Migrating from OWSM 10gR3 Gateway to Oracle Enterprise Gateway 11.1.1

1. Introduction

This document describes how to migrate OWSM 10gR3 Gateway configuration to Oracle Enterprise Gateway 11.1 configuration.

Mapping of OWSM to Oracle Enterprise Gateway filters

This section shows the mapping from steps available in OWSM to the equivalent filter / policy settings in the Oracle Enterprise Gateway.

Step: Active Directory Authenticate

OWSM 10gR3 contains a step called “Active Directory Authenticate” where by credentials are authenticate against Active Directory, the default settings for this is as follows:

<table>
<thead>
<tr>
<th>Active Directory Authenticate</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Basic Properties</strong></td>
</tr>
<tr>
<td>Enabled (*)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Authentication Properties</th>
<th>Type</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>AD host (*)</td>
<td>string</td>
<td>localhost</td>
</tr>
<tr>
<td>AD port (*)</td>
<td>int</td>
<td>389</td>
</tr>
<tr>
<td>AD SSL port</td>
<td>int</td>
<td>636</td>
</tr>
<tr>
<td>AD baseDN (*)</td>
<td>string</td>
<td>cn=users,dc=oracle,dc=com</td>
</tr>
<tr>
<td>AD domain (*)</td>
<td>string</td>
<td>oracle.com</td>
</tr>
<tr>
<td>ADSSLEnabled (*)</td>
<td>boolean</td>
<td>false</td>
</tr>
<tr>
<td>Uid Attribute (*)</td>
<td>string</td>
<td>samAccountName</td>
</tr>
<tr>
<td>User Attributes to be retrieved</td>
<td>string[]</td>
<td></td>
</tr>
</tbody>
</table>

Fault Code


Oracle Enterprise Gateway provides the same functionality by following these steps:

- External connections > LDAP Connections > Add a LDAP Connection

- **LDAP Connections**
  - A pool of LDAP connection details
    - Add a LDAP Connection

Child Items
Enter configuration details for the LDAP connection

- **Name:** Active Directory connection details
- **URL:** ldap://127.0.0.1:389
- **Cache Timeout:** 300000
- **Cache Size:** 8
- **Authenticate LDAP Requests:**
  - **Type:** Simple
  - **User Name:** cn=users,dc=oracle,dc=com
  - **Password:** **********
  - **Realm:**
  - **SSL Enabled:**

**Additional JNDI properties:**

<table>
<thead>
<tr>
<th>Name</th>
<th>Value</th>
</tr>
</thead>
</table>

**Buttons:**

- Test Connection
- Add
- Edit
- Delete
- OK
- Cancel
- Help
- Create an LDAP authentication repository which then can be used in HTTP Basic, Digest, and WS-Security username password filters for authentication.
- External connections > Authentication Repository Profiles > LDAP Repositories > Add a new Repository

- Configure the repository as

![Authentication Repository](image)

- In a filter that requires authentication against Active Directory, simply select the repository named above.

**Step: Active Directory Authorize**

Authorizes request by retrieving roles from Active Directory and checking against roles allowed by
service.

**OWSM Configuration:**

<table>
<thead>
<tr>
<th>Active Directory Autorize*</th>
<th>Environment Properties</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Basic Properties</strong></td>
<td><strong>Type</strong></td>
</tr>
<tr>
<td>Enabled (*)</td>
<td>boolean</td>
</tr>
<tr>
<td><strong>Authorization Properties</strong></td>
<td><strong>Type</strong></td>
</tr>
<tr>
<td>AD host (*)</td>
<td>string</td>
</tr>
<tr>
<td>AD port (*)</td>
<td>int</td>
</tr>
<tr>
<td>AD SSL port</td>
<td>int</td>
</tr>
<tr>
<td>AD baseDN (*)</td>
<td>string</td>
</tr>
<tr>
<td>ServiceRoles</td>
<td>string[]</td>
</tr>
<tr>
<td>ADAdminUser (*)</td>
<td>string</td>
</tr>
<tr>
<td>ADAdminPwd</td>
<td>string</td>
</tr>
<tr>
<td>AD domain (*)</td>
<td>string</td>
</tr>
<tr>
<td>ADSSLEnabled (*)</td>
<td>boolean</td>
</tr>
<tr>
<td>Uid Attribute (*)</td>
<td>string</td>
</tr>
</tbody>
</table>

**Oracle Enterprise Gateway Configuration:**

1. Use LDAP connection and repository setup as detailed in **Active Directory Authenticate** above.
2. After successful Authentication with HTTP Basic, Digest or WS-Security username password filters send circuit to a "Retrieve Attribute from Directory Server" filter followed by "Validate Message Attribute" filter. This combination can be used to check a group for a user’s membership.
3. Repeat for each group that needs checking e.g.
4. Example of a Retrieve from Directory Server for group “testgroup1”

Configure retrieval of attributes from an LDAP directory.

**Retrieve Attributes from Directory Server**

**Name:** Retrieve from Directory Server (testgroup1)

**LDAP Directory:** Active Directory connection details

**Retrieve unique user identity**

- From message attribute: authentication.subject.id
  - Configure Directory Search

**Retrieve Attributes:**

- **Base Criteria:** cn=testgroup1, cn=Users, dc=oracle, dc=com
- **Search Filter:** member=cn=[authentication.subject.id], cn=Users, dc=oracle, dc=com
- **Search Scope:** Object Level, One Level, Sub-Tree

**Attribute Name**

- cn
- distinguishedName
5. Example of a **Validate Message Attribute** for group “testgroup1”
This filter returns success if the user is found to be a a member of testgroup1. The authorized user is forwarded to the next group check and finally to the user authorized policy which is shown as a shortcut in the circuit above.

6. Similarly the user not authorized branch is a shortcut. This could be to a policy that eventually reflects an appropriate failure message back to the client.

**Step: Decrypt and Verify Signature**
XML Decryption And Signature Verification

**OWSM Configuration:**
Oracle Enterprise Gateway Configuration:
To decrypt and verify a request in the Gateway perform the XML Decryption and Verify Signature steps in one policy. Please refer to the XML Decryption and Verify Signature the sections for configuration of the relevant options.

NOTE: Ensure that the relevant certificate/key is loaded into the Gateway Certificate store and that the correct certificate is selected under Decryption and Signature Verification filters.

A sample policy where the request is decrypted and the signature verified:
Step: Extract Credentials
Locates and extracts credentials and presents the credentials in a form that they can be authenticated. You must know from where the credentials are to be extracted.

