



An Oracle White Paper
June 2011

Oracle Directory Server Enterprise Edition 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.

1. Introduction	4
1.1. Purpose.....	4
1.2. About Oracle Directory Server Enterprise Edition.....	5
1.3. Setup Used for this Guide:.....	5
2. Oracle DSEE Details.....	5
2.1. Directory Structure	5
2.2. Connection Details for Gateway to ODSEE.....	7
3. Authenticate User with HTTP Basic HTTP Filter	7
STEP 1: Create Policy to Authenticate User in LDAP directory.....	7
STEP 2: Create a new relative path for the Policy.....	13
STEP 3: Ensure policies are updated on the Gateway.....	14
STEP 4: Test the configuration in OEG Service Explorer	14
4. Adding a Retrieve from Directory Server filter	16
STEP1: Modify the Policy to include an 'Retrieve from Directory Server' filter	16
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	23
STEP:1 Add an "Insert SAML Authentication Assertion" filter	23
STEP 2: Configuring the 'Insert SAML Authentication Assertion' filter:	24
STEP 3: Test the configuration in OEG Service Explorer.....	24
6. Conclusion	25
7. Appendix	26

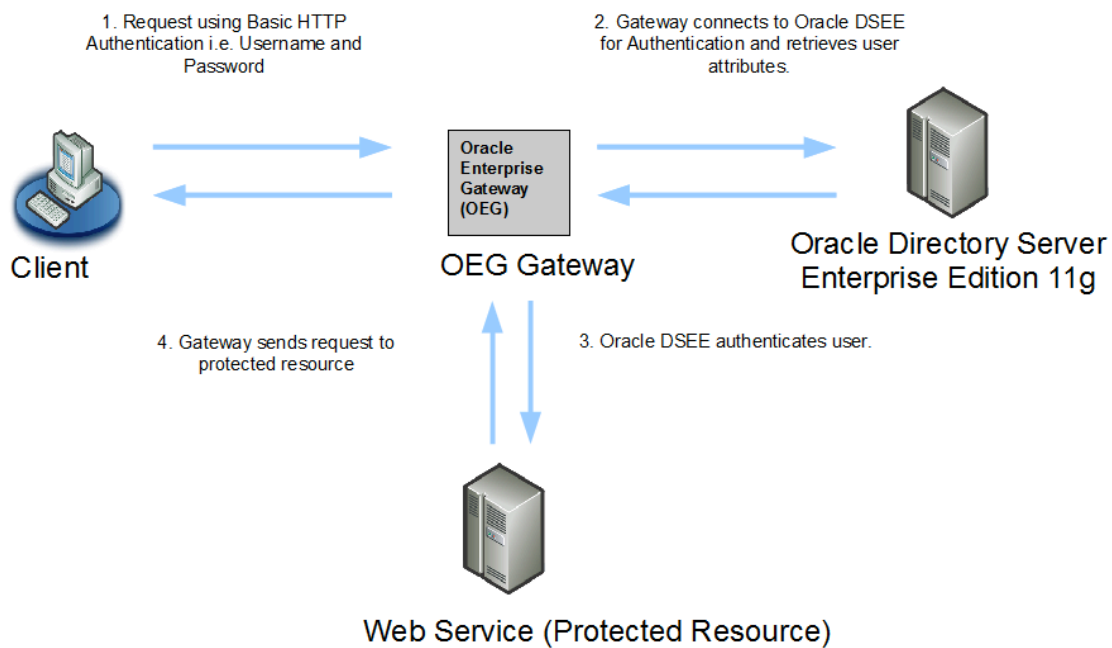
1. Introduction

1.1. Purpose

This document describes how to configure the Gateway to authenticate via an Oracle Directory Server Enterprise Edition 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 DSEE directory.
- Upon successful authentication the Gateway will be configured to extract attributes belonging to this user from the Oracle DSEE directory.
- A SAML Authentication Assertion will also be added to demonstrate Single Sign On capability.

