Oracle Access Manager Integration Guide

Published: May 2013

Applies To

Oracle API Gateway 11.1.2.x
Oracle Access Manager 10g, 11gR1, and 11gR2

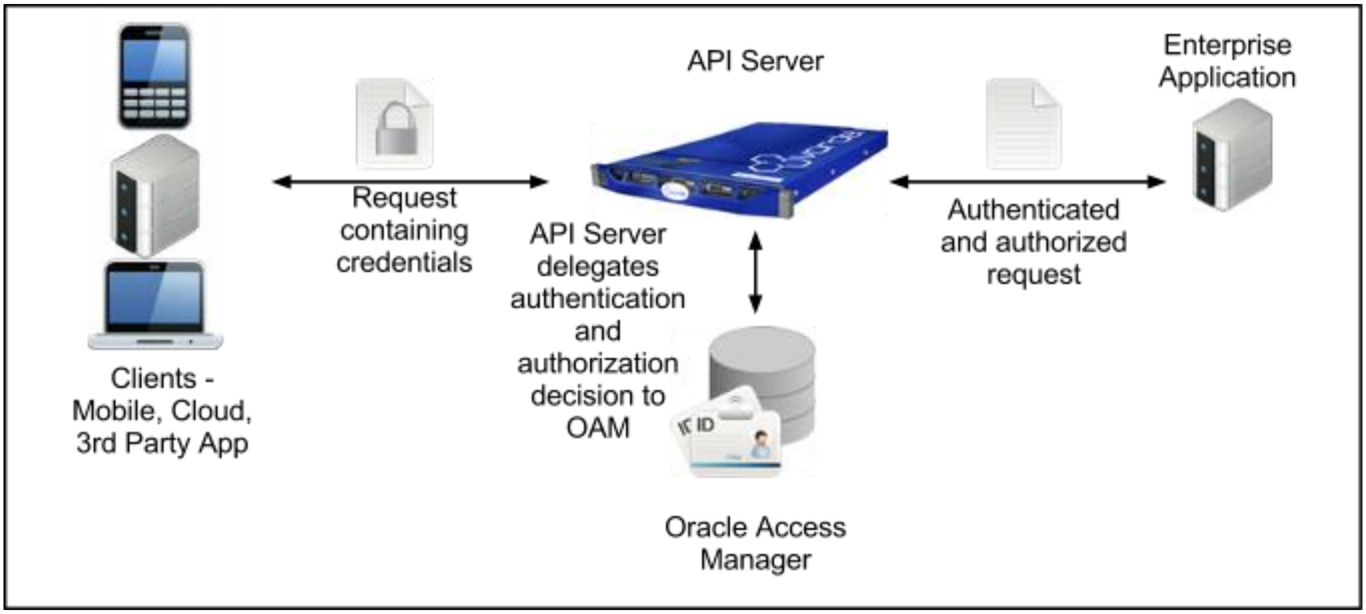
Contents

Introduction

This document describes how to configure the API Gateway to authenticate and authorize user requests against Oracle Access Manager 10g, 11gR1, and 11gR2. This is demonstrated as follows:

1. The API Gateway is configured to authenticate a client against Oracle Access Manager using a username and password.
2. Upon successful authentication, the API Gateway authorizes the user against Oracle Access Manager.

The following overview diagram shows the message flow through the API Server, which authenticates and authorizes a user for a particular resource against Oracle Access Manager before routing the message on to the web service.


API Server
Enterprise Application
Request containing credentials
Authenticated and authorized request
API Server delegates authentication and authorization decision to OAM
Clients - Mobile, Cloud, 3rd Party App
ID
Oracle Access Manager

Deployment of API Gateway with OAM

Prerequisites

API Gateway:

You must have installed API Server 11.1.2.x.

AccessServer SDK:

The AccessServer SDK (ASDK) must be installed on the machine running the API Server. The API Server can work with both ASDK 10g and 11g. Depending on what version of the ASDK you intend to use, refer to the relevant installation instructions below.