OWSM Configuration:

<table>
<thead>
<tr>
<th>Extract Credentials Properties Type</th>
<th>Default</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Credentials location (*)</td>
<td>string</td>
<td>HTTP</td>
</tr>
<tr>
<td>Namespaces</td>
<td>string</td>
<td></td>
</tr>
<tr>
<td>UserID xpath</td>
<td>string</td>
<td></td>
</tr>
<tr>
<td>Password xpath</td>
<td>string</td>
<td></td>
</tr>
</tbody>
</table>

**Credential location** allowed values are HTTP, JMS, WS-BASIC or XPath

**Oracle Enterprise Gateway Configuration:**

**HTTP Basic** - filter to check against a previously created repository name. e.g. LDAP, Active Directory, Oracle Access Manager
WS-BASIC - WS-Security UsernameToken will be stored in the wss.usernameToken message attribute.

XPath e.g. retrieve WS-Security username and password then store in message attributes authentication.subject.id and authentication.subject.password
Within the policy drag a “Retrieve from message” filter from the Attributes category.
Then do the same for wsse:Password

**File Authenticate**
**Step Name:** Authenticate username and password against a local .htpasswd file. This step depends on Extract Credentials Step

**OWSM Configuration:**
**Oracle Enterprise Gateway Configuration:**

Oracle Enterprise Gateway is not able to use a htpasswd format file like OWSM 10gR3 but the Local User Store would offer equivalent functionality and more. User credentials are securely stored in the Local User Store for authentication with a filter for example HTTP Basic.

1. To add a user to the Local User Store
2. Then use an authentication filter against the users in the store. e.g A circuit to use HTTP Basic authentication.

3. The Attributes Authentication filter can be used to authenticate any message attributes in the Local User Store.
Step: File Authorize
Authorize remote user against a local roles file. This step depends on Extract Credentials Step

OWSM Configuration:

<table>
<thead>
<tr>
<th>File Authorize</th>
<th>Environment Properties</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Basic Properties</strong></td>
<td><strong>Type</strong></td>
</tr>
<tr>
<td>Enabled (*)</td>
<td>boolean</td>
</tr>
<tr>
<td><strong>Authorization Properties</strong></td>
<td><strong>Type</strong></td>
</tr>
<tr>
<td>User roles file location (*)</td>
<td>string</td>
</tr>
<tr>
<td>Allowed roles</td>
<td>string[]</td>
</tr>
</tbody>
</table>

Oracle Enterprise Gateway Configuration:
Similar to File Authentication step the Fusion Gateway can implement role based Authorization via the Gateways own Local User Store. Authorization can be added to the previous step after successfully authenticating the user via the HTTP Basic filter.

1. Add a comma separated list of roles to the user attributes of a Local Store user. This is very similar to the roles list used in the OWSM authorization file.
2. After **HTTP Basic** authentication add a "**Retrieve from User Store**" filter to get the user's roles. Note that the message attribute `authentication.subject.id` will contain the username after successfully running HTTP Basic filter.
3. Next test if the user has a particular role on their roles list with a **Validate Message Attribute** filter. Click **Add** to enter a regular expression to test if a particular role like administrators is on the list. Note the Retrieve from user store copies the message attributes to user.&lt;user attribute name&gt; (so user.roles in this case)
4. If success then so far the forward to the path for administrators.
5. Add more checks for other roles that require testing.
6. An example circuit might look like the following.
**Step: Handle Generic Fault**

Example generic fault handler step

**OWSM Configuration:**

<table>
<thead>
<tr>
<th>Basic Properties</th>
<th>Type</th>
<th>Default</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Enabled (*)</td>
<td>boolean</td>
<td>true</td>
<td>true</td>
</tr>
<tr>
<td>CustomMessage (*)</td>
<td>string</td>
<td>A fault was thrown in the step</td>
<td>A fault was thrown in the step</td>
</tr>
</tbody>
</table>

Provides custom message in the SOAP fault when errors are encountered.

**Oracle Enterprise Gateway Configuration:**

**SOAP Fault Filter**

1. By default, the Gateway returns a very basic SOAP Fault to the client when a message filter has failed.
2. **SOAP Fault** processor can be added to a policy to return more complicated error information to the client.
3. Using the **SOAP Fault** processor, administrators have the flexibility to configure just how much information to return to clients, depending on their individual requirements.
4. The SOAP Fault filter can either be returned in a circuit path or it can be set as the fault handler for a particular policy. In this case it will be invoked when a filter aborts.

Customized SOAP Faults

1. It is possible to produce a customized SOAP fault to return to the client. The Set Message filter can change the contents of the message body to any arbitrary content. When an exception occurs in a policy, it is possible to use this filter to customize the body of the SOAP fault. e.g.

2. Create a Fault Policy

Create a policy called "Fault Circuit". This policy will use the Set Message filter to customize the body of the SOAP fault. When configuring this filter, enter the contents of the customized SOAP fault that you want to return to clients in the text area provided.

The second Reflect filter is used to return the SOAP Fault back to the client.

3. Drop a Policy Shortcut onto the policy canvas.

4. Point the shortcut to the newly created Fault Policy

5. Right click on the Policy Shortcut and set it as the fault handler
Step: Insert Oracle Access Manager Token
Inserts an ObSSOCookie in the SOAP security header.

**OWSM Configuration:**

<table>
<thead>
<tr>
<th>Insert Oracle Access Manager Token</th>
<th>Environment Properties</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Basic Properties</strong></td>
<td><strong>Type</strong></td>
</tr>
<tr>
<td>Enabled (*)</td>
<td>boolean</td>
</tr>
</tbody>
</table>

**Oracle Enterprise Gateway Configuration:**
The equivalent functionality to ObSSOCookie can be achieved with the Fusion Gateway by passing the SSO token that was inserted into an HTTP header during the Oracle Access Manager Authorization phase. An incoming policy can be setup to check for the sso token header and call the SSO token validate filter rather than authenticating the user again. For example the circuit could add the following filters to the Oracle Access Manager Authenticate Authorization :-
1. Configure a ‘Validate HTTP Headers’ filter to validate that an existing token exists in the HTTP Header of the message.
2. Configure a ‘Retrieve from Attribute’ filter that will retrieve the existing token attribute from the HTTP Header of the message.

   ![Retrieve from HTTP header filter configuration](image)

3. Configure an ‘Oracle Access Manager Session Validation’ filter that will check for the validity of the SSO token.

   ![Oracle Access Manager Session Validation filter configuration](image)

4. Connect to the ‘Oracle Access Manager Authorization’ filter that was specified in the Oracle Access Manager Authenticate Authorize step.
Step: Insert WSBASIC Credentials
Inserts user name password credentials in a security soap header

**OWSM Configuration:**

<table>
<thead>
<tr>
<th>Insert WS BASIC Credentials Step</th>
<th>Environment Properties</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Basic Properties</strong></td>
<td><strong>Type</strong></td>
</tr>
<tr>
<td>Enabled (*)</td>
<td>boolean</td>
</tr>
</tbody>
</table>

**Oracle Enterprise Gateway Configuration:**
Drag an **“Insert WS-Security UsernameToken”** filter from the Authentication category to the target circuit path. This should follow on from extracting and authenticating credentials from another filter like HTTP Basic..

