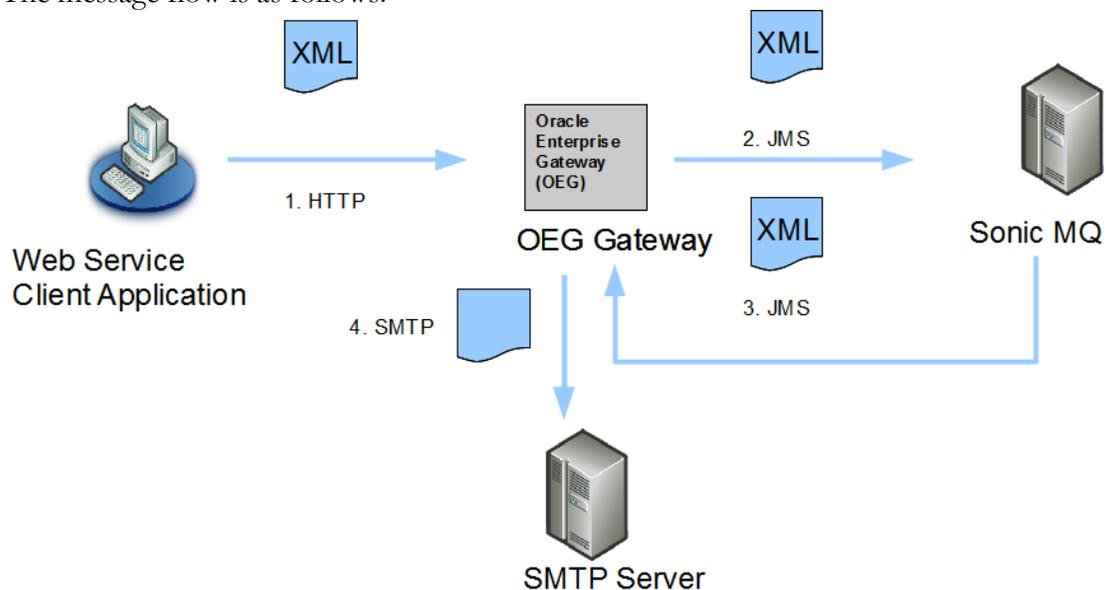
Purpose

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 Oracle Enterprise Gateway, from version 6.0 upwards.

In this guide the message queuing system that will be used is **SonicMQ**.

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.

Prerequisites

- SonicMQ available from <http://www.sonicsoftware.com>
- Oracle Enterprise Gateway Software available from www.oracle.com
- Java Virtual Machine. SonicMQ will install the appropriate JVM or use an existing installation if there is one available.

Configuration Steps

1. Download and install SonicMQ
2. Configure SonicMQ
3. Install OEG Gateway
4. Configure Gateway to send messages to SonicMQ
5. Configure Gateway to listen for messages from SonicMQ queue
6. Test Setup

2. Setting up the SonicMQ environment

Download SonicMQ

SonicMQ is available from:

http://www.sonicsoftware.com/products/sonicmq/eval_downloads/index.ssp

The most current version (at time of writing) is SonicMQ V7.6

Installation of SonicMQ

1. Once SonicMQ has been downloaded extract the files and run setup.bat
2. This will create a directory "Sonic" containing the following directories: Archives, Docs7.6, InstallLogs7.6 and MQ7.6.