Install the Access SDK that is appropriate for your target platform. If you are installing the ASDK on a Vordel Appliance, you should install the Linux version of the ASDK.

Access SDK 10g (10.1.4.3.0):

Download and install ASDK 10g as follows:

1. Download the ASDK from the Oracle website here: http://www.oracle.com/technetwork/middleware/ias/downloads/101401-099957.html
2. Download the "Oracle Access Manager Core Components (10.1.4.3.0)" component.
3. Extract the OAM AccessServer SDK installer archive it and run/extract, depending on your target platform. For example, on Windows the installer is called "Oracle_Access_Manager10_1_4_3_0_Win32_AccessServerSDK.exe".

3. The instructions outlined later in this document will assume that you have installed the ASDK to the default location. On Windows, the default installation path is: `C:\Program Files (x86)\NetPoint\AccessServerSDK.`
   On Linux, the default installation path is: `/opt/netpoint/AccessServerSDK.`

Access SDK 11gR2 (11.1.2.0.0):

The following instructions describe how to install ASDK 11gR2 on Windows:

1. Create a new directory, for example, C:\Oracle\AccessServerSDK11. This directory will be referred to as ASDK_HOME throughout the remainder of this document.
2. Obtain the ASDK archive, called "V33747-01 - Access Manager Access SDK 11.1.2.0.0.zip", and extract its contents to the ASDK_HOME directory.
3. After extracting the files, ensure that you have a jps-config.xml file in the ASDK_HOME\config directory.
4. Similarly, make sure that the ASDK_HOME directory contains the ASDK jars, including oamasdk-api.jar.

Install OAM:

You have installed and configured OAM 10g, 11gR1, or 11gR2 and have started it using the following commands on UNIX-based systems (assuming a weblogic domain of "idm_domain", a server name of "oam_server1", and a hostname of "oam_host").

Start WebLogic using the following commands:

```
# cd ~/middleware/user_projects/domains/idm_domain/bin
# ./startWebLogic.sh
```

You can then start *managed* WebLogic using the following:

```
# cd ~/middleware/user_projects/domains/idm_domain/bin
# ./startManagedWebLogic.sh oam_server1 t3://oam_host:7001
```

Enter the username of your administrator user when prompted:

```
Please enter your username :weblogic
Please enter your password :
```

OAM User:

This guide assumes that a user called “weblogic” is available in OAM. Please refer to the OAM documentation for instructions on how to add a user.

In the Testing the Integration Steps section of this guide, a curl client passes authentication details to the API Server using HTTP basic authentication, which are then used to authenticate to OAM. For this reason and **for testing purposes only**, a weblogic user with the same password as your OAM user must be added to the API Server User Store.

Curl Test Utility:

To test the integration steps outlined in this guide, we will use the Curl testing utility to POST requests to the API Server. It is available from the following URL:
http://curl.haxx.se/download.html

You can, however, use any client capable of sending HTTP POST requests with HTTP basic authentication.

Integration Configuration Steps

OAM Configuration:

The following steps describe how to create a 10g AccessGate or an 11g Webgate (depending on whether you have installed a 10g or an 11g OAM instance) and configure an authentication policy for it using the **OAM Administration Console**. Please refer to the appropriate set of instructions below for your OAM version. For more detailed instructions on OAM configuration, please refer to the OAM documentation.

Configuring a 10g AccessGate with OAM 10g

You can use the web-based **Oracle Access Administration Console** to create the

AccessGate. The web interface is available at the following URL, where [OAM_HOST] refers to the IP or hostname of the machine on which OAM is running:

```
http://[OAM_HOST]/access/oblix
```

Login using the appropriate credentials and complete the following steps.