Flow of request:



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

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

1.2. About Oracle Directory Server Enterprise Edition

Oracle Directory Server Enterprise Edition (formerly SUN Directory Server Enterprise Edition) is the best known directory server with proven large deployments in carrier and enterprise environments. It is also the most supported directory by ISVs, so it is ideal for heterogeneous environments. ODSEE provides a core directory service with embedded database, directory proxy, Active Directory (AD) synchronization and a Web administration console.

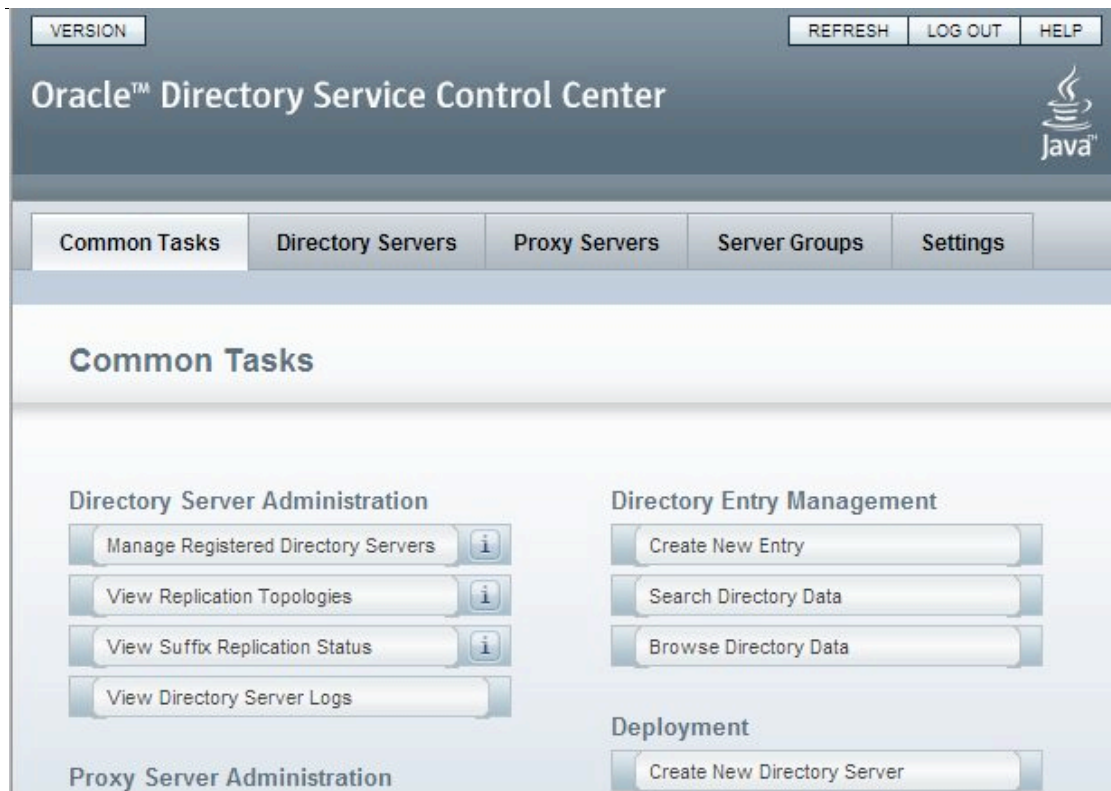
1.3. Setup Used for this Guide:

- Oracle Enterprise Gateway 11.1.1.x
- Oracle Directory Server Enterprise Edition 11g
- LDAP Browser