Start SonicMQ and configure JMS settings

Start the SonicMQ by completing the following steps:

1. Start "SonicMQ Domain Manager"
2. Start "Management Console" and go to "Tools -> JMS Administered Objects".
 - Select "JNDI Naming Service" radio button and check "Sonic Storage" checkbox.
 - Leave other input fields empty and hit "Connect" button. A new object should appear under "JMS Object Stores" on the left. Click on the new object.
 - A new dialog box will appear with two tabs, "Destinations" and "Connection Factories".
 - Create a new destination clicking on "Destinations" Tab, and click a "New" button (located at the bottom of the window, may be not visible if window is not expanded to the full screen).
3. Enter new "Lookup name", "Type" and "Destination Name".
 - Any Lookup name could be used. The Destination name should be the same as the name of the existing queue/topic in SonicMQ.
4. For this guide "SampleQ1", "Queue", "SampleQ1" is as the Destination values.
5. Click on "Connection Factories" tab, and hit the "New" button.
6. Enter new "Lookup Name" (For this guide 'OEGConnectionFactory' is used).
7. Enter "Connection URL(s)" (Use 'localhost:2506')
8. Hit "Update" button to save the entry.

3. Setting up the OEG environment

Download OEG Gateway software

Contact your Oracle Representative for more information if needed.

Load SonicMQ Provider files onto the OEG Gateway

SonicMQ provides a particular JMS provider that the Gateway will use to connect to SonicMQ. The JMS provider takes the form of Java archive files (i.e. JAR files). Once SonicMQ is installed it is a simple matter to drop the JMS provider JAR files onto the OEG Gateway.

Instructions for OEG:

Preparing the jar files for upload:

The first thing to do is to prepare the jar files that the Gateway requires. As there might be more than one jar file, the easiest way is to add all the JAR files to a tar file before uploading it to the appliance.

The required JAR files are stored in the lib directory of the Sonic installation directory, e.g. /Sonic/lib

Add all the following jar files to a tar archive file:

- sonic_Client.jar
- sonic_XA.jar
- mfcontext.jar

For Windows: Open an application capable of creating tar files. Select the files above and archive to a tar file.

For Linux: Add the necessary jar files to a tar file by using the tar command. Browse to the directory where the jar files reside and run the following command:

```
>tar -cvf jar.tar *.jar
```

The next step is to upload the archive to the appliance. This can be done using either the web interface or the command line.

iUpload via the web interface:

1. Log on to the Web Administrator Interface via a browser at the following URL: https://IP_OF_Appliance:10000

2. Login using the user credentials assigned. The default Username is “admin” with a default password of “changeme”.
3. Click on the Upload and Download option on the top menu options.
4. Click on the Browse button and browse to the tar file that contains all the jar files.
5. Upload the file to the /opt/OEG/OEG_product_dir/ext/lib directory.
6. For ease of use it is recommended to select “Yes” when prompted and then click the Delete button next to the Extract Tar files option.
7. Click on Upload. This will upload the tar file, extract it, and then delete the tar file from the appliance.
8. If this has succeeded, a confirmation screen will be displayed showing the list of files that were uploaded.

The following screenshot shows the Upload and Download screen on the Web Administrator Interface of the Appliance:

The screenshot displays the Vordel Web Administrator Interface. At the top, there is a navigation bar with the Vordel logo and several menu items: System, Administration Interface, Servers, and a sub-menu containing Bootup and Shutdown, Network Configuration, RAID Status, System Logs, System Time, and Upload and Download. The main content area is titled "Upload files to server" and is divided into two sections: "Upload files to server" and "Download file from server to PC".

In the "Upload files to server" section, there are several fields and buttons:

- Files to upload:** A text input field containing "C:\SonicMQ.Jars.tar" with a "Browse..." button next to it. Below it, there are two more empty text input fields, each with a "Browse..." button.
- File or directory to upload to:** A text input field containing "/opt/vordel/vordelxmlgateway-vxp-5.0.2/ext/lib" with a "..." button next to it. To the right of this field is a checkbox labeled "Create directory if needed?".
- Owned by user:** A text input field containing "root" with a "..." button next to it.
- Owned by group:** A radio button labeled "Default" is selected. To its right is an empty text input field with a "..." button.
- Extract TAR files?:** Three radio buttons are present: "Yes, then delete" (unselected), "Yes" (unselected), and "No" (selected).
- Buttons:** An "Upload" button is located below the "Extract TAR files?" section.