**Insert WS-Security Username Token**
Insert a WS-Security username token into the message for the client or issuer.

- **Name:** Insert WS-Security Username Token
- **Actor:** Current actor/role only

**Credential details**
- **User name:** ${authentication.subject.id}
- **Include Nonce**
- **Include Password**
  - **Password:**
  - **Clear** | **SHA1 Digest**
  - **Wildcard:** ${authentication.subject.password}

**Advanced Options**
- **Indent**

**Step: Ldap Authenticate**
Performs the authentication with a LDAP Server

**OWSM Configuration:**

<table>
<thead>
<tr>
<th>Basic Properties</th>
<th>Type</th>
<th>Default</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Enabled (*)</td>
<td>boolean</td>
<td>true</td>
<td>true</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Authentication Properties</th>
<th>Type</th>
<th>Default</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>LDAP host (*)</td>
<td>string</td>
<td>localhost</td>
<td>local</td>
</tr>
<tr>
<td>LDAP port (*)</td>
<td>int</td>
<td>389</td>
<td>389</td>
</tr>
<tr>
<td>LDAP SSL port</td>
<td>int</td>
<td>636</td>
<td>636</td>
</tr>
<tr>
<td>User objectclass (*)</td>
<td>string</td>
<td>inetOrgPerson</td>
<td>inetOrgPerson</td>
</tr>
<tr>
<td>LDAP baseDN (*)</td>
<td>string</td>
<td>ou=People,dc=corp,dc=oracle,dc=com</td>
<td>ou=People,dc=corp,dc=oracle,dc=com</td>
</tr>
<tr>
<td>LDAP adminDN (*)</td>
<td>string</td>
<td>ou=People,dc=corp,dc=oracle,dc=com</td>
<td>ou=People,dc=corp,dc=oracle,dc=com</td>
</tr>
<tr>
<td>LDAP admin password</td>
<td>string</td>
<td>***********</td>
<td></td>
</tr>
<tr>
<td>LDAP admin login enabled (*)</td>
<td>boolean</td>
<td>false</td>
<td>true</td>
</tr>
<tr>
<td>LDAPSSLEnabled (*)</td>
<td>boolean</td>
<td>false</td>
<td>true</td>
</tr>
<tr>
<td>Uid Attribute (*)</td>
<td>string</td>
<td>uid</td>
<td></td>
</tr>
</tbody>
</table>

User Attributes to be retrieved: string[]

**Oracle Enterprise Gateway Configuration:**

External connections > LDAP Connections > Add a LDAP

[Diagram of Oracle Enterprise Gateway Configuration]

Enter configuration details for the LDAP
Create an LDAP authentication repository entry which then can be used in HTTP Basic, Digest, and WS-Security username password filters for authentication.

External connections > Authentication Repository Profiles > LDAP Repositories > Add a new Repository

Configure the repository as required:
In a filter that requires authentication against LDAP, simply select the repository named above.
Step: Ldap Authorize
Authorizes request by retrieving role from LDAP and checking against roles allowed by service.

OWSM Configuration:

<table>
<thead>
<tr>
<th>Ldap Authorize</th>
<th>Basic Properties</th>
<th>Environment Properties</th>
</tr>
</thead>
<tbody>
<tr>
<td>Enabled (*)</td>
<td>boolean</td>
<td>true/false</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Authorization Properties</th>
<th>Type</th>
<th>Default</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>LDAP host (*)</td>
<td>string</td>
<td>localhost</td>
<td></td>
</tr>
<tr>
<td>LDAP port (*)</td>
<td>int</td>
<td>389</td>
<td>389</td>
</tr>
<tr>
<td>LDAP SSL port</td>
<td>int</td>
<td>636</td>
<td>636</td>
</tr>
<tr>
<td>LDAP baseDN (*)</td>
<td>string</td>
<td>ou=People,dc=corp,dc=oracle,dc=com</td>
<td>ou=People,dc=corp,dc=oracle,dc=</td>
</tr>
<tr>
<td>ServiceRoles</td>
<td>string[]</td>
<td>group1,group2</td>
<td></td>
</tr>
<tr>
<td>LDAPAdminDN (*)</td>
<td>string</td>
<td>ou=People,dc=corp,dc=oracle,dc=com</td>
<td>ou=People,dc=corp,dc=oracle,dc=</td>
</tr>
<tr>
<td>LDAPAdminPwd</td>
<td>string</td>
<td></td>
<td></td>
</tr>
<tr>
<td>LDAPAdminLoginEnabled (*)</td>
<td>boolean</td>
<td>false</td>
<td>true/false</td>
</tr>
<tr>
<td>LDAPSSLEnabled (*)</td>
<td>boolean</td>
<td>false</td>
<td>true/false</td>
</tr>
<tr>
<td>Uid Attribute (*)</td>
<td>string</td>
<td>uid</td>
<td></td>
</tr>
<tr>
<td>LDAP Group Object Class (*) string</td>
<td>groupofuniquenames</td>
<td>groupofuniquenames</td>
<td></td>
</tr>
</tbody>
</table>

Oracle Enterprise Gateway Configuration:
1. Use LDAP connection and repository setup as detailed in LDAP Authenticate above.
2. After Authentication with HTTP Basic, Digest or WS-Security username password filters send circuit to a "Retrieve Attribute from Directory Server" filter and then "Validate Message Attribute" filter. This combination can be used to check a role (group) for a user’s membership.
3. Repeat for each group that needs checking. e.g.
4. Example of a Retrieve from Directory Server for group “group1”
5. Example of a Validate Message Attribute for group “group1”
This filter returns success if the user is found to be a member of the role group1. The authorized user is forwarded to the next role check and finally to the user authorized policy which is shown as a shortcut in the circuit above.
6. Similarly the user not authorized branch is a shortcut. This could be to a policy that eventually reflects an appropriate failure message back to the client.

**Step: Log**
Log the request/response message

**OWSM Configuration:**
Oracle Enterprise Gateway Configuration:
Insert “Log Message Payload” filter from the Monitoring category into desired logging interception points in circuitry.

The Format field supports the following wildcards:

- **level:**
  - The log level (i.e. fatal, fail, success).
- **id:**
  - The unique transaction ID assigned to the message.
- **ip:**
  - The IP address of the client that sent the request.
- **timestamp:**
  The time that the message was processed in user-readable form.
- **filterName:**
  The name of the filter that generated the log message.
- **filterType:**
  The type of the filter that logged the message.
- **text:**
  The text of the log message that was configured in the filter itself.
- **payload:**
  The complete contents of the HTTP request, including HTTP headers, body, and attachments.