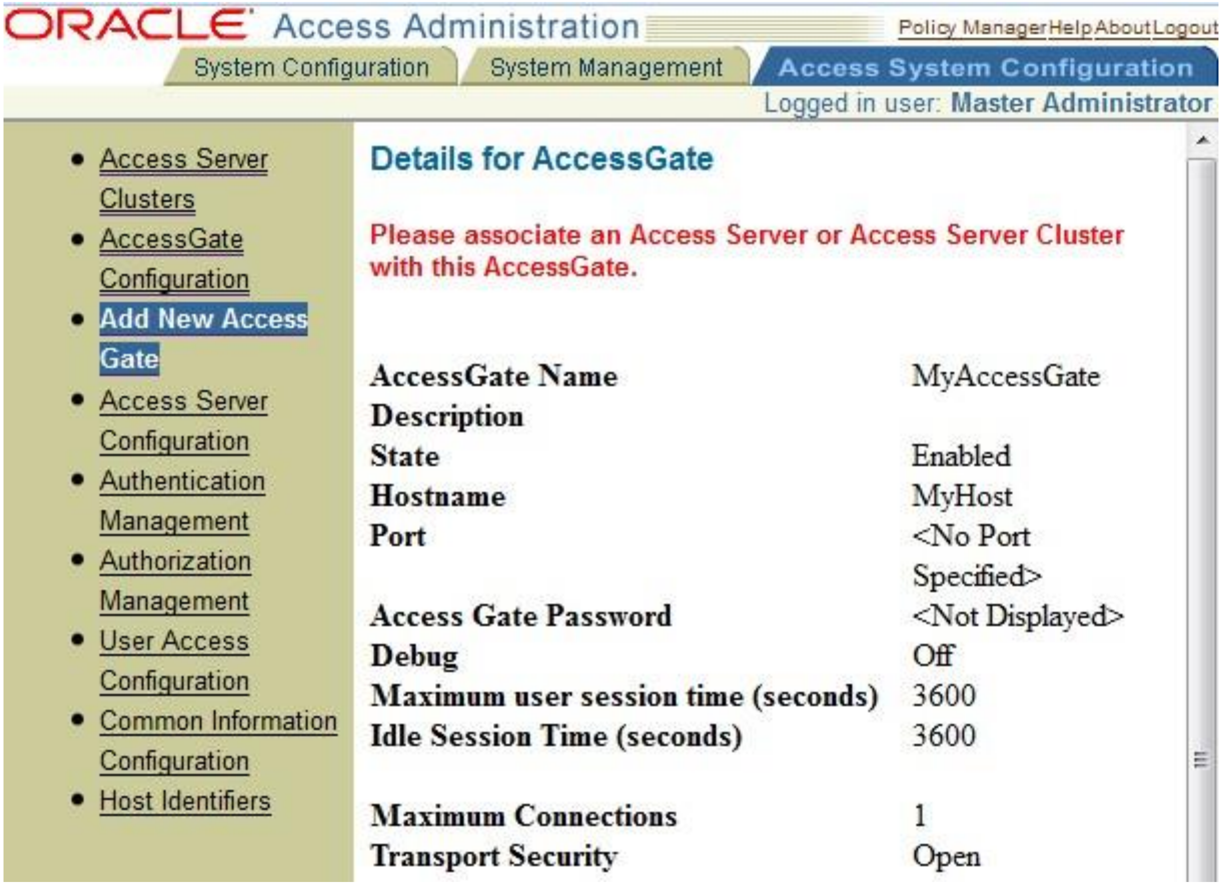
Step 1 - Create the AccessGate

Create a new AccessGate by completing the following steps:

1. After logging in to the console, click on the **Access System Configuration** link.
2. Now open the **Access System Configuration** tab.
3. Click on the **Add New AccessGate** link.
4. Complete the following fields on the **Add New AccessGate** page:
   - **AccessGate Name:** Enter a suitable name for this AccessGate.
   - **Hostname:** Type in the hostname of the AccessGate machine.
   - **AccessGate Password:** Specify a password.
   - **Preferred HTTP Host:** The hostname entered here will appear in all HTTP requests as they attempt to access the protected resource, regardless of the way the hostname was defined by the client in the HTTP request.
5. You can leave all the other fields as default.
6. Click **Save** when you have finished.