In the "Download file from server to PC" section, there are:

- File to download:** An empty text input field with a "..." button next to it.
- Show in browser if possible?:** Two radio buttons: "Yes" (unselected) and "No" (selected).
- Buttons:** A "Download" button is located below the "Show in browser if possible?" section.

The confirmation screen showing the list of uploaded files appears as follows:

The screenshot displays the Vordel Web Administrator Interface, showing the confirmation screen for the upload process. The interface includes the Vordel logo and the same navigation bar as the previous screenshot. The main content area is titled "Successfully uploaded the following files:" and contains the following text:

```

/opt/vordel/vordelxmlgateway-vxp-5.0.2/ext/lib/SonicMQ Jars.tar (1587200 bytes), and extracted files ..
mfcontext.jar
sonic_client.jar
sonic_XA.jar

```

Alternatively upload the jar files via command line:

Note: The directory where jar files should be placed on the appliance is as follows:

/opt/OEG/OEG_product_dir/ext/lib (where OEG_product_dir is the version of the gateway installed)

On Unix/Linux:

Run SCP using the following command syntax to perform the upload:

1. Browse to directory where the tar file was created.
2. At the prompt type: scp [options] [[user@]host1:]filename1 ...
[[user@]host2:]filename2

Example: > scp jar.tar root@IP_OF_APPLIANCE:
/opt/OEG/OEGgateway/ext/lib

3. You will be prompted for the user (root) password before the transfer begins.

For more information on how to use the scp command type **scp -help** or **man scp** at the prompt.

Extracting TAR file manually:

1. Use SSH to connect to the appliance securely by typing the following command from the prompt: > ssh root@IP_OF_APPLIANCE

Example: > ssh root@192.168.2.300

2. You will be prompted for the user (root) password to complete the connection.
3. Browse to /opt/OEG/OEG_product_dir/ext/lib to extract the file from the tar file with the following command: > tar -xvzf myfile.tar

Example: > tar -xvzf jar.tar

On Windows:

Use PSCP to copy files to the Appliance:

1. PSCP can be used to transfer files securely from Windows to the Appliance. It can be downloaded from <http://www.chiark.greenend.org.uk/~sgtatham/putty/download.html>

2. Open a Windows command prompt and run pscp using the following syntax: > pscp filename username@ipadress:where to place on remote

Example: > pscp jar.tar root@192.168.2.300:
/opt/OEG/OEGgateway/ext/lib

Extracting the TAR file manually:

Use SSH to connect to the appliance securely using a program like Putty.

Browse to /opt/OEG/OEG_product_dir/ext/lib

Extract the tar file with the following command: `tar -xvzf myfile.tar`

Example: `> tar -xvzf jar.tar`

Instructions for Software install:

Copy the following jar files from the `/Sonic_install_dir/lib` directory to the `OEG_product_dir/ext/lib` directory

- `sonic_Client.jar`
- `sonic_XA.jar`
- `mfcontext.jar`

4. Configuring the Gateway to read from a SonicMQ Queue

The gateway will be configured to place messages it receives on a queue (i.e. destination) named “SampleQ1” in SonicMQ.

Creating a JMS Session:

1. In Policy Studio **External Connections** navigation panel, right click on **JMS Services** and **Add a JMS Service** then configure the following fields:
 - ▲ **Name:** SonicMQ
 - ▲ **Provider URL:** IP_OF_SonicMQ:2506 (where IP_OF_SonicMQ points to the machine on which SonicMQ has been installed)
 - ▲ **Initial Context Factory:**
com.sonicsw.jndi.mfcontext.MFContextFactory
 - ▲ **Connection Factory:** OEGConnectionFactory
 - ▲ **Username:** admin
 - ▲ **Password:** admin (or whatever was configured during installation)
2. Click on Ok to save the JMS service before proceeding with the Custom Message Properties
3. Right click on the SonicMQ service and click on Edit then click Add to configure each of the following Custom Message Properties:
 - **Name:** com.sonicsw.jndi.mfcontext.domain
Value: Domain1
 - **Name:** java.naming.factory.initial
Value: com.sonicsw.jndi.mfcontext.MFContextFactory
 - **Name:** java.naming.provider.url
Value: IP_OF_SonicMQ:2506 (where IP_OF_SonicMQ points to the machine on which SonicMQ has been installed)
4. Click on the **Services** navigation panel. Right click on the **OEG Gateway** process and in **Messaging System** choose to **Add JMS Session**.
5. Select **SonicMQ** and not to allow duplicates then OK.