Then enable logging to file, database, syslog or console from a right click on the Fusion Gateway process.
Select the "Logging->Custom ..." to bring up the configure logging window. Example below shows the text file logging has been enabled.

or for a database logging first add the database connection details in the External Connections selection panel. Right click on "Database Connections" and select "Add a Database Connection". Fill out the database details like the example below and click OK.
then set the logging to the configured database connection from the database tab in the "configure logging" database tab
**Step: Oracle Access Manager Authenticate Authorize**

Authenticate and Authorize URLs access with Oracle Access Manager Access Server

**OWSM Configuration:**
Oracle Enterprise Gateway Configuration:
An Oracle Access Manager repository needs to be registered in the Gateway. To do this
1. Click on the "External Connections" navigation bar.
2. Right Click on the "Oracle Access Manager Repositories" and "add a new Repository"
3. Fill in the repository details as below including the path to the AccessServerSDK on the local gateway machine.
4. Authentication to Oracle Access Manager is handled by either the **HTTP Basic** (used in this example) or **HTTP Digest** filter from the Authentication
5. An **Oracle Access Manager Authorization** filter is used next to authorize an authenticated user for the current resource uri and http verb against Oracle Access Manager. The user must first have been authenticated to OAM via the **HTTP Basic** or **HTTP Digest** filter.
6. Once the user has been authenticated and authorized the circuit can be sent to the target Web Service.
7. A “Single Sign On” (SSO) token can be created and stored in a Fusion Gateway message attribute, so that it can be added later to the response that the client receives. The check box for creating an SSO token was ticked in the repository configuration. Add the session token to the HTTP headers with the ’Add HTTP Header’ filter from the ‘Conversion’ group.

8. The full policy diagram will look like this.
Step: SAML - Insert WSS 1.0 sender-vouches token
Step to Insert SAML token as per WSS 1.0 token profile with Sender-Vouches confirmation method

**OWSM Configuration:**

<table>
<thead>
<tr>
<th><strong>SAML - Insert WSS 1.0 sender-vouches token</strong></th>
<th>Environment Properties</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Basic Properties</strong></td>
<td></td>
</tr>
<tr>
<td>Enabled</td>
<td>boolean true</td>
</tr>
<tr>
<td><strong>Assertion Properties</strong></td>
<td></td>
</tr>
<tr>
<td>Subject Name Qualifier</td>
<td>string</td>
</tr>
<tr>
<td>Default</td>
<td><a href="http://www.company.com">www.company.com</a></td>
</tr>
<tr>
<td>Value</td>
<td><a href="http://www.company.com">www.company.com</a></td>
</tr>
<tr>
<td><strong>Assertion Issuer (*)</strong></td>
<td>string</td>
</tr>
<tr>
<td>Default</td>
<td><a href="http://www.company.com">www.company.com</a></td>
</tr>
<tr>
<td>Assertion valid till before current time (*): int</td>
<td>30</td>
</tr>
<tr>
<td>Value</td>
<td>30</td>
</tr>
<tr>
<td>Assertion valid till on/after current time (*): int</td>
<td>60</td>
</tr>
<tr>
<td>Value</td>
<td>60</td>
</tr>
<tr>
<td>User Attributes for attribute statements</td>
<td>string[]</td>
</tr>
<tr>
<td>Value</td>
<td>sn,uid</td>
</tr>
<tr>
<td>Corresponding namespace URIs for the user attributes</td>
<td>string[]</td>
</tr>
<tr>
<td><strong>Signing Properties</strong></td>
<td></td>
</tr>
<tr>
<td>Sign the assertion (*)</td>
<td>boolean true</td>
</tr>
<tr>
<td>Default</td>
<td>true</td>
</tr>
<tr>
<td>Value</td>
<td>false</td>
</tr>
<tr>
<td>Keystore location</td>
<td>string</td>
</tr>
<tr>
<td>Default</td>
<td>/opt/owsm-keystore.jks</td>
</tr>
<tr>
<td>Keystore Type</td>
<td>string</td>
</tr>
<tr>
<td>Default</td>
<td>jks</td>
</tr>
<tr>
<td>Keystore password</td>
<td>string</td>
</tr>
<tr>
<td>Value</td>
<td>**********************</td>
</tr>
<tr>
<td>Signature Method</td>
<td>string</td>
</tr>
<tr>
<td>Default</td>
<td>RSA-SHA1</td>
</tr>
<tr>
<td>Value</td>
<td>RSA-SHA1</td>
</tr>
<tr>
<td>Signing key alias</td>
<td>string</td>
</tr>
<tr>
<td>Value</td>
<td>mykey</td>
</tr>
<tr>
<td>Signing key password</td>
<td>string</td>
</tr>
<tr>
<td>Value</td>
<td>**********************</td>
</tr>
</tbody>
</table>

**Oracle Enterprise Gateway Configuration:**

Refer to the "Appendix" section for how to add Certificates and Keys to the Gateway Certificate Store.

The "Insert SAML Authentication Assertion" filter requires user credentials to be passed to it from a predecessor filter in a policy. A "HTTP Basic" filter will be added at the start of the policy. Use the "Sender Vouches" confirmation method to assert that the Gateway is acting on behalf of the authenticated end-user. No other information relating to the context of the assertion is sent. It is recommended that both the assertion and the SOAP Body must be signed if this option is selected. These message parts can be signed by using the "XML Signature Generation" filter.
1. Add a "HTTP Basic" filter from the "Authentication" filter category. For "Credential Format" select "User Name" from the drop down list.

2. For "Repository Name" select the configured authentication repository. By default only the "Local User Store" will be available. Authentication repositories can be configured under the "External Connections" module in Policy Studio and can be a database, LDAP etc... For this demonstration the "Local User Store" will be selected.

3. Next add a "Insert SAML Authentication Assertion" filter located in the "Authentication" filter category.

4. Under the "Assertion" tab configure the expiry of the assertion token. Select "Current Actor/Role" only for the "SOAP Actor/Role" value.

5. Select the version of SAML assertion required. Here 1.1 is selected.

6. For "Issuer Name" select the certificate containing the Distinguished Name (DName) that will be used as the Issuer of the SAML assertion. This DName will appear in the SAML assertion as the value of the Issuer attribute of the <saml:Assertion> element.

7. Under the "Confirmation Method" tab select "Sender Vouches" from the drop down list that will assert that the Gateway is acting on behalf of the authenticated end-user. No other information relating to the context of the assertion is sent.

8. The rest of the options can be left default. For more information on the configuration options for this filter please refer to the Gateway documentation or click on the help tab in the filter itself.