2. Oracle DSEE Details

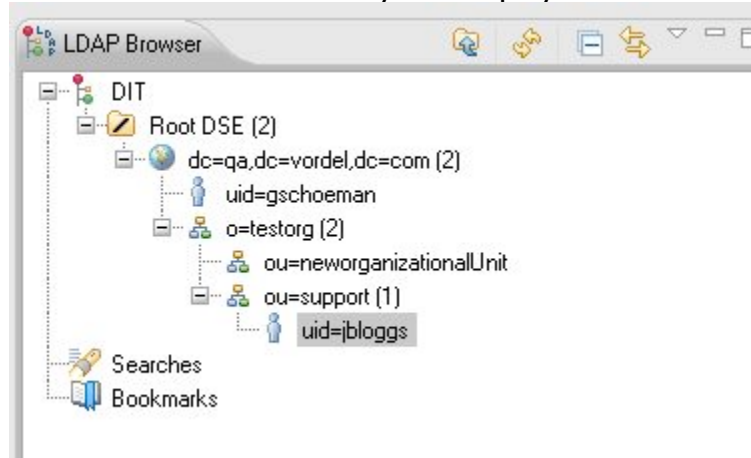
2.1. Directory Structure

Oracle Directory Server Enterprise Edition can be managed via the Oracle Directory Service Control Centre.



Click on the "Browse Directory Data" tab to view entries in the directory. For more information on Oracle Directory Service Control Center please refer to the Oracle documentation.

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



User in ODSEE used for Authentication

DN: uid=jbloggs,ou=support,o=testorg,dc=qa,dc=vordel,dc=com

Attribute Description	Value
objectClass	inetOrgPerson (structural)
objectClass	organizationalPerson (structural)
objectClass	person (structural)
objectClass	top (abstract)
cn	Joe Bloggs
sn	Bloggs
departmentNumber	007
l	Dublin
o	Vordel
uid	jbloggs
userPassword	SSHA encrypted password
createTimestamp	18-Jan-2011 16:29:17 GMT (20110118162917Z)
creatorsName	cn=admin,cn=administrators,cn=dsc
entrydn	uid=jbloggs,ou=support,o=testorg,dc=qa,dc=vordel,dc=com
entryid	7
hasSubordinates	FALSE
modifiersName	cn=admin,cn=administrators,cn=dsc
modifyTimestamp	18-Jan-2011 16:38:20 GMT (20110118163820Z)
nsUniqueId	06230981-232011e0-80a7fc6f-9bcec190
numSubordinates	0
parentid	6
subschemaSubentry	cn=schema

2.2. Connection Details for Gateway to ODSEE

The connection details for Oracle ODSEE are:

- LDAP URL: ldap://oracle-dsee.qa.vordel.com:1389
- user: cn=admin,cn=administrators,cn=dsc
- 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 DSEE. 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.
- Double click on the Gateway process listed to open the configuration workspace.
- Click on the "External Connections" module.
 - Right Click on "LDAP Connections" and Click "Add a LDAP Connection"
 - Name: For this guide "ODSEE" is used
 - Enter the Connection URL: ldap://oracle-dsee.qa.vordel.com:1389
 - For the "Type" dropdown box select "Simple"
 - Enter the credentials to connect to the LDAP directory
 - o Username: cn=admin,cn=administrators,cn=dsc
 - o Password: vordel12
 - Realm: Can be left blank
- Click on "Test Connection" to verify that the connection to the LDAP database has been configured successfully. NOTE: connectivity here only applies from where Policy Studio is installed and could be different on the machine that the Gateway is installed on.
 - 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 "ODSEE" is used
- LDAP Store: For the "LDAP Directory" choose the previously created LDAP connection "ODSEE" from the drop down list
 - Specify the "User Search Conditions"
 - For this guide the following details are used based on the Directory information above:
 - o Base Criteria: 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: Entry Domain Name

- Authorization Attribute: cn
- Authorization Attribute Format: User Name
 - Click "OK"
- The new LDAP Repository should now be visible in the "LDAP Repositories" Tree
- Click on the "Policies" module and then right click on the "Policies" tree on the left hand side of Policy Studio
- Click "Add Policy" and name the Policy "ODSEE Auth"
- Click on the Policy and drag a "HTTP Basic" filter located in the "Authentication" filter category 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 "ODSEE" 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.

- Base Criteria: dc=qa,dc=vordel,dc=com
- User Class: "inetOrgPerson" LDAP Class (from the drop down list)
- 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 "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.

