



An Oracle White Paper
June 2011

Oracle Internet Directory 11g – 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.

Table of Contents

1. Introduction	4
1.1. Purpose.....	4
1.2. LDAP Architecture	5
1.3. Setup Used for this Guide:.....	5
2. Directory Details.....	5
2.1. Directory Structure	5
2.2. Connection Details.....	5
3. Authenticate User with HTTP Basic HTTP Filter	6
STEP 1: Create Policy to Authenticate User in LDAP directory.....	6
STEP 2: Create a new relative path for the Policy.....	13
STEP 3: Ensure policies are updated on the Gateway.....	13
STEP 4: Test the configuration in OEG Service Explorer	13
4. Adding a Retrieve from Directory Server filter	15
STEP1: Modify the Policy to include an 'Retrieve from Directory Server' filter	15
STEP 1: Configuring the Retrieve from Directory Server Filter:.....	16
STEP 2: Refresh Gateway Configuration	20
STEP 3: Test the configuration in OEG Service Explorer.....	20
5. Adding an Insert SAML Authentication Assertion filter	22
STEP:1 Add an "Insert SAML Authentication Assertion" filter	22
STEP 2: Configuring the 'Insert SAML Authentication Assertion' filter:	24
STEP 3: Test the configuration in OEG Service Explorer.....	24
6. Conclusion	26
7. Appendix	26

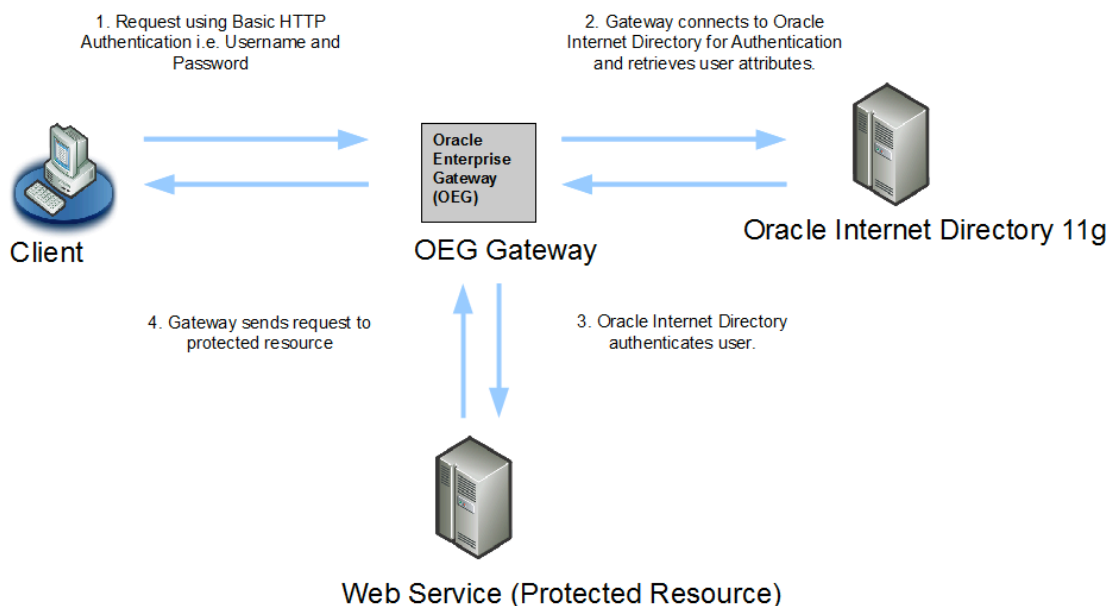
1.Introduction

1.1. Purpose

This document describes how to configure the Gateway to authenticate via an Oracle OID directory server and to extract attributes/roles from the LDAP repository. This will be demonstrated by the following:

- ✦ The Gateway will be configured to authenticate a user located in the Oracle OID LDAP directory.
- ✦ Upon successful authentication the Gateway will be configured to extract attributes belonging to this user from the Oracle OID LDAP directory.
- ✦ A SAML Authentication Assertion will also be added to demonstrate Single Sign On capability.

Flow of request:



This guide applies to OEG software products, from version 11.1.1.x upwards.

In this guide the LDAP Server used is **Oracle Internet Directory 11g**.

1.2. LDAP Architecture

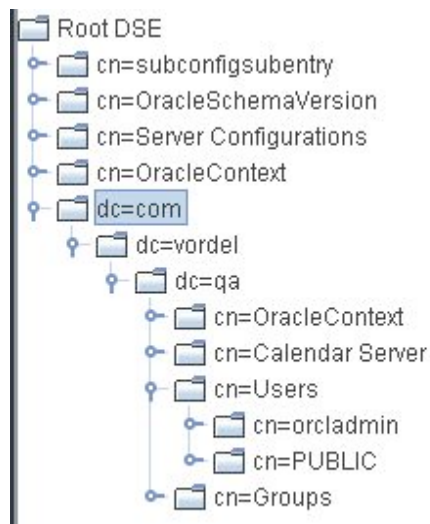
LDAP refers to *Lightweight Directory Access Protocol*. LDAP is based on a simplified version of X.500 directories. It is used to access a hierarchical directory of information on a directory server.