9. It is recommended that both the assertion and the SOAP Body must be signed if the "Sender Vouches" option is selected. These message parts can be signed by using the "XML Signature Generation" filter.

10. Connect the "HTTP Basic" to the "Insert SAML Authentication Assertion" via a success path.

11. Add a "XML Signature Generation" filter from the "Integrity" filter category.

12. Under the "Signing Key" tab choose the type of key to be used to sign the request. It is possible to use either a symmetric or an asymmetric key to sign the message content. Select the appropriate radio button and configure the fields on the corresponding tab.

13. Select the location of the Private Key to be used to sign the request. This can either be a key stored in the Certificate Store or in an available message attribute if the key has already been used by predecessor filters.

14. Once the key type has been configured, click on the "Key Info" tab where it can be configured how the <KeyInfo> block of the generated XML Signature will appear.

15. To achieve the same structure as that of the signed OWSM assertion select the "Security Token Reference" radio button, select "X509V3" from the drop down field. Then select the "Embed" radio button and tick the "Include TokenType" checkbox.

16. The "What to Sign" tab is used to configure to identify the parts of the message that must be signed. Each signed part will be referenced from within the generated XML Signature. Here it is recommended that the assertion and SOAP body be signed.

17. Select the "Use WS-Ids" option which uses wsu:Id attribute to reference the signed data. Also ensure that the ensure that "Use SAML Ids for SAML Elements" and "Add and Dereference Security Token Reference for SAML" is checked.

18. Select the part of the request that needs to be signed from either the "Nodes Location", "XPath", "XPath Predicates" or "Message Attribute" tab. The assertion and SOAP body is selectable under the the "Nodes Location" tab.

19. Under the "Advanced/Options" tab the "Insert CarriedKeyName for EncryptedKey" can be
unchecked. The rest of the options can be left default. For more information on the configuration options for this filter please refer to the Gateway documentation or click on the help tab in the filter itself.

20. Connect the “Insert SAML Authentication Assertion” to the “XML Signature Generation” filter via a success path.

The sample policy:

The HTTP Basic Filter:

The Insert SAML Authentication Assertion filter:
The Confirmation Method tab:
The Key Info tab in the XML Signature Generation filter:
The What to Sign tab in the XML Signature Generation filter:
Step: SAML - Verify WSS 1.0 Token
Verify SAML tokens as per WSS SAML token profile 1.0

OWSM Configuration:
Oracle Enterprise Gateway Configuration:

1. Add a “SAML Authentication” filter from the “Authentication” filter category.
2. Under the “Details” tab, “Assertion Location” section, choose the location of the SAML assertion by selecting the “SOAP Actor/Role” option or the “XPath” option.
3. Enforcement of the SAML namespace (SAML version) can be configured if required.
4. The rest options can be left default. For more information on the configuration options for this filter please refer to the Gateway documentation or click on the help tab in the filter itself.
5. Under the “Trusted Issuer” tab the issuer of the SAML assertion that requires verification needs to be entered or selected. To verify the request that has been created by OWSM for test purposes, enter www.company.com as the trusted issuer value.

The SAML Authentication verification filter:
The Trusted Issuer tab configuration:
**Step: Sign message**
Digitally Sign the message

**OWSM Configuration:**
Oracle Enterprise Gateway Configuration:
Refer to the "Appendix" section for how to add Certificates and Keys to the Gateway Certificate Store.

Messages can be signed in the Gateway using the "XML Signature Generation" filter. For configuring the settings to be the same as that configured in OWSM, please follow the steps below.

1. Add a "XML Signature Generation" filter located in the "Integrity" filter category.
2. Under the "Signing Key" tab choose the "Asymmetric Key" option.
3. Select the location of the Private Key to be used to sign the request. This can either be a key stored in the Certificate Store or in an available message attribute if the key has already been used by predecessor filters.
4. Once the key type has been configured, click on the "Key Info" tab where it can be configured how the <KeyInfo> block of the generated XML Signature will appear.
5. To achieve the same structure as that of a signed OWSM request, select the "Security Token Reference" radio button, select "X509v3" from the drop down field. Then select the "Embed" radio button and tick the "Include TokenType" checkbox.
6. The "What to Sign" tab is used to configure to identify the parts of the message that must be signed. Each signed part will be referenced from within the generated XML Signature.
7. Select either the "Use WS-Ids" option which uses wsu:Id attribute to reference the signed data or the "Use Ids" option that uses generic Id’s that references the signed data.
8. Select the part of the request that needs to be signed from either the "Nodes Location", "XPath", "XPath Predicates" or "Message Attribute" tab.
9. Under the "Where to place Signature", select where the signature is to be placed. As per OWSM, the option to select is the "In the WS-Security Element for SOAP Actor/Role". Then select "Current Actor/Role only" from the drop down field.
10. The "Advanced" tab contains additional but not mandatory options for example inserting a timestamp, select a particular algorithm for signing if required and other options. Ensure that "ShaRsa1" for the "Algorithm Method" and "Sha1" is selected for the "Digest Algorithm" option under the "Algorithm Suite" tab. Under the "Options" tab the "Insert CarriedKeyName for EncryptedKey" can be unchecked.
11. Please refer to Gateway documentation for more information on all the options available in the "Sign Message" filter.
The Signing Key tab settings:

The What to Sign tab settings:
The Where to Place Signature tab settings:

<table>
<thead>
<tr>
<th>Name</th>
<th>Namespace</th>
</tr>
</thead>
<tbody>
<tr>
<td>Assertion</td>
<td>urn:oasis:names:tc:SAML:2.0:assertion</td>
</tr>
<tr>
<td>Assertion</td>
<td>urn:oasis:names:tc:SAML:1.0:assertion</td>
</tr>
<tr>
<td>BinarySecurityToken</td>
<td><a href="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss">http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss</a>...</td>
</tr>
<tr>
<td>Body</td>
<td><a href="http://www.w3.org/2003/05/soap-envelope">http://www.w3.org/2003/05/soap-envelope</a></td>
</tr>
<tr>
<td>Body</td>
<td><a href="http://schemas.xmlsoap.org/soap/envelope/">http://schemas.xmlsoap.org/soap/envelope/</a></td>
</tr>
<tr>
<td>FaultTo</td>
<td><a href="http://www.w3.org/2005/08/addressing">http://www.w3.org/2005/08/addressing</a></td>
</tr>
<tr>
<td>FaultTo</td>
<td><a href="http://schemas.xmlsoap.org/ws/2004/08/addressing">http://schemas.xmlsoap.org/ws/2004/08/addressing</a></td>
</tr>
<tr>
<td>From</td>
<td><a href="http://www.w3.org/2005/08/addressing">http://www.w3.org/2005/08/addressing</a></td>
</tr>
<tr>
<td>From</td>
<td><a href="http://schemas.xmlsoap.org/ws/2004/08/addressing">http://schemas.xmlsoap.org/ws/2004/08/addressing</a></td>
</tr>
</tbody>
</table>
The Advanced Tab settings:
Step: Sign Message And Encrypt
XML Signature and Encryption