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"

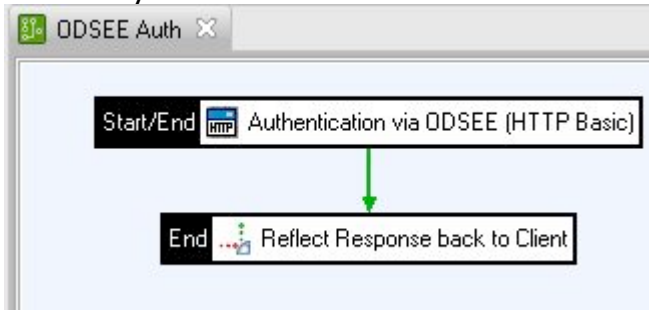
For this guide this section has been configured as follows:

1. Login Authentication Attribute: Leave blank
2. Authorization Attribute: cn
3. 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 ODSEE has been selected and setting this to be the value used for authorization.

The Policy will look as follows:



The configuration of the HTTP Basic filter as described above:

The screenshot shows the configuration dialog for "Authentication via ODSEE (HTTP Basic)". The dialog has the following fields and options:

- Name:** Authentication via ODSEE (HTTP Basic)
- Credential Format:** User Name
- Allow client challenge
- Remove HTTP authentication header
- Repository Name:** ODSEE

At the bottom, there are five buttons: Help, < Back, Next >, Finish, and Cancel.

The Connection settings window:

Configure LDAP Server

Name: ODSEE

URL: ldap://oracle-dsee.qa.vordel.com:1389

Cache Timeout: 300000

Cache Size: 8

Authenticate LDAP Requests

Type: Simple

User Name: cn=admin,cn=administrators,cn=dsc

Password: *****

Realm:

SSL Enabled

Test Connection

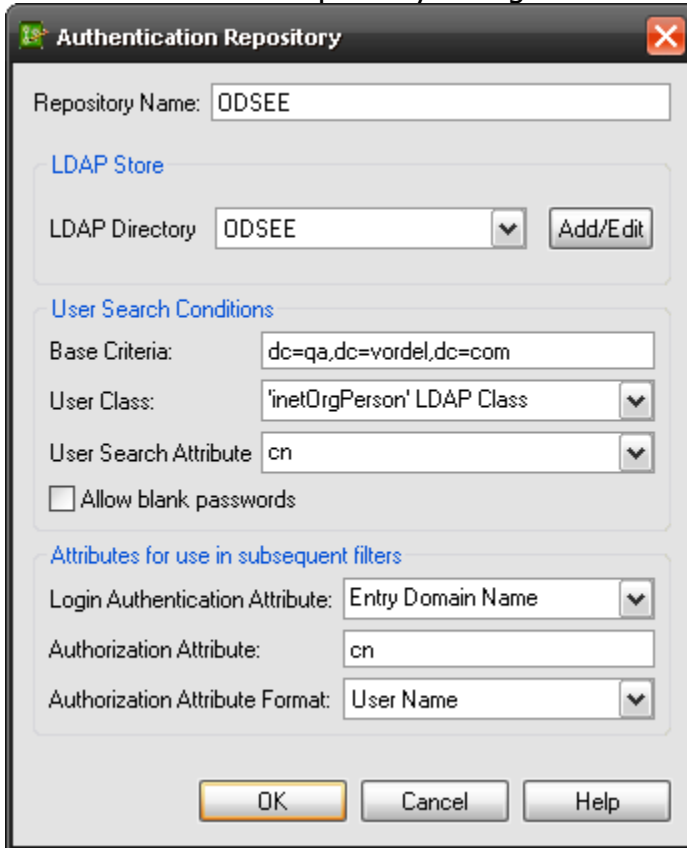
Additional JNDI properties:

Name	Value

Add Edit Delete

OK Cancel Help

The Authentication Repository configuration:

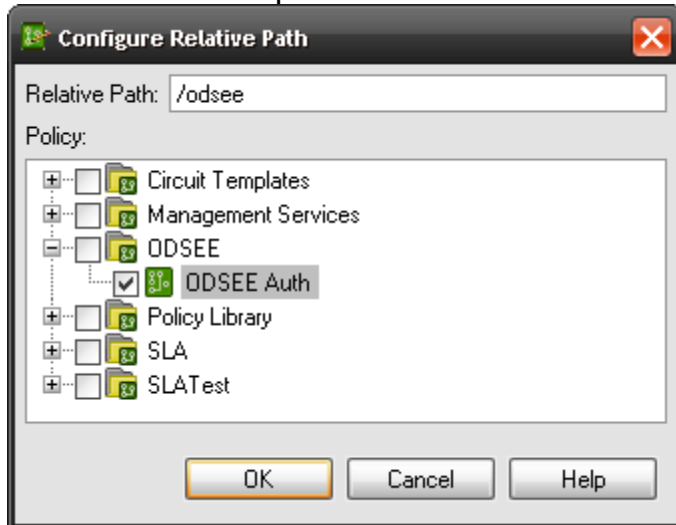


The screenshot shows the "Authentication Repository" configuration dialog box. The "Repository Name" field is set to "ODSEE". Under the "LDAP Store" section, the "LDAP Directory" dropdown is set to "ODSEE" with an "Add/Edit" button next to it. The "User Search Conditions" section includes "Base Criteria" set to "dc=qa,dc=vordel,dc=com", "User Class" set to "'inetOrgPerson' LDAP Class", and "User Search Attribute" set to "cn". There is an unchecked checkbox for "Allow blank passwords". The "Attributes for use in subsequent filters" section has "Login Authentication Attribute" set to "Entry Domain Name", "Authorization Attribute" set to "cn", and "Authorization Attribute Format" set to "User Name". At the bottom are "OK", "Cancel", and "Help" buttons.

STEP 2: Create a new relative path for the Policy

1. Click on the "Services" module in Policy Studio.
2. Expand "Processes", "Gateway" and right click on the "Default Services".
3. Select "Add Relative Path" and enter: /odsee
4. Map the path to the policy titled "ODSSE Auth"
5. 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 "OEG Service Explorer.exe" (win32) or "OEG Service Explorer.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/odsee`
- 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 DSEE server.
- User Credentials used for this demonstration is for a user residing in the ODSEE directory:
 - o User: Joe Bloggs
 - o Password: test

- Click on "Run"
- The request will be sent to the Gateway which will connect to Oracle DSEE to authenticate the user "Joe Bloggs"

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

```
-----  
DEBUG 21:16:17:266 [1328] Incoming HTTP request:  
method=POST, host=(unset), port=(unset), path=/od  
see, query=(unset), version=1.1  
DEBUG 21:16:17:266 [1328] handle type text/xml with  
factory class com.vordel.mime.XMLBody$Factory  
DEBUG 21:16:17:266 [1328] run circuit "ODSEE Auth"...  
DEBUG 21:16:17:266 [1328] run filter [HTTP Basic] {  
DEBUG 21:16:17:266 [1328]  
LDAPRepository.checkCredentials: Check user via LDAP  
DEBUG 21:16:17:266 [1328]  
LDAPRepository.getQueryResultsFromCache. Key=Joe  
Bloggs::ou=support,  
o=testorg,dc=qa,dc=vordel,dc=com:inetOrgPerson:cn:entrydn:  
cn  
DEBUG 21:16:17:282 [1328] LDAP search to be run:  
((&(objectClass=inetOrgPerson)(cn=Joe Bloggs))  
DEBUG 21:16:17:297 [1328] adding the additional jndi  
properties: {}  
DEBUG 21:16:17:500 [1328] cache  
com.vordel.common.ldap.LdapLookup$ContextCache@1aa5882  
grows to 1  
DEBUG 21:16:17:875 [1328] }  
DEBUG 21:16:17:875 [1328] loginUser - dname to be  
used: uid=jbloggs,ou=support,o=testorg,dc=qa  
,dc=vordel,dc=com  
DEBUG 21:16:17:875 [1328] Attempting to authenticate  
the user:  
uid=jbloggs,ou=support,o=testorg,dc=qa,dc=vordel,dc=com  
DEBUG 21:16:18:032 [1328] The authenticated the  
user:  
uid=jbloggs,ou=support,o=testorg,dc=qa,dc=vordel,dc=com  
DEBUG 21:16:18:032 [1328]  
LDAPRepository.addQueryResultsFromCache. Key=Joe  
Bloggs::ou=support,
```

```

o=testorg,dc=qa,dc=vordel,dc=com:inetOrgPerson:cn:entrydn:
cn
DEBUG    21:16:18:032 [1328]      UsernameAuthN.getResponse:
Mapped 'Joe Bloggs' to 'Joe Bloggs'. Format=Username
DEBUG    21:16:18:032 [1328] } = 1, in 766 milliseconds
DEBUG    21:16:18:032 [1328] run filter [Reflect] {
DEBUG    21:16:18:032 [1328] } = 1, in 0 milliseconds
DEBUG    21:16:18:032 [1328] ... "ODSEE Auth" complete.