1.3. Setup Used for this Guide:

- ▲ Gateway 11.1.1.x
- ▲ Oracle OID Server 11g
- ▲ LDAP Browser

2. Directory Details

2.1. Directory Structure



The details of this directory are displayed here in a LDAP browser:

2.2. Connection Details

The connection details for this Oracle OID LDAP directory are:

- LDAP URL: ldap://oid11g.qa.vordel.com:3060
- user: cn=orcladmin
- password: vordel12

NOTE: The connection details referenced here are specific to the implementation used for this guide.

3. Authenticate User with HTTP Basic HTTP Filter

STEP 1: Create Policy to Authenticate User in LDAP directory

The first policy that will be created is to authenticate an existing user located in the Oracle OID LDAP directory. Before creating this policy it will be necessary to create a LDAP Connection and a LDAP Repository.

Create a policy to authenticate an existing user located in an LDAP directory:

- Start Policy Studio by running “policystudio.exe” (Windows) or “policystudio.sh” (Unix/Solaris) from the Policy Studio root directory.
- Click on the Gateway process listed to open the configuration window in a new tab.
- Click on the “External Connections” module.
- Right Click on “LDAP Connections” and Click “Add a LDAP Connection”
- Name: For this guide “Oracle OID” is used
- For the “Type” dropdown box select “Simple”
- Enter the Connection details to connect to the LDAP directory
- Realm: Leave blank
- Click on “Test Connection” to verify that the connection to the LDAP database has been configured successfully
- Click on “OK”
- The new “LDAP Connection” should be visible in the “LDAP Connections” Tree
- Within the “External Connections” Tree expand the “Authentication Repository” Profiles Tree
- Right Click on “LDAP Repositories” and Click “Add a new Repository”
- Repository Name: For this guide “OID11g” is used
- LDAP Store: For the “LDAP Directory” choose the previously created LDAP connection “Active Directory” from the drop down list
- Now the “User Search Conditions” needs to be specified
- For this guide the following details are used based on the Directory information above:
 - o Base Criteria: cn=Users, dc=qa, dc=vordel, dc=com

-
- o User Class: 'inetOrgPerson' LDAP Class (from the drop down list)
 - o User Search Attribute: cn
 - For “Attributes for use in subsequent filters” enter the following values (see NOTE 2):
 - o Login Authentication Attribute: This can be left blank
 - o Authorization Attribute: cn
 - o Authorization Attribute Format: User Name
 - Click “OK”
 - The new LDAP Repository should now be visible in the “LDAP Repositories” Tree
 - Click on “Policies” and then Right Click on the “Policies” Tree on the left hand side of Policy Studio
 - Click “Add Policy” and name the Policy “Oracle OID”
 - Click on the Policy and drag a “HTTP Basic” filter located in the “Authentication” group of the filter palette located on the right pane of “Policy Studio”
 - Name of the filter can be left default or changed to any descriptive name.
 - Credential Format: select User Name from the drop down list
 - Repository Name: For this guide “Oracle OID” is used
 - Click on “Finish”
 - Add a reflect filter from the “Utilities” filter category and connect the HTTP Basic filter to it with a success path connector.

NOTE 1: Explanation of “User Search Conditions” values

How the LDAP Authentication filter works:

The first step is to find the entry for the user in the Directory Server.

- o Base Criteria: cn=Users, dc=qa, dc=vordel, dc=com
- o User Class: 'inetOrgPerson' LDAP Class (from the drop down list)
- o User Search Attribute: cn

The query that will be run looks like this:

```
(&(objectclass=inetOrgPerson)(cn=${authentication.subject.id}))
```

The query describes the following:

Look for the object in the hierarchy of type 'inetOrgPerson' where the attribute 'CN' can be used to identify the user in the hierarchy under 'cn=Users, dc=qa, dc=vordel, dc=com'

In summary, the two fields in the LDAP repository "User class" and "User search attribute" are both combined to create a search filter:

```
(&(objectclass=***User class value goes here ***)(***User search attribute goes here***=***Authentication username from HTTP Header goes here ***))
```

If the user is found in the Directory Server then the Distinguished Name is returned, in this case:

```
cn=orcladmin, cn=Users, dc=qa, dc=vordel, dc=com
```

NOTE 2: Explanation of “Attributes for use in subsequent filters” values

Once the user is found the second step is for the Gateway to attempt to "bind" to the Directory Server on behalf of the user, using the Distinguished Name returned from the search and the password provided by the user for authentication. If the Gateway can bind to the Directory Server on behalf of the client then the HTTP Basic authentication filter will pass otherwise it will fail.