Create a “Route to SonicMQ” Policy

Create a small test policy to route messages on to the SonicMQ Queue by completing the following steps:

- Open Policy Studio and the **Policies** navigation panel. Create a new Policy titled “Route to SonicMQ” by right clicking on Policies and select **Add Policy**.
- Create a new relative path on the Gateway Process called “/ToSonicMQ” by going to **Services** navigation panel and expanding Processes, then expand OEG Gateway and right click on Default Services to add the new path
- Map the /ToSonicMQ path to the policy called “Route to SonicMQ”. This means that when a message is received by the Gateway on the path “/ToSonicMQ”, it will be passed to the “Route to SonicMQ” policy, which will then process the message.

Configuring the Messaging System Filter

When a policy that routes to a JMS provider (such as SonicMQ) 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:

1. Go back to the “Route to SonicMQ” policy canvas. Drag a Messaging System from the Filter Palette on the right onto the policy field.
2. Under the Request tab select the JMS Service that has been configured above (titled “SonicMQ”) from the JMS Session dropdown.
3. Set the Destination to “SampleQ1”, which was configured during the SonicMQ destination configuration.
4. The Message Type should be specified. For example, in "Message Type", select "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 Finish

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.

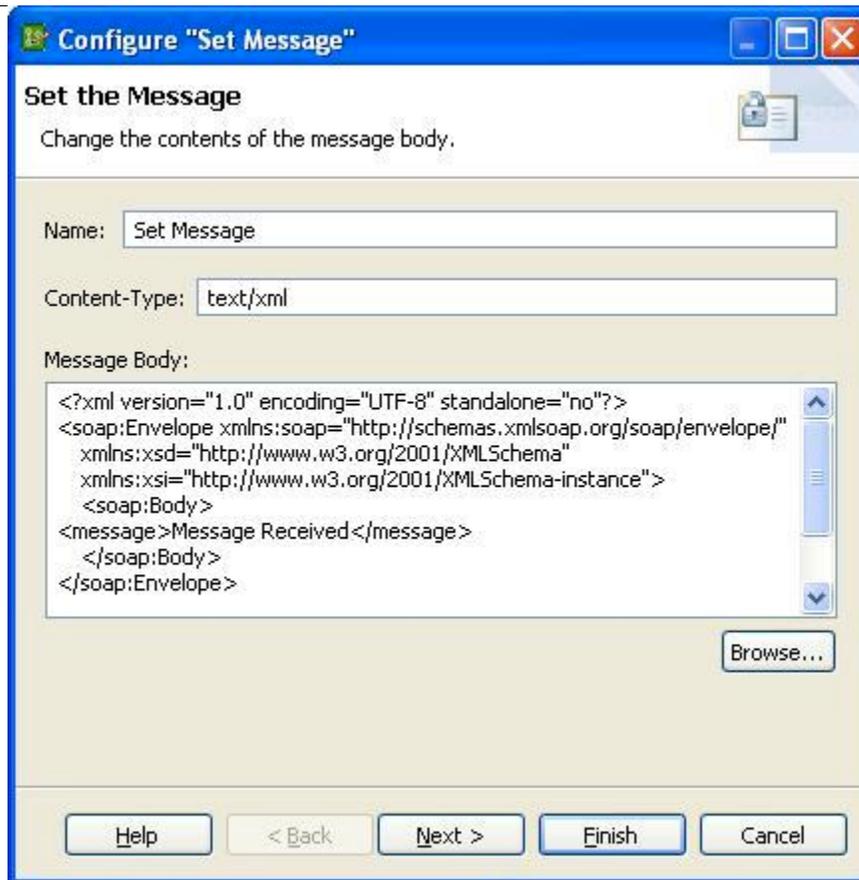
- Create a `TextMessage` from the attribute named below:

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:

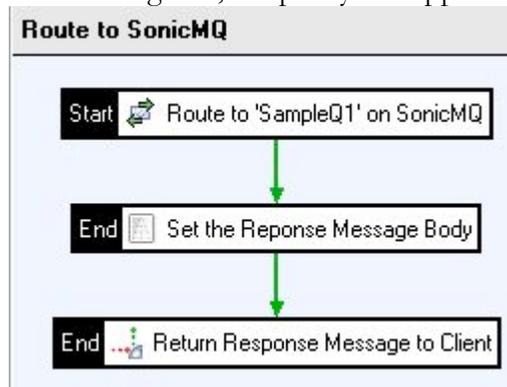
1. Messaging System Filter: This filter should be configured as described above. This is a mandatory filter in the policy.
2. Set Message Filter: Used to set the content of an XML response message that can be returned to the client to acknowledge that the message has been placed on the Open JMS queue. This step is not mandatory, but is useful for informational purposes. This filter can be found in the Conversion category of filters.

How the Set Message filter has been configured:



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.

Once configured, the policy will appear as follows:



Having configured the JMS Session and the “Route to SonicMQ” policy, we will be able to place messages on the SonicMQ “SampleQ1” queue.

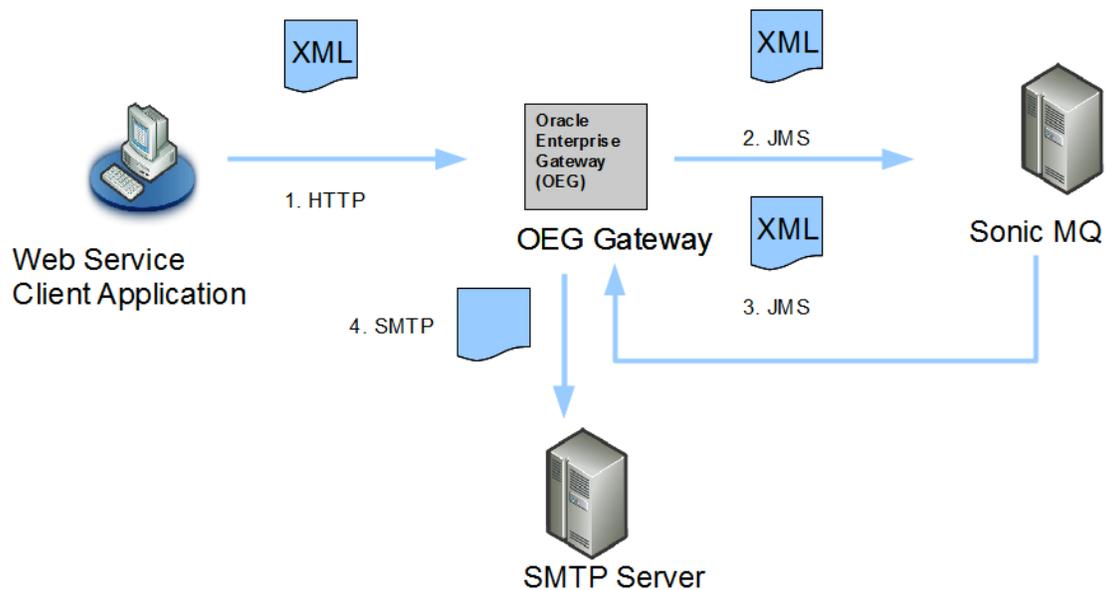
Ensure policies are updated on the Gateway

- Open the Policy Studio.
- Click on Settings.
- Select “Deploy” to ensure that the changes made are propagated to the live Gateway.