OWSM Configuration:
### Basic Properties

<table>
<thead>
<tr>
<th>Property</th>
<th>Type</th>
<th>Default</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Enabled (*)</td>
<td>boolean</td>
<td>true</td>
<td>true</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>false</td>
</tr>
</tbody>
</table>

### Signing Properties

<table>
<thead>
<tr>
<th>Property</th>
<th>Type</th>
<th>Default</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Signing Keystore location (*)</td>
<td>string</td>
<td></td>
<td>/opt/owsm-keystore.jks</td>
</tr>
<tr>
<td>Signing Keystore Type (*)</td>
<td>string</td>
<td>jks</td>
<td>jks</td>
</tr>
<tr>
<td>Signing Keystore password</td>
<td>string</td>
<td></td>
<td>******</td>
</tr>
<tr>
<td>Signer’s private-key alias (*)</td>
<td>string</td>
<td></td>
<td>mykey</td>
</tr>
<tr>
<td>Signer’s private-key password</td>
<td>string</td>
<td></td>
<td>******</td>
</tr>
<tr>
<td>Signature Algorithm (*)</td>
<td>string</td>
<td>RSA-SHA1</td>
<td>RSA-SHA1</td>
</tr>
<tr>
<td>Signed Content (*)</td>
<td>string</td>
<td>BODY</td>
<td>BODY</td>
</tr>
</tbody>
</table>

### Encryption Properties

<table>
<thead>
<tr>
<th>Property</th>
<th>Type</th>
<th>Default</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Encryption Keystore location (*)</td>
<td>string</td>
<td></td>
<td>/opt/owsm-keystore.jks</td>
</tr>
<tr>
<td>Encrypt Keystore Type (*)</td>
<td>string</td>
<td>jks</td>
<td>jks</td>
</tr>
<tr>
<td>Encryption Keystore password</td>
<td>string</td>
<td></td>
<td>******</td>
</tr>
<tr>
<td>Decryptor’s public-key alias (*)</td>
<td>string</td>
<td></td>
<td>mykey</td>
</tr>
<tr>
<td>Encryption Algorithm (*)</td>
<td>string</td>
<td>3DES</td>
<td>3DES</td>
</tr>
<tr>
<td>Key Transport Algorithm (*)</td>
<td>string</td>
<td>RSA-1_5</td>
<td>RSA-1_5</td>
</tr>
<tr>
<td>Encrypted Content (*)</td>
<td>string</td>
<td>BODY</td>
<td>BODY</td>
</tr>
</tbody>
</table>

### Environment Properties

Oracle Enterprise Gateway Configuration:
To sign and encrypt a request in the Gateway perform the XML Sign and XML Encrypt steps in one policy. Please see the XML Sign and XML Encrypt sections here for configuration of the relevant options.
NOTE: It is recommended to use different keys for signing and encryption.

A sample policy where the request is signed and encrypted.

![Diagram showing policy flow]

**Step: SiteMinder Authentication**
Authentication via SiteMinder

**OWSM Configuration:**

<table>
<thead>
<tr>
<th>SiteMinder Authentication</th>
<th>Configure</th>
<th>Add Step Below</th>
<th>Delete</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Basic Properties</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Enabled (*)</td>
<td>boolean</td>
<td>true</td>
<td>true</td>
</tr>
<tr>
<td><strong>Authentication Properties</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SmServer host (*)</td>
<td>string</td>
<td>localhost</td>
<td>10.0.7.231</td>
</tr>
<tr>
<td>SmAgent name (*)</td>
<td>string</td>
<td>agentname</td>
<td>gatewayagent</td>
</tr>
<tr>
<td>SmAgent secret</td>
<td>string</td>
<td>********</td>
<td></td>
</tr>
<tr>
<td>Resource (*)</td>
<td>string</td>
<td>/basicauthn</td>
<td>/basicauthn</td>
</tr>
<tr>
<td>Operation (*)</td>
<td>string</td>
<td>ProcessSoap</td>
<td>ProcessSoap</td>
</tr>
</tbody>
</table>

**Oracle Enterprise Gateway Configuration:**

**Register a SiteMinder Agent in the Gateway:**
The SiteMinder agent needs to be registered under the “SiteMinder/SOA Security Manager Connections” object under the “External Connections” module.

1. Click on the “External Connections” module.
2. Right click on “SiteMinder/SOA Security Manager Connections” and select “Add SiteMinder Connection”.
3. Click on the “Register” button.
4. Enter the “IP Address” of the SiteMinder Policy Server.
5. Enter the “User Name” and “Password” of the account used to connect to the SiteMinder Policy Server.
6. Under “Agent Details” enter a value for the “Name of the host to be registered”. This can be ANY descriptive value and should not already exist under the “Trusted Host” list on the SiteMinder Policy Server.
7. Enter the “Name of the host configuration object” as is configured in SiteMinder.

The SiteMinder Agent configuration window:

![Register Host window](image)

Add a SiteMinder Authentication Repository:
Once the SiteMinder Agent has been registered, expand the “Authentication Repository Profiles” object under the “External Connections” tree.

1. Right click on “CA SiteMinder Repositories” and select “Add a new Repository”.
2. Enter a name for the repository. For this guide “SiteMinder” is used.
3. Select the configured agent in the drop down menu for the “Agent Name” value.
4. The rest of the options can be left default. Click on the “Help” button if more details are required for the configuration options here.

The SiteMinder Authentication Repository configuration window:
The Siteminder Authentication Repository is now available for any filter that can connect to a configured Repository for example the "HTTP Basic" filter where available repositories where users are stored can be selected from a drop down menu. The "HTTP Basic" filter is available in the "Authentication" filter group.

The HTTP Basic filter configured to connect to the Siteminder Repository.

Example of the HTTP Basic filter for Authentication via Siteminder in a policy:
For more information on how the Gateway integrates with SiteMinder please refer to the Fusion Gateway documentation as well as the Fusion Gateway Integration with SiteMinder Guide.