In the “Login Authentication Attribute” user friendly strings map to the following:

- Distinguished name=distinguishedName
- Entry Domain Name=entrydn

If the "Login Authentication Attribute" is left blank it means that the Gateway will then automatically concatenate the specified Base Criteria: cn=Users, dc=qa, dc=vordel, dc=com with the contextualized DName returned from the directory server in the first lookup (i.e. "cn=orcladmin") to obtain the fully qualified DName (i.e. "cn=orcladmin,cn=Users, dc=qa, dc=vordel, dc=com").

The Repository configuration window also has a section titled “Attributes for use in subsequent filters”

For this guide this section has been configured as follows:

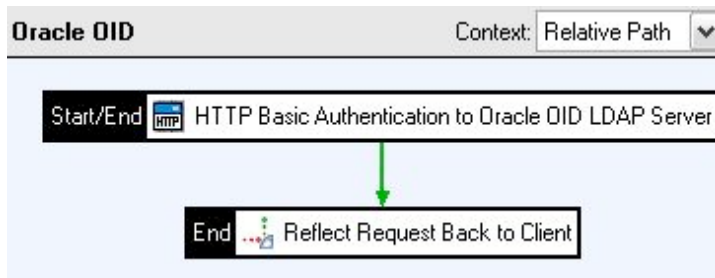
- o Login Authentication Attribute: Leave blank
- o Authorization Attribute: cn
- o Authorization Attribute Format: User Name

In “Attributes for use in subsequent filters” the first field “Login Authentication Attribute” is used in the second step (binding to LDAP). The “Login Authentication Attribute” is the attribute that is retrieved from the first step above which can be used to uniquely identify the user in the Directory Server (this is normally the Distinguished

Name), this is then used in step two as to who the Gateway binds to the LDAP Directory Server as.

For subsequent transactions (i.e. authorization) it is possible to use an attribute/s retrieved from the original LDAP search to authorize the user, in the example above the "distinguishedName" attribute contained in the user object in the Directory Server has been selected and setting this to be the value used for authorization.

The completed Policy will look as follows:



The configuration of the HTTP Basic filter as described above:

Configure "HTTP Basic"

HTTP Basic Authentication
Configure authentication using HTTP basic.

Name:

Credential Format:

Allow client challenge
 Remove HTTP authentication header

Repository Name:

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

The Connection settings window:

Configure LDAP Server

Name:

URL:

Cache Timeout:

Cache Size:

Authenticate LDAP Requests

Type:

User Name:

Password:

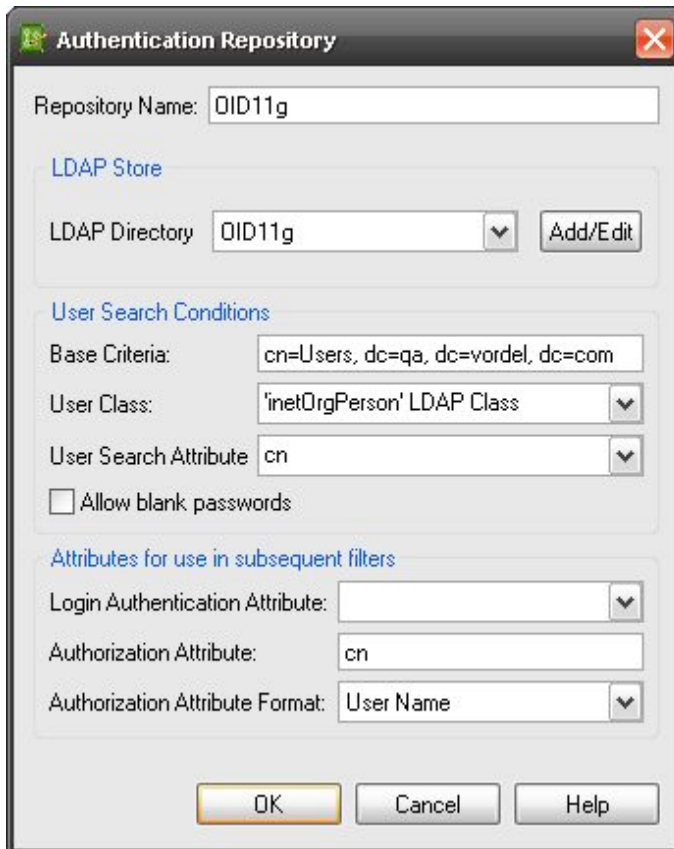
Realm:

SSL Enabled

Additional JNDI properties:

Name	Value
------	-------

The Authentication Repository configuration:



The screenshot shows the 'Authentication Repository' configuration dialog box. It has a title bar with a close button (X) in the top right corner. The dialog is organized into several sections:

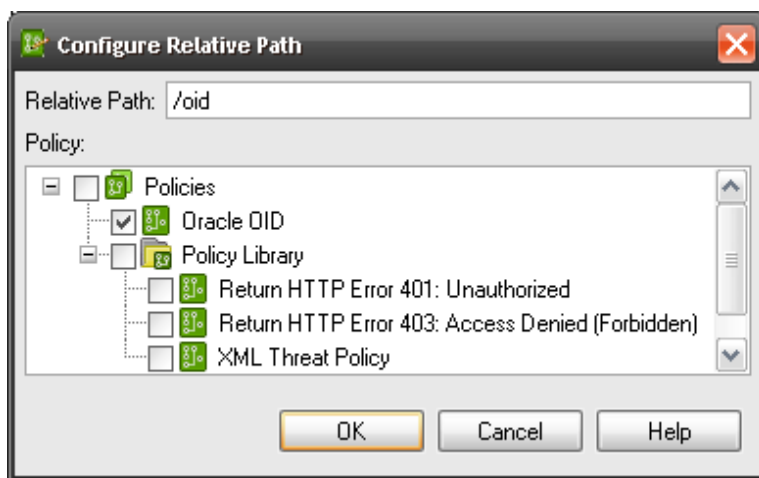
- Repository Name:** A text field containing 'OID11g'.
- LDAP Store:** A section header in blue text. Below it, 'LDAP Directory' is a dropdown menu showing 'OID11g' with a small downward arrow, and an 'Add/Edit' button to its right.
- User Search Conditions:** A section header in blue text. Below it:
 - 'Base Criteria:' is a text field containing 'cn=Users, dc=qa, dc=vordel, dc=com'.
 - 'User Class:' is a dropdown menu showing ''inetOrgPerson' LDAP Class' with a downward arrow.
 - 'User Search Attribute' is a dropdown menu showing 'cn' with a downward arrow.
 - There is an unchecked checkbox labeled 'Allow blank passwords'.
- Attributes for use in subsequent filters:** A section header in blue text. Below it:
 - 'Login Authentication Attribute:' is a dropdown menu that is currently empty.
 - 'Authorization Attribute:' is a text field containing 'cn'.
 - 'Authorization Attribute Format:' is a dropdown menu showing 'User Name' with a downward arrow.

At the bottom of the dialog, there are three buttons: 'OK' (highlighted with a yellow border), 'Cancel', and 'Help'.

STEP 2: Create a new relative path for the Policy

- Click on the “Services” module in Policy Studio.
- Expand “Processes”, “Gateway” and right click on the “Default Services”.
- Select “Add Relative Path” and enter: /oid
- Map the path to the policy titled ‘Oracle OID’
- Click ‘OK’

The Add a relative path window:

**STEP 3: Ensure policies are updated on the Gateway**

- Refresh the Gateway by pressing the “F6” key or select “Settings” located in the top menu of Policy Studio and click on “Deploy F6”

STEP 4: Test the configuration in OEG Service Explorer

To test the policy OEG Service Explorer can be used to send through a message embedded with user credentials (Username/Password)

- Start OEG Service Explorer by running “serviceexplorer.exe” (win32) or “serviceexplorer.sh” (UNIX) located in the OEG Service Explorer root directory.
- Load a message request
- Click on “Request Settings” on the drop down list on the green “Send Request” button

-
- Make sure that the URL is set correctly. In this case it will be
 http://gateway_ip:8080/oid
 - Click on the “Security” tab followed by the “HTTP Authentication” tab
 - Select “HTTP Basic” and enter the Username and Password of the user that will be authenticated via the Oracle OID server.
 - User Credentials used for this demonstration is:
 - oUser: orcladmin
 - oPassword: vordel12
 - Click on “Run”

Here is an extract from the Gateway trace showing the successful authentication via OID:

```

-----
DEBUG 12:31:07:589 [1304] run circuit "Oracle OID"...
DEBUG 12:31:07:589 [1304] run filter [HTTP Basic] {
DEBUG 12:31:07:589 [1304] LDAPRepository.checkCredentials: Check user via LDAP
DEBUG 12:31:07:589 [1304] LDAPRepository.getQueryResultsFromCache.
Key=orcladmin::cn=Users, dc=qa, dc=vordel, dc=com:inetOrgPerson:cn
DEBUG 12:31:07:589 [1304] LDAPRepository.getQueryResultsFromCache. Expired
DEBUG 12:31:07:605 [1304] LDAP search to be run:
(&(objectClass=inetOrgPerson)(cn=orcladmin))
DEBUG 12:31:07:605 [1304] adding the additional jndi properties: {}
INFO 12:31:07:605 [1304] cache
com.vordel.common.ldap.LdapLookup$ContextCache@e596c9 grows to 1
DEBUG 12:31:07:621 [1304] }
DEBUG 12:31:07:621 [1304] loginUser - dname to be used: cn=orcladmin,cn=Users,
dc=qa, dc=vordel, dc=com
DEBUG 12:31:07:621 [1304] Attempting to authenticate the user:
cn=orcladmin,cn=Users, dc=qa, dc=vordel, dc=com
DEBUG 12:31:07:636 [1304] The authenticated the user: cn=orcladmin,cn=Users, dc=qa,
dc=vordel, dc=com
DEBUG 12:31:07:636 [1304] LDAPRepository.addQueryResultsFromCache.
Key=orcladmin::cn=Users, dc=qa, dc=vordel, dc=com:inetOrgPerson:cn
DEBUG 12:31:07:636 [1304] UsernameAuthN.getResponse: Mapped 'orcladmin' to
'orcladmin'. Format=Username
DEBUG 12:31:07:636 [1304] } = 1, in 47 milliseconds
DEBUG 12:31:07:636 [1304] run filter [Reflect] {
-----