Step 2 - Configure a Primary Access Server for the new AccessGate

After saving the new AccessGate details, you will see a warning message at the top of the screen stating that you must "**associate an Access Server or Access Server Cluster with this AccessGate**". The following screenshot shows the warning:


ORACLE Access Administration
Policy Manager
Help
About
Logout
System Configuration
System Management
Access System Configuration
Logged in user: Master Administrator
Access Server Clusters
AccessGate Configuration
Add New Access Gate
Access Server Configuration
Authentication Management
Authorization Management
User Access Configuration
Common Information Configuration
Host Identifiers

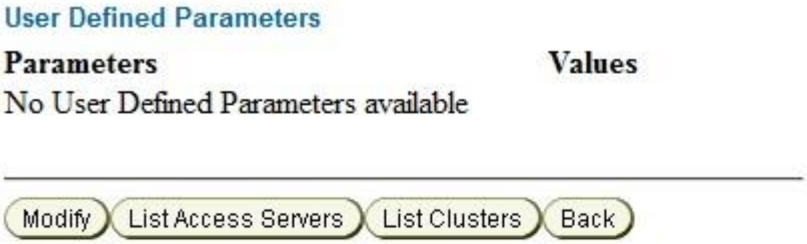
Details for AccessGate

Please associate an Access Server or Access Server Cluster with this AccessGate.

AccessGate Name	MyAccessGate
Description	
State	Enabled
Hostname	MyHost
Port	<No Port Specified>
Access Gate Password	<Not Displayed>
Debug	Off
Maximum user session time (seconds)	3600
Idle Session Time (seconds)	3600
Maximum Connections	1
Transport Security	Open

Details for new AccessGate - you must now associate an Access Server with this new AccessGate

To associate an Access Server with this AccessGate, click on the **List Access Servers** button at the bottom of the page, as shown in the following screenshot:

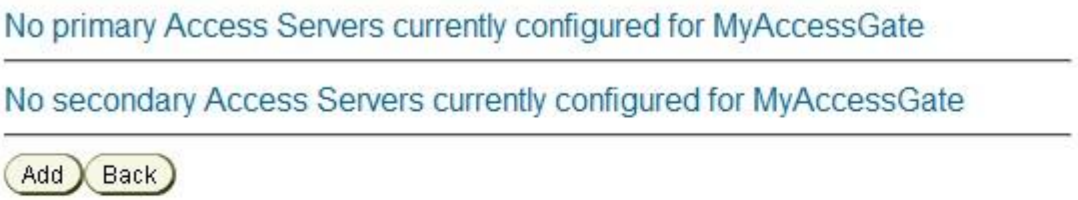


User Defined Parameters

Parameters	Values
No User Defined Parameters available	

The **List Access Servers** button

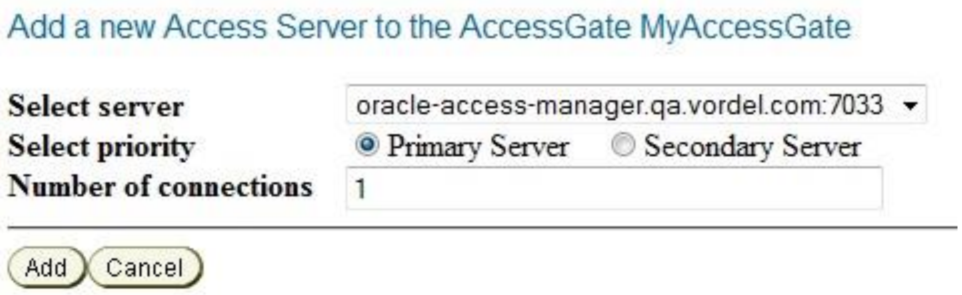
If you have no “primary” or “secondary” Access Servers configured for the new AccessGate, you will see the following screenshot:


No primary Access Servers currently configured for MyAccessGate

No secondary Access Servers currently configured for MyAccessGate

No primary of secondary Access Servers configured for new AccessGate

Click the **Add** button to add a new Access Server to the AccessGate. You can add the Access Server using the following simple form.