**Step: SiteMinder Authorize**
SiteMinder Authorization to be used after SiteMinder Authentication Step

**OWSM Configuration:**

<table>
<thead>
<tr>
<th>SiteMinder Authorize</th>
<th></th>
<th></th>
<th>Configure</th>
<th>Add Step Below</th>
<th>Delete</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Basic Properties</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Enabled (*)</td>
<td>bool</td>
<td>true</td>
<td>true</td>
<td></td>
<td></td>
</tr>
<tr>
<td>TransactionId</td>
<td>string</td>
<td>CCore</td>
<td>CCore</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Oracle Enterprise Gateway Configuration:**
Authorization via SiteMinder is configured by adding the SiteMinder Validation and SiteMinder Authorization filter to a policy, normally after having authenticated via SiteMinder.

1. Add a “**Session Validation**” filter located in the “**CA SiteMinder**” filter category.
2. For the “**Agent Name**” select the available SiteMinder agent in the drop down list as configured in the **SiteMinder Authentication** section above.
3. Ensure that the “**Message Attribute Containing Session**” is set to “**siteminder.session**”.
4. For the “**Resource**” and “**Action**” fields the values can be left default. The Gateway will retrieve these attributes from the request and the wildcarded values will be set to those matching attributes.
5. Next is to add the "**Siteminder Validation**" filter also located in the "**CA SiteMinder**" filter category.

6. Additionally, an attribute can be configured that is used as a SAML Attribute Token for consumption downstream if required. By default this is not enabled.
The policy:

For more information on how the Gateway integrates with SiteMinder please refer to the Fusion Gateway documentation as well as the Fusion Gateway Integration with SiteMinder Guide.

**Step: Verify Certificate**
Verify a certificate against a local keystore.

**OWSM Configuration:**

<table>
<thead>
<tr>
<th>Verify Certificate</th>
<th>Configure</th>
<th>Add Step Below</th>
<th>Delete</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Basic Properties</strong></td>
<td><strong>Type</strong></td>
<td><strong>Default</strong></td>
<td><strong>Value</strong></td>
</tr>
<tr>
<td>Enabled (*)</td>
<td>boolean</td>
<td>true</td>
<td>true</td>
</tr>
<tr>
<td>Keystore location (*)</td>
<td>string</td>
<td>/opt/owsm-keystore.jks</td>
<td></td>
</tr>
<tr>
<td>Keystore password</td>
<td>string</td>
<td>*********</td>
<td></td>
</tr>
</tbody>
</table>

**Oracle Enterprise Gateway Configuration:**
Refer to the "Appendix" section for how to add Certificates and Keys to the Gateway Certificate Store.

After the certificates and keys have been loaded to the Gateway Certificate Store, any filter using the Certificate Store has access to these Certificate and Keys.

For Certificate Validation the Gateway has the following filters available under the "Certificates" filter category that can be used in any policy.

The "Certificate Chain" filter which ensures that a certificate has been issued by a trusted source is the closest match to the "Verify Certificate" option in OWSM.

To configure the "Certificate Chain" filter:

1. Add a "Certificate Chain" filter from the "Certificates" filter category into the policy palette.
2. Select the Certificate Authority which is to be trusted to issue certificates to clients by selecting the checkbox next to the certificate D-Name. Multiple CA certificates can be selected.

3. The "Certificate Chain" filter has to be preceded by a filter that generates the "Certificates" attribute, which is a list containing certificates that has been retrieved by predecessor filters in the Gateway. Normally this will be one of the following filters: SAML PDP XML-Signature Response Verification, SAML PDP Response XML-Signature Verification, HTTP Header Authentication, Integrity, XML-Signature Verification, SAML, XML-Signature Verification, SAML Authorization, XML-Signature Verification, SSL Authentication and XML-Signature Authentication for example.

The Cert Chain filter:

```
Configure a new 'Certificate Chain' filter

Certificate Chain Check
Verify that a trusted Certificate Authority (CA) issued the certificate.

Name: Certificate Chain

Select the CAs which are trusted to issue certificates to clients:

- [ ] CN=Baltimore CyberTrust Root, OU=cybertrust, C=US
- [ ] CN=DigiCert Assured ID Root CA, O=digicert.com, C=US
- [ ] CN=DigiCert Global Root CA, OU=digicert.com, C=US
- [ ] CN=DigiCert High Assurance SV Root CA, OU=digicert.com, C=US
- [ ] CN=Entrust.net Certification Authority, O=entrust.net, C=US
- [ ] CN=Entrust.net Certificate Authority, O=entrust.net, C=US
- [ ] CN=Entrust.net Secure Server Certification Authority, O=entrust.net, C=US

Finish Cancel
```

The Cert Chain filter used in a simple policy

```
Cert Chain Check

Start/End SSL

End Certificate Chain

End Call 'Main Policy'
```
In addition to the Cert Chain filter for certificate validation the Gateway is also capable of providing certificate validation using the following technologies:

- **CRL Dynamic** (Validate certificate against a Certificate Revocation List URL)
- **CRL Static** (Validate certificate against a local Certificate Revocation List file)
- **CRL LDAP** (Validate certificate against a Certificate Revocation List located in a LDAP directory)
- **CRL Responder** (Allows Gateway to behave as a Certificate Revocation List) responder)
- **Certificate Chain** (Ensures that a certificate has been issued by a trusted source)
- **Certificate Validity** (Performs a simple check on a certificate to make sure that it has not expired.)
- **OCSP** (Validate certificate against an Online Certificate Status Protocol responder)
- **XKMS** (Validate certificate against a XML Key Management Service responder)

For more information about configuring these filters please refer to the Fusion Gateway Administrators Guide. Please also refer to the SSL Integrity and Confidentiality guide.

**Step: Verify Signature**
XML Signature Verification

**OWSM Configuration:**

<table>
<thead>
<tr>
<th>Verify Signature</th>
<th>Configure</th>
<th>Add Step Below</th>
<th>Delete</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Basic Properties</strong></td>
<td>Type</td>
<td>Default</td>
<td>Value</td>
</tr>
<tr>
<td>Enabled (*)</td>
<td>boolean</td>
<td>true</td>
<td>true</td>
</tr>
</tbody>
</table>

**XML Signature Verification Properties**

<table>
<thead>
<tr>
<th>Keystore location (*)</th>
<th>string</th>
<th>/opt/owsm-keystore.jks</th>
</tr>
</thead>
<tbody>
<tr>
<td>Verifying Keystore Type (*)</td>
<td>string</td>
<td>jks</td>
</tr>
<tr>
<td>Keystore password</td>
<td>string</td>
<td>**********</td>
</tr>
<tr>
<td>Signer's public-key alias (*)</td>
<td>string</td>
<td>jpkay</td>
</tr>
<tr>
<td>Remove Signatures (*)</td>
<td>boolean</td>
<td>true</td>
</tr>
<tr>
<td>Enforce Signing (*)</td>
<td>boolean</td>
<td>true</td>
</tr>
</tbody>
</table>

**Oracle Enterprise Gateway Configuration:**
The Gateway can verify signed messages using the **XML Signature Verification** filter.