```

- The request will be sent to the Gateway which will connect to Oracle OID to authenticate the user 'orcladmin'

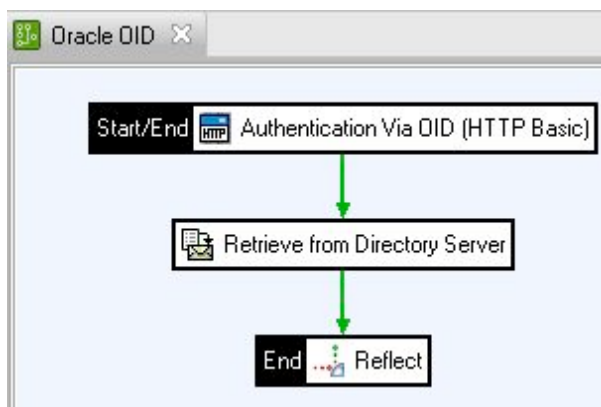
4. Adding a Retrieve from Directory Server filter

By having successfully authenticated a user from using an LDAP lookup, it is now possible to retrieve attributes from this user.

STEP1: Modify the Policy to include an 'Retrieve from Directory Server' filter

- Click on the "Oracle OID" policy
- From the "Attributes" filter category add a "Retrieve from Directory Server" filter to the circuit. For configuration see STEP 1 below
- The flow of the filters will be, HTTP Basic->Retrieve from Directory Server->Reflect all connected with success path connectors

The modified Policy:



STEP 2: Configuring the Retrieve from Directory Server Filter:

- LDAP Directory: (choose LDAP directory from the drop down list as configured in Step 1)
- Retrieve Unique User Identity: Two options are available here to choose from
 - o From Message Attribute: select "authentication.subject.id" (as this attribute is provided by having authenticated using the Basic HTTP filter)

Select this option if the user ID is stored in a message attribute. A user's credentials are stored in the `authentication.subject.id` message attribute after authenticating to the Gateway and so this is the most likely attribute to enter in this field. Typically this will contain the Distinguished Name (DNName) or username of the authenticated user. The name extracted from the selected message attribute will be used to query the directory server.

- o From LDAP Search: This option can be used to specify a search location in the directory for a required attribute.

Select this option to configure the Gateway to retrieve the user's identity from an LDAP search. Click the Configure Directory Search button to configure the search criteria to use to retrieve the user's identity. This option can be selected in cases where the `authentication.subject.id` attribute has not been pre-populated by an authentication filter. In this case the user's unique Distinguished Name must be retrieved from the LDAP repository.

Retrieve Unique User Identity from Message Attribute:

- Base Criteria: `cn=Users, dc=qa, dc=vordel, dc=com`

- Search Filter: `(&(objectclass=inetOrgPerson)(cn=${authentication.subject.id}))`

Retrieve Unique User Identity from LDAP Search:

- Search Scope: Sub Tree is selected

- Query Search Filter: `(&(objectclass=inetOrgPerson)(cn=${authentication.subject.id}))`

- The Attribute Name table lists the attributes that the Gateway will retrieve from the user profile. If no attributes are explicitly listed here, the Gateway will extract all user attributes. In both cases, the retrieved attributes will be set to the `attribute.lookup.list` message attribute. For this guide an additional user attribute has been added:

- o Attribute name: mail

- o Value: `oracleadmin@vordel.com`

The Attribute value is “mail”. This should return the value “`oracleadmin@vordel.com`” when the message is processed by the Gateway.

The search options above are using the base criteria of the directory structure as far down as the Common Name Object: Users

The Query syntax used can also be validated by performing a search in an LDAP browser using the same string:

(&(objectclass=inetOrgPerson)(cn=orcladmin,cn=Users, dc=qa, dc=vordel, dc=com))

The Retrieve from Directory Server configuration:

Configure "Retrieve from Directory Server"

Retrieve Attributes from Directory Server
Configure retrieval of attributes from an LDAP directory.