```

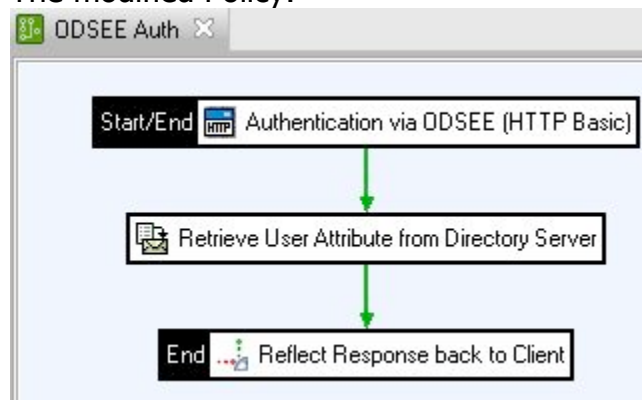
4. Adding a Retrieve from Directory Server filter

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 "ODSEE Auth" 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 1: 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

- 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.
- 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: dc=qa,dc=vordel,dc=com

- Search Filter:

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

Retrieve Unique User Identity from LDAP Search:

- Search Scope: Sub Tree is selected

- Query Search Filter:

```
(amp(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:

- Attribute name: departmentNumber
- Value: 007

The Attribute value is "departmentNumber". This should return the value "007" 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)(
 uid=jbloggs,ou=support,o=testorg,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 User Attribute from Directory Server

LDAP Directory: ODSEE

Retrieve unique user identity

From message attribute: authentication.subject.id

From LDAP search: Configure Directory Search

Retrieve Attributes:

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

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

Search Scope: Object level One level Sub-tree

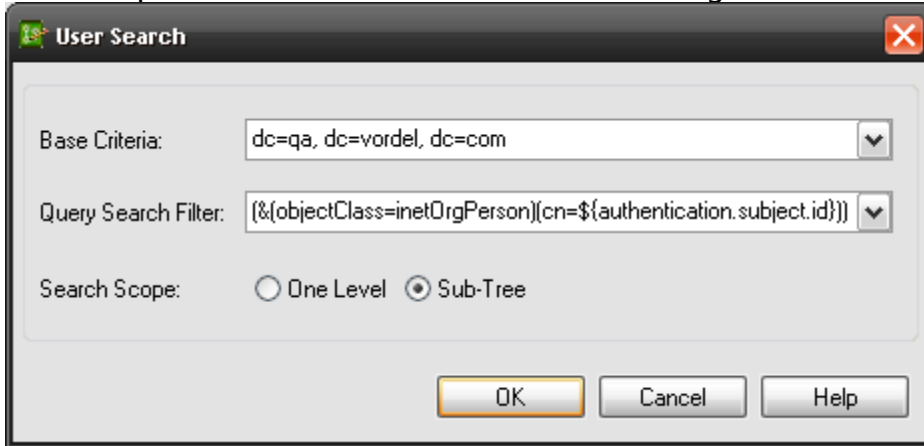
Unique result

Attribute Name
departmentNumber

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 2: 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 3: Test the configuration in OEG Service Explorer

- Start OEG Service Explorer by running "OEG Service Explorer.exe" (win32) or "OEG Service Explorer.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/odsee`
- 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 DSEE server.
- User Credentials used for this guide:
 - o User: Joe Bloggs
 - o Password: test
- Click on "Run"

A snippet from the trace console showing the retrieved attribute:

```
-----
DEBUG 21:26:01:564 [1d90] Incoming HTTP request:
method=POST, host=(unset), port=(unset), path=/odsee,
query=(unset), version=1.1
DEBUG 21:26:01:564 [1d90] handle type text/xml with
factory class com.vordel.mime.XMLBody$Factory
DEBUG 21:26:01:564 [1d90] run circuit "ODSEE Auth"...
DEBUG 21:26:01:564 [1d90] run filter [Authentication via
ODSEE (HTTP Basic)] {
DEBUG 21:26:01:564 [1d90]
LDAPRepository.checkCredentials: Check user via LDAP
DEBUG 21:26:01:564 [1d90]
LDAPRepository.getQueryResultsFromCache. Key=Joe
Bloggs::dc=qa,dc=vo
rdel,dc=com:inetOrgPerson:cn:entrydn:cn
DEBUG 21:26:01:564 [1d90] LDAP search to be run:
(&(objectClass=inetOrgPerson)(cn=Joe Bloggs))
DEBUG 21:26:01:564 [1d90] adding the additional jndi
properties: {}
DEBUG 21:26:01:564 [1d90] cache
com.vordel.common.ldap.LdapLookup$ContextCache@d2883b
grows to 1
DEBUG 21:26:01:564 [1d90] }
DEBUG 21:26:01:564 [1d90] loginUser - dname to be
used:
uid=jbloggs,ou=support,o=testorg,dc=qa,dc=vordel,dc=com
DEBUG 21:26:01:564 [1d90] Attempting to authenticate
the user:
uid=jbloggs,ou=support,o=testorg,dc=qa,dc=vordel,dc=com
DEBUG 21:26:01:564 [1d90] The authenticated the
user:
uid=jbloggs,ou=support,o=testorg,dc=qa,dc=vordel,dc=com
DEBUG 21:26:01:564 [1d90]
LDAPRepository.addQueryResultsFromCache. Key=Joe
Bloggs::dc=qa,dc=vo
rdel,dc=com:inetOrgPerson:cn:entrydn:cn
DEBUG 21:26:01:564 [1d90] UsernameAuthN.getResponse:
Mapped 'Joe Bloggs' to 'Joe Bloggs'. Format=Username
DEBUG 21:26:01:564 [1d90] } = 1, in 0 milliseconds
DEBUG 21:26:01:564 [1d90] run filter [Retrieve User
Attribute from Directory Server] {
DEBUG 21:26:01:564 [1d90] LookupHandler.process:
userIdentity: Joe Bloggs
```