1. Add a **XML Signature Verification** filter from the **Integrity** filter category to the policy palette.
2. Under the **Signature Location** section choose the location where the signature resides from the drop down list.
3. Set the **Signature Position** to 1.
4. Under the **Find Signing** key section select the location of the public key that will be used to verify the signature.
5. Under the **What Must Be Signed** tab the particular part of the message can be selected that
should be signed, although not mandatory, if selected strengthens the integrity verification process, if it is known that a specific part of the message should be signed.

6. The "Advanced" tab covers additional options, for example to fail if no signature is present, which can be enabled or disabled. Please refer to Gateway documentation for explanation of all features in the filter.

The Signature Verification configuration

The What Must be Signed configuration
The Advanced configuration
Step: XML Decrypt

XML Decryption

OWSM Configuration:
Oracle Enterprise Gateway Configuration:
To decrypt messages in the Gateway, the “Decryption Settings” filter followed by the “Decryption” filter is used.

1. Add a “Decryption Settings” filter from the “Encryption” filter category.
2. Under the “Nodes to Decrypt” section there are two options to select from. Select “Decrypt All” to decrypt all encrypted content in a request or “Use XPath” to explicitly select the EncryptionData block in a request.
3. Under the “Decryption Key” section select where the decryption key will be retrieved from. The options are from the “Key Info” section in the request or via a message attribute generated by predecessor filters in the policy.
4. Under the options section the filter can be configured to fail if not encryption data was found in the request and also to remove the encrypted key element used for decryption.
5. Add an “Decryption” filter from the “Encryption” filter category. No configuration is required in this filter. Connect to the “Decryption Settings” filter via a success path.
6. For information on all the options please refer to the Gateway documentation.

A sample Decryption Policy:

![Diagram of Decryption Policy]

The Decryption Settings filter:
Step: XML Encrypt
XML Encryption

OWSM Configuration:
Oracle Enterprise Gateway Configuration:
To encrypt messages in the Gateway, the "Encryption Settings" filter followed by the "Encryption" filter is used. The "Encryption Settings" filter is dependant on the previous filters in the policy chain to supply the "Certificate" attribute that holds the key that will be used to encrypt the request. For this the "Find Certificate" filter will be used to locate the key in the Gateway certificate store.

1. Add a "Find Certificate" filter from the "Certificates" filter category.
2. Select the option where the key is to be obtained from. In this example the "Certificate Store" option has been selected with the relevant certificate/key combination chosen from the list of certificates and keys then click on "Finish".
4. Under the "Encryption Key" tab the settings can be left as default.
5. Under the "Key Info" tab select the "Security Token Reference" radio button and select "X509v3" from the drop down list. Also select "Embed" option and select the "Include Security Token" check box.
6. Under the "Recipients" tab the options can be left as default.
7. Under the "What to Encrypt" tab select the part of the request that needs to be encrypted from either the "Nodes Location", "XPath" or "Message Attribute" tab.
8. The "Advanced" tab contains additional options for example inserting a timestamp, select a particular algorithm for encryption if required and other options. Select "TripleDes" under for the "Encryption Algorithm" option under the "Algorithm Suite" tab.
9. Click on "Finish" once the configuration of the "Encryption Settings" filter is complete and connect to the "Find Certificates" filter via a success path.
10. Add an "Encryption" filter from the "Encryption" filter category. No configuration is required in this filter. Connect to the "Encryption Settings" filter via a success path.
11. Please refer to the Gateway documentation for more information on all the options in the "Encryption Settings" filter.

Example Encryption Policy:

The Find Certificates filter:
The Encryption Key tab settings:

**Encryption Key**
- Generate Encryption Key
- Encryption Key in message attribute: symmetric.key

Note: The encryption key is a symmetric key. It may be encrypted asymmetrically for each recipient.
- Include encryption key in message

**Specify method of associating the Encryption Key with the Encrypted Data**
- Point to Encryption Key with Security Token Reference
- Embed symmetric key inside Encrypted Data
- Specify Encryption Key via carried keyname
- Specify Encryption Key via retrieval method
- Encryption Key refers to Encrypted Data

Use derived key

**Key Info Format Description:**
The encryption key is included in the message and encrypted for the recipient. The EncryptedData KeyInfo points to the EncryptedKey with ValueType "#.EncryptedKey". The EncryptedKey KeyInfo is defined via the Key Info tab.

The Key Info tab Settings:
The Recipients tab settings:
The What to Encrypt tab settings:

<table>
<thead>
<tr>
<th>Recipients</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default Recipient</td>
</tr>
</tbody>
</table>
The Advanced tab settings:
Step: XML Transform
Transform message using XSL

OWSM Configuration:

<table>
<thead>
<tr>
<th><strong>XML Transform</strong></th>
<th><strong>Environment Properties</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Basic Properties</strong></td>
<td><strong>Type</strong></td>
</tr>
<tr>
<td>Enabled (*)</td>
<td>boolean</td>
</tr>
<tr>
<td>XSLTUrl</td>
<td>string</td>
</tr>
<tr>
<td>XSLTFileName</td>
<td>string</td>
</tr>
</tbody>
</table>

Where

**XSLTUrl** is the URL that specifies the location of the XSLT file.

**XSLTFileName** is the Path to the XSLT file on the machine where Oracle WSM is installed.

**Oracle Enterprise Gateway Configuration:**

Drag and drop a Stylesheet Conversion filter from the Utility class into the circuit at the desired point.
Appendix:

Oracle Enterprise Gateway Certificate Store:

Adding Certificates and Keys to the Oracle Enterprise Gateway Certificate Store:
If OWSM is using certificates or keys contained in java keystore file then this can be imported into the Gateway's configuration via the following steps:

1. Add Certificates and Keys from the OWSM Keystore to the Oracle Enterprise Gateway Certificate Store
Click on the “Certificates” module in Policy studio then select “Keystore”, click on the “Browse” button and browse to the keystore file and connect to it using Keystore password
A list of certificates and keys will be show. Select the certificates and keys to export to the Gateway Certificate store. Then click on “Import to Trusted Certificate Store”.

2. To add a Certificate or Certificate and Key to the Oracle Enterprise Gateway Certificate store that does not reside in another Keystore:
   1. Click on the “Certificates” module in the left side of Policy Studio.
   2. Click on the “Certificates” object at the top.
   3. On the right hand side of Policy Studio click on “Create/Import” button.
   4. Click on the “Import Certificate” button if it is a certificate with public key that needs importing.
   5. Click on the “Import Certificate and Key” option to import a certificate and private key.
   6. Certificate and Key can also be imported separately by following step 4 for importing the certificate, then click on the “Private Key” tab where the corresponding private key can be imported.