Name: Retrieve from Directory Server

LDAP Directory: OID11g

Retrieve unique user identity

From message attribute authentication.subject.id

From LDAP search [Configure Directory Search](#)

Retrieve Attributes:

Base Criteria: cn=Users, dc=qa, dc=vordel, dc=com

Search Filter: (&(objectclass=inetOrgPerson)(cn=\${authentication.subject.id}))

Search Scope: Object level One level Sub-tree

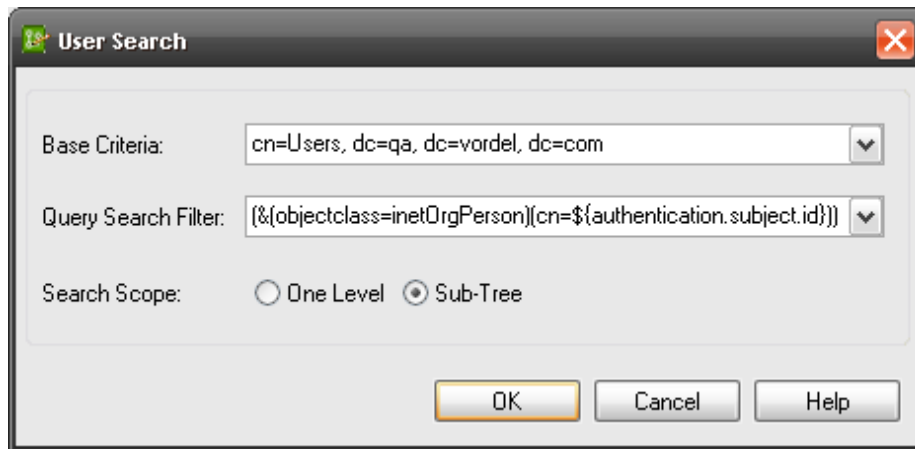
Unique result

Attribute Name
mail

Add Edit Delete

Help < Back Next > Finish Cancel

The "Configure Directory Search" configuration screen if the "From LDAP search" option is used instead of the 'From Message Attribute' section



STEP 3: Refresh Gateway Configuration

- Refresh the Gateway by pressing the “F6” key or select “Settings” located in the top menu of Policy Studio and click on “Deploy F6”

STEP 4: Test the configuration in OEG Service Explorer

- Start OEG Service Explorer by running “serviceexplorer.exe” (win32) or “serviceexplorer.sh” (UNIX) located in the OEG Service Explorer root directory.
- Load a message request
- Click on “Request Settings” on the drop down list on the green “Send Request” button
- Make sure that the URL is set correctly. In this case it will be `http://localhost:8080/oid`
- Click on the “Security” tab followed by the “HTTP Authentication” tab
- Select “HTTP Basic” and enter the Username and Password of the user that will be authenticated via the Oracle OID server.
- User Credentials used for this guide:
 - o User: orcladmin
 - o Password: vordel12
- Click on “Run”