Test the configuration to place message on SonicMQ 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 SonicMQ queue.

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



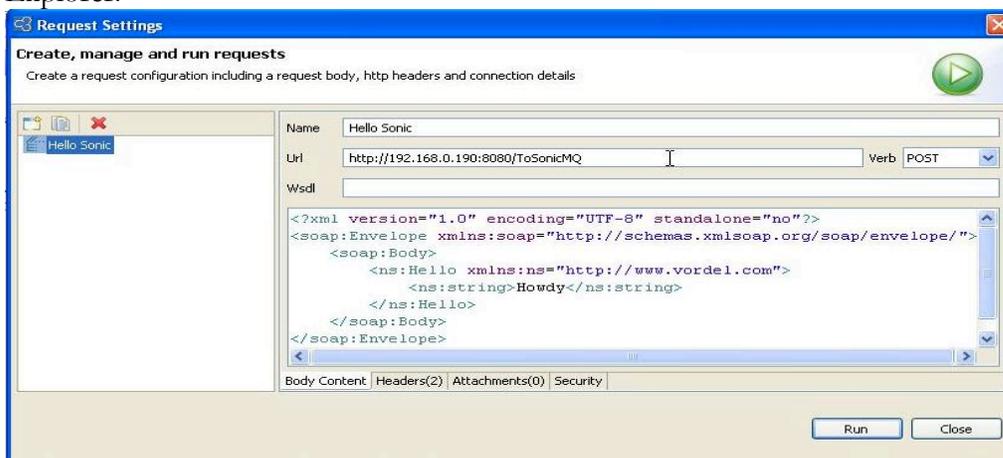
Configure SonicMQ Test Client

1. Start JMS Test Client using the Start menu or from Sonic Management Console Tools -> JMS Test Client menu.
2. Create new connection: Enter 'localhost' for "Broker Host" and anything (like '123') for "Connect ID". Hit "Connect" button.
3. Create new session: Click on the localhost: in on the left. Choose 'Queue' for a "Type" and some string for "Name". Click "Create" button.
4. Create receiver: Click on "Receivers" on the left. Enter 'SampleQ1' for the "Queue" and hit "Create" button.
5. Inspect queue: Click on the "SampleQ1" on the left, you will see the messages sent to 'SampleQ1' queue here.

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 "SonicMQ" path on the Gateway.

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



Send Message and check SonicMQ Queue

By sending messages using OEG Service Explorer, the Gateway will route the messages to the SonicMQ Queue, in this case "Sample1" as configured.

Once test messages have been sent, open the SonicMQ Management Console. It should show a total number of messages that have been delivered to "queue1".

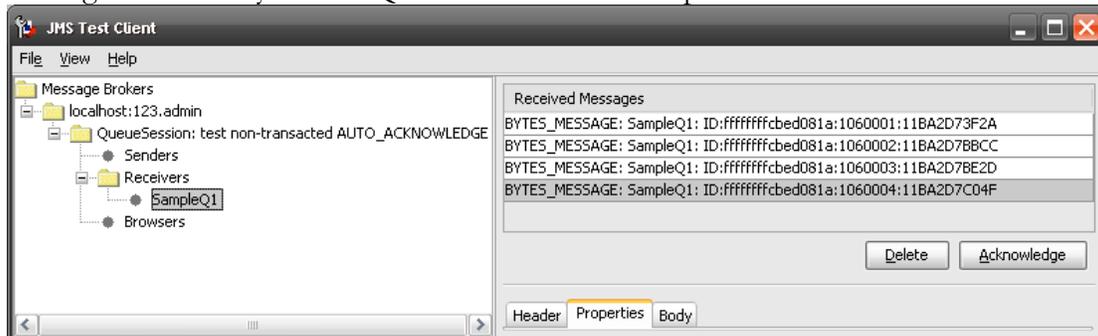
Having sent the SOAP request, the response will be displayed in the SOAP Response panel, as displayed in the screenshot below:

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <soap:Body>
  <message>Message Received</message>
  </soap:Body>
</soap:Envelope>
```

Once test messages have been sent, open the SonicMQ Management Console.

1. Start JMS Test Client using the Start menu or from Sonic Management Console Tools -> JMS Test Client menu.
2. Click on the "SampleQ1" on the left, you will see the messages sent to 'SampleQ1' queue here.

Messages received by SonicMQ from OEG Service Explorer:



Four messages have so far been received in this case

5. Configuring the Gateway to place messages on SonicMQ Queue

The Gateway will be configured to read the messages from “SampleQ1”

Create Policy that will be invoked with message read from SonicMQ

- To create the second policy that the JMS consumer will point to:
- In Policy Studio go to the **External Connections** panel and right click on **SMTP Servers**
- 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)
- With the **Policies** panel selected right click on **Policies**
- Click **Add Policy** and create a new policy titled “Read from SonicMQ Queue”.
- Click on the newly created Policy.
- Drag an **SMTP** filter from the **Routing** group located in the palette on the right of the Policy Studio.
- Configure the filter with newly created SMTP server setting and test message settings.

SMTP Filter settings used for test:

Configure a new 'SMTP' filter

Send message via SMTP

Name: SMTP

Settings

SMTP Server Settings: Vordel Mail Server

Message settings

To: kingsofsoa@gmail.com

From: dev_test@vordel.com

Subject: Another message from queue delivered

Buttons: Help, < Back, Next >, Finish, Cancel

SMTP Server Settings:

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: *****

Buttons: OK, Cancel, Help

 Creating a JMS Session:

NOTE: If a JMS Session has already been created as per section 4.1, skip to number 4 below to add a JMS consumer to the existing JMS Session.

1. In **Policy Studio External Connections** navigation panel, right click on **JMS Services** and **Add a JMS Service** then configure the following fields
Click on Ok to save the JMS service before proceeding with the Custom Message Properties
 - ▲ **Name:** SonicMQ
 - ▲ **Provider URL:** IP_OF_SonicMQ:2506 (where IP_OF_SonicMQ points to the machine on which SonicMQ has been installed)
 - ▲ **Initial Context Factory:**
com.sonicsw.jndi.mfcontext.MFContextFactory
 - ▲ **Connection Factory:** OEGConnectionFactory
 - ▲ **Username:** admin
 - ▲ **Password:** admin (or whatever was configured during installation)
2. Click on Ok to save the JMS service before proceeding with the Custom Message Properties
3. The following Custom Message Properties should be configured:
 - ▲ **Name:** com.sonicsw.jndi.mfcontext.domain
Value: Domain1
 - ▲ **Name:** java.naming.factory.initial
Value: com.sonicsw.jndi.mfcontext.MFContextFactory
 - ▲ **Name:** java.naming.provider.url
Value: IP_OF_SonicMQ:2506 (where IP_OF_SonicMQ points to the machine on which SonicMQ has been installed)
4. Right click on the JMS Service and add a JMS Session. Select not to allow duplicates.