```
DEBUG 21:26:01:564 [1d90] Looking up user cache with
the key: Joe Bloggs
DEBUG 21:26:01:564 [1d90] User's attribute from
cache: (null)
DEBUG 21:26:01:564 [1d90] No attributes for user in
cache so do lookup
DEBUG 21:26:01:564 [1d90] The user identity whose
attributes are looked for is [Joe Bloggs]
DEBUG 21:26:01:564 [1d90] Searching for a attributes
with base [dc=qa, dc=vordel, dc=com] and filter
[(&(objectclass=inetOrgPerson)(cn=Joe Bloggs))]
DEBUG 21:26:01:564 [1d90] adding the additional jndi
properties: {}
DEBUG 21:26:01:579 [1d90] cache
com.vordel.common.ldap.LdapLookup$ContextCache@1781288
grows to 1
DEBUG 21:26:01:595 [1d90] Retrieving attributes for
the result uid=jbloggs,ou=support,o=testorg
DEBUG 21:26:01:595 [1d90]
LdapLookup.addToAttributeHashMap:
attribute=[departmentNumber] value=[007]
DEBUG 21:26:01:595 [1d90]
LdapAttrLookupHandler.getAttributes:
Attributes={departmentNumber=key=[departmentNumber]
name=[departmentNumber] values=[007]
namespace=[##nonamespace##]
namespaceForAssertion=[urn:vordel:attribute:1.0]useForAsse
rtion=[true]}
DEBUG 21:26:01:595 [1d90] Retrieved attributes:
l==> key=[departmentNumber] name=[departmentNumber]
values=[007]
namespace=[##nonamespace##]namespaceForAssertion=[urn:vord
el:attribute:1.0] use ForAssertion=[true]
DEBUG 21:26:01:595 [1d90] Copy user attribute
[departmentNumber] value=[007] to message attribute
[user.departmentNumber]
DEBUG 21:26:01:595 [1d90] } = 1, in 31 milliseconds
DEBUG 21:26:01:595 [1d90] run filter [Reflect Response
back to Client] {
DEBUG 21:26:01:595 [1d90] } = 1, in 0 milliseconds
DEBUG 21:26:01:595 [1d90] ..."ODSEE Auth" complete.
```

The "departmentNumber" 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.departmentNumber`) is appended with the index value starting at 1, i.e. user.mail.1. In cases where the "departmentNumber" 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.departmentNumber.1, user.departmentNumber.2, user.departmentNumber.3 and so on.

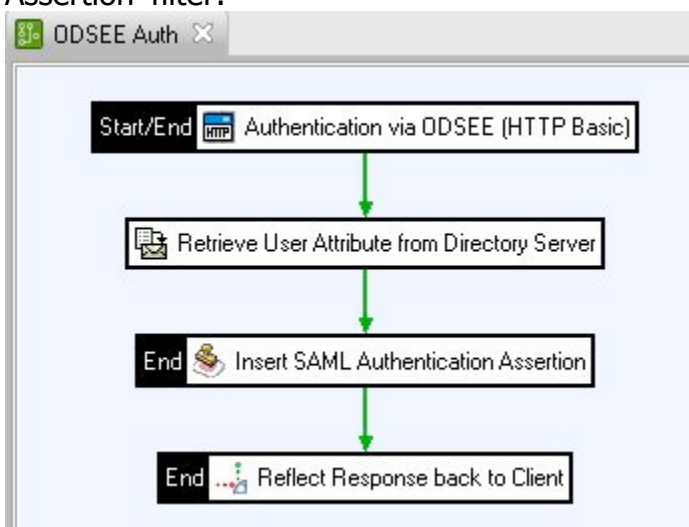
5. Adding an Insert SAML Authentication Assertion filter

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 "ODSEE Auth" 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 "OEG Service Explorer.exe" (win32) or "OEG Service Explorer.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/odsee`
- 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: Joe Bloggs
 - o Password: test
- Click on Finish

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://oracle-dsee.qa.vordel.com:1636

Note: In order to use the "Test Connection" in the LDAP configuration window the certificate of ODSEE also needs to be added to the java keystore of Policy Studio. Use the same method as detailed above for the Gateway except open keystore file location:

PolicyStudio_Dir/jre/lib/security/cacerts (Windows and Linux)

LDAPS configuration in the LDAP Configuration Window:

Configure LDAP Server

Name: ODSEE

URL: ldaps://oracle-dsee.qa.vordel.com:1636

Cache Timeout: 300000

Cache Size: 8

Authenticate LDAP Requests

Type: Simple

User Name: cn=admin,cn=administrators,cn=dsc

Password: *****

Realm:

SSL Enabled

Test Connection

Additional JNDI properties:

Name	Value

Add Edit Delete

OK Cancel Help



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.