A snippet from the trace console showing the retrieved attribute:

```
-----
DEBUG 14:19:36:322 [27e4] run filter [HTTP Basic] {
```

```
DEBUG 14:19:36:322 [27e4] LDAPRepository.checkCredentials: Check user via LDAP
DEBUG 14:19:36:322 [27e4] LDAPRepository.getQueryResultsFromCache.
Key=orcladmin::cn=Users, dc=qa, dc=vordel, dc=com:inetOrgPerson:cn:cn
DEBUG 14:19:36:322 [27e4] LDAPRepository.getQueryResultsFromCache. Expired
DEBUG 14:19:36:322 [27e4] LDAP search to be run:
(&(objectClass=inetOrgPerson)(cn=orcladmin))
DEBUG 14:19:36:322 [27e4] adding the additional jndi properties: {}
INFO 14:19:36:337 [27e4] cache
com.vordel.common ldap.LdapLookup$ContextCache@f88377 grows to 1
DEBUG 14:19:36:431 [27e4] }
DEBUG 14:19:36:431 [27e4] loginUser - dname to be used: cn=orcladmin,cn=Users,
dc=qa, dc=vordel, dc=com
DEBUG 14:19:36:431 [27e4] Attempting to authenticate the user:
cn=orcladmin,cn=Users, dc=qa, dc=vordel, dc=com
DEBUG 14:19:36:494 [27e4] The authenticated the user: cn=orcladmin,cn=Users, dc=qa,
dc=vordel, dc=com
DEBUG 14:19:36:494 [27e4] LDAPRepository.addQueryResultsFromCache.
Key=orcladmin::cn=Users, dc=qa, dc=vordel, dc=com:inetOrgPerson:cn:cn
DEBUG 14:19:36:494 [27e4] UsernameAuthN.getResponse: Mapped 'orcladmin' to
'orcladmin'. Format=Username
DEBUG 14:19:36:494 [27e4] } = 1, in 172 milliseconds
DEBUG 14:19:36:494 [27e4] run filter [Retrieve from Directory Server] {
DEBUG 14:19:36:494 [27e4] LookupHandler.process: userIdentity: orcladmin
DEBUG 14:19:36:494 [27e4] Looking up user cache with the key: orcladmin
DEBUG 14:19:36:494 [27e4] User's attribute from cache: (null)
DEBUG 14:19:36:494 [27e4] No attributes for user in cache so do lookup
DEBUG 14:19:36:494 [27e4] The user identity whose attributes are looked for is
[orcladmin]
DEBUG 14:19:36:494 [27e4] Searching for a attributes with base [cn=Users, dc=qa,
dc=vordel, dc=com] and filter [(&(objectclass=inetOrgPerson)(cn=orcladmin))]
DEBUG 14:19:36:494 [27e4] adding the additional jndi properties: {}
INFO 14:19:36:540 [27e4] cache
com.vordel.common ldap.LdapLookup$ContextCache@1bc1fe5 grows to 1
DEBUG 14:19:36:619 [27e4] Retrieving attributes for the result cn=orcladmin
DEBUG 14:19:36:619 [27e4] LdapLookup.addToAttributeHashMap: attribute=[mail]
value=[oracleadmin@vordel.com]
DEBUG 14:19:36:619 [27e4] LdapAttrLookupHandler.getAttributes:
Attributes={mail=key=[mail] name=[mail] values=[oracleadmin@vordel.com]
namespace=[###nonamespace###] namespaceForAssertion=[urn:vord
el:attribute:1.0] useForAssertion=[true]}
```

```

DEBUG 14:19:36:619 [27e4] Retrieved attributes: 1===> key=[mail] name=[mail]
values=[oracleadmin@vordel.com] namespace=[##nonamespace##]
namespaceForAssertion=[urn:vordel:attribute:1.0] useForAssertion=[true]
DEBUG 14:19:36:619 [27e4] Copy user attribute [mail] value=[oracleadmin@vordel.com]
to message attribute [user.mail]
-----

```

The “mail” attribute is the attribute that was retrieved by the “Retrieve from Directory” filter. At runtime when the attribute is retrieved it is pre-pended with “user” to identify this attribute as a user specific attribute. Because only one attribute was retrieved from the retrieval filter the dynamically generated attribute name (i.e. ‘user.mail’) is appended with the index value starting at 1, i.e. user.mail.1. In cases where the “mail” attribute is a multi-valued attribute then multiple values will be returned by the attribute retrieval filter. In such cases each attribute will be stored incrementally, for example, user.mail.1, user.mail.2, user.mail.3, and so on.

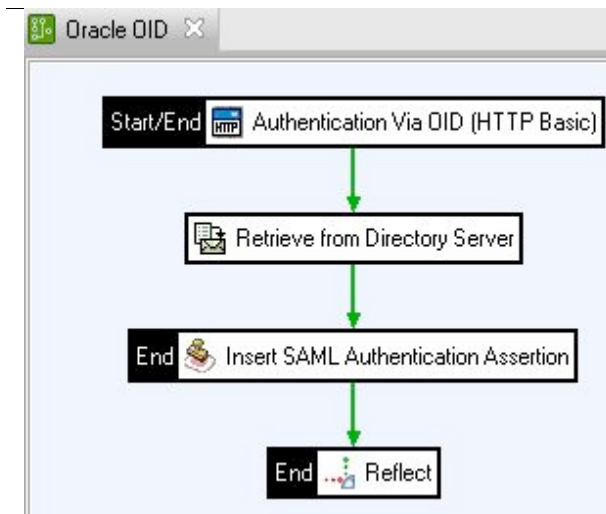
5. Adding an Insert SAML Authentication Assertion filter

A SAML Authentication Assertion filter can also be added to the policy flow to provide single sign on capability. Below it will be demonstrated to do this.

STEP:1 Add an “Insert SAML Authentication Assertion” filter

- Click on the “Oracle OID” policy
- From the “Authentication” group in the filter palette drag an “Insert SAML Authentication Assertion” filter to the circuit. For configuration see STEP1 below
- The flow of the filters will be, HTTP Basic->Retrieve from Directory Server->Insert SAML Authentication Assertion->Reflect all connected with success path connectors

The modified Policy after having added the ‘Insert SAML Authentication Assertion’ filter:



STEP 2: Configuring the 'Insert SAML Authentication Assertion' filter:

- Add an "Insert SAML Authentication Assertion" filter located in the "Authentication" filter category.
- Configure the filter as follows:
 - Expiry Date: Set to any desired value.
 - SOAP Actor/Role: Choose "Current Actor/Role Only" from the drop down list.
 - SAML Version: Select the version of SAML required. Options are 1.0, 1.1 or 1.2.
 - For "Issuer Name" select the desired issuer from the drop down field.
 - Click on the "Confirmation Method" tab and select the desired confirmation method from the list. Please click on the "Help" button for more information on the different options.
 - The "Advanced" tab contains more options in regards to layout, using Security Token Reference etc.
- Click on "Finish" once the filter is configured as desired.