5. Right click on the JMS Session and add a JMS Consumer and configure as follows:
 - ▲ **Destination:** "SampleQ1"
 - ▲ **Extraction Method:** For simplicity, select "Create a content.body attribute based on SOAP Over JMS draft specification"
 - ▲ Point **the** JMS Consumer to the "Read from SonicMQ Queue" policy.

Ensure policies are updated on the Gateway

Complete the following steps to refresh the policies:

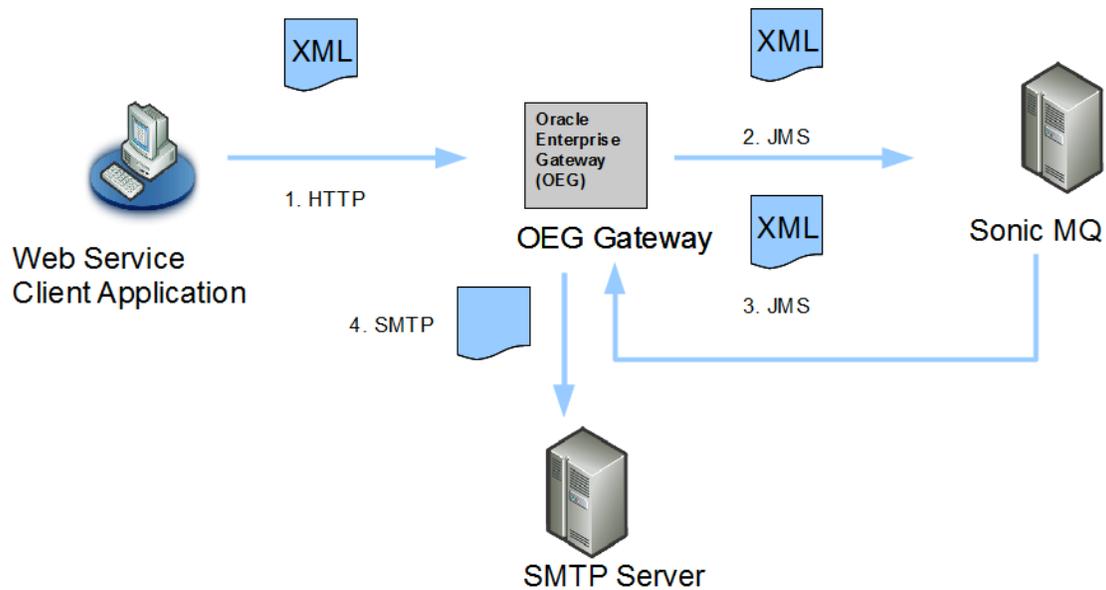
1. Open the Policy.
2. Studio Click on Settings.

3. Select “Refresh Server” to ensure that the changes made are propagated to the live Gateway.

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 SonicMQ Queue”) that contains a SMTP filter, the messages have been read from “SampleQ1” and sent to a mail client as configured in the SMTP filter.

The flow demonstrates messages read from SonicMQ, passing back through the Gateway to the SMTP client:



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

Archive **Report Spam** **Delete** **More Actions** [Refresh](#)

Select: All, None, Read, Unread, Starred, Unstarred

<input type="checkbox"/> ☆ dev_test (4)	Message from SonicMQ Queue
<input type="checkbox"/> ☆ dev	Another message from queue delivered

 **Email** **More** ▾

dev_test@vordel.com

6. Conclusion

This document is a simplistic demonstration on how to setup the connection from a Oracle Enterprise Gateway to the SonicMQ provider using the JMS Service and filter options in the Gateway.

This connection can be part of a larger policy, including features such as XML threat detection and conditional routing, features which are out of scope in this document but are covered in other documents which can be obtained from the Oracle site.

Please direct any support questions to <http://support.oracle.com>



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



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

Copyright © 2011, Oracle and/or its affiliates. All rights reserved. This document is provided for information purposes only and the contents hereof are subject to change without notice. This document is not warranted to be error-free, nor subject to any other warranties or conditions, whether expressed orally or implied in law, including implied warranties and conditions of merchantability or fitness for a particular purpose. We specifically disclaim any liability with respect to this document and no contractual obligations are formed either directly or indirectly by this document. This document may not be reproduced or transmitted in any form or by any means, electronic or mechanical, for any purpose, without our prior written permission.

Oracle and Java are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

AMD, Opteron, the AMD logo, and the AMD Opteron logo are trademarks or registered trademarks of Advanced Micro Devices. Intel and Intel Xeon are trademarks or registered trademarks of Intel Corporation. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. UNIX is a registered trademark licensed through X/Open Company, Ltd. 0410

SOFTWARE. HARDWARE. COMPLETE.