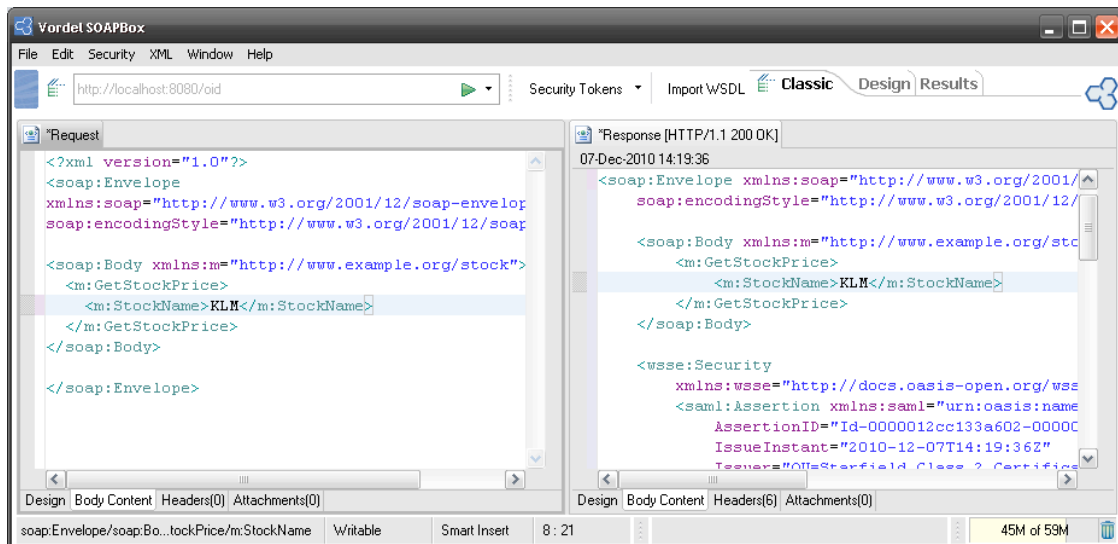
STEP 3: Test the configuration in OEG Service Explorer

With the 'Insert SAML Authentication Assertion' filter added to the policy OEG Service Explorer will be used to verify the configuration.

Set up a message in OEG Service Explorer

- Start OEG Service Explorer by running “serviceexplorer.exe” (win32) or “serviceexplorer.sh” (UNIX) located in the OEG Service Explorer root directory.
- Load a message request
- Click on Settings just above the Send Request button
- Then Click on Connection Settings
- Make sure that the URL is set correctly. In this case it will be http://localhost:8080/oid
- Click on “OK”
- Click on the HTTP Authentication tab followed by the HTTP Basic tab
- Enter the Username and Password of the user that will be Authenticated via LDAP
- User Credentials:
 - o Username: orcladmin
 - o Password: vordel12
- Click on Finish

The result in OEG Service Explorer after the SAML Authentication has been inserted:



6. Conclusion

This document is a simplistic demonstration on how to configure the Gateway to authorize users residing in an Oracle Internet Directory.

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>.

7. Appendix

Creating a secure connection using SSL to Active Directory:

The Certificate Authority that issued the LDAP Server certificate is required by Gateway keystore.

Once the CA certificate is obtained it is necessary to import it into the Gateway JAVA keystore.

- In Policy Studio click on the “Certificates” module
- Click on “Certificates” then on the right hand side click on “Create” then on “Import Certificate”
- Browse to the LDAP Certificate and click on “Open”
- Tick the “Use Subject” box next to the “Alias” field and click on “OK”
- The LDAP server certificate is now imported into the Gateway Certificate store
- It now needs to be added to the JAVA keystore
- Click on “Keystore” in the “Certificate” window then on the browse button
- Browse to the following file:
 - Gateway_Dir/win32/jre/lib/security/cacerts (Windows)
 - Gateway_Dir/posix/jre/lib/security/cacerts (Linux/Unix)
- Click on “Open” and enter the Keystore password. Default password is: changeit
- Ensure that the LDAP connection configuration has been modified appropriately by checking the “SSL Enabled” checkbox and to ensure the LDAP URL is set to use ldaps and the configured secured port for example:
ldaps://oid11g.qa.vordel.com:3361

See screenshot below:

Configure LDAP Server

Name:

URL:

Cache Timeout:

Cache Size:

Authenticate LDAP Requests

Type:

User Name:

Password:

Realm:

SSL Enabled

Additional JNDI properties:

Name	Value



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.