Add a new Access Server to the AccessGate MyAccessGate

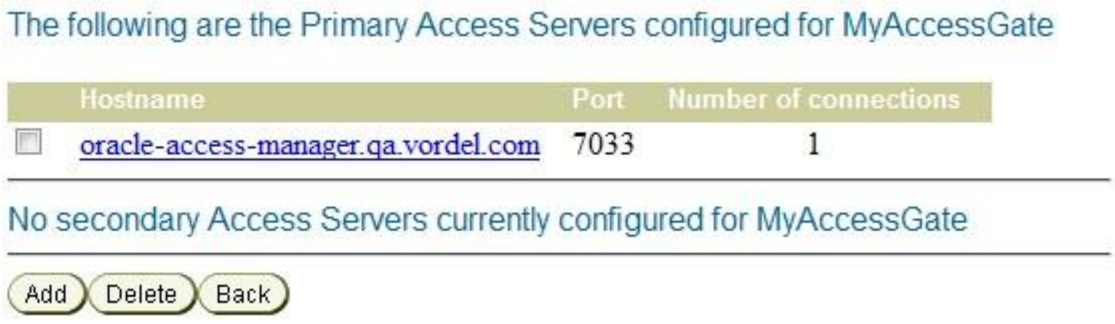
Select server	oracle-access-manager.qa.vordel.com:7033 ▾
Select priority	● Primary Server ○ Secondary Server
Number of connections	1

Add a new Access Server to the new AccessGate

Complete the following steps to add the new Access Server:

1. Using the **Select server** dropdown, select the relevant OAM server from the list.
2. From the **Select priority** options, select **Primary Server**.
3. Specify an appropriate **Number of connections** in the field provided.
4. Click the **Add** button.

The list of Primary Access Servers is now updated to include the newly added Access Server:


The following are the Primary Access Servers configured for MyAccessGate

	Hostname	Port	Number of connections
□	oracle-access-manager.qa.vordel.com	7033	1

No secondary Access Servers currently configured for MyAccessGate

List of Primary Access Servers now shows the newly added Access Server

This completes the setup for the new AccessGate. The next step is to configure the AccessGate on the client side.

Step 2 - Configure the AccessGate

Once the AccessGate entry has been created the **configureAccessGate** tool must be run. The **configureAccessGate** tool is available in the following directory of your ASDK:

```
[ASDK_HOME]\oblix\tools\configureAccessGate
```

The **configureAccessGate** tool will create a file called **ObAccessClient.xml**, which contains data on how the AccessGate is to connect to the Access Manager server. The following series of commands and prompts shows an example of running the tool on a Windows platform:

```
configureAccessGate.exe -i "C:\Program Files (x86)\NetPoint\AccessServerSDK"
-t AccessGate

 Please enter the Mode in which you want the AccessGate to run : 1(Open)
2(Simple) 3(Cert) : 1
 Please enter the AccessGate ID : API_SERVER-pc
Please enter the Password for this AccessGate :
 Please enter the Access Server ID : AccessServer_01
 Please enter the Access Server Host Machine Name : oracle-access-
manager.qa.com  Please enter the Access Server Port
: 7033  Preparing to connect to Access Server.
Please wait.
 AccessGate installed Successfully.

 Press enter key to continue ...
```

Running the **configureAccessGate** tool generates the all-important "ObAccessClient.xml" file in the following directory:

```
[ASDK_HOME]/oblix/lib
```

When configuring the OAM-based filters later in Policy Studio, you will need to specify this location as your OAM ASDK installation directory.

Configuring an 11g Webgate with OAM 11gR1 and 11gR2

If you are using an 11g OAM server, you will need to complete the instructions in this section. You can use the web-based **OAM Administration Console** to create the new Webgate. The web interface is available at the following URL, where [OAM_HOST] refers to the IP or hostname of the machine on which OAM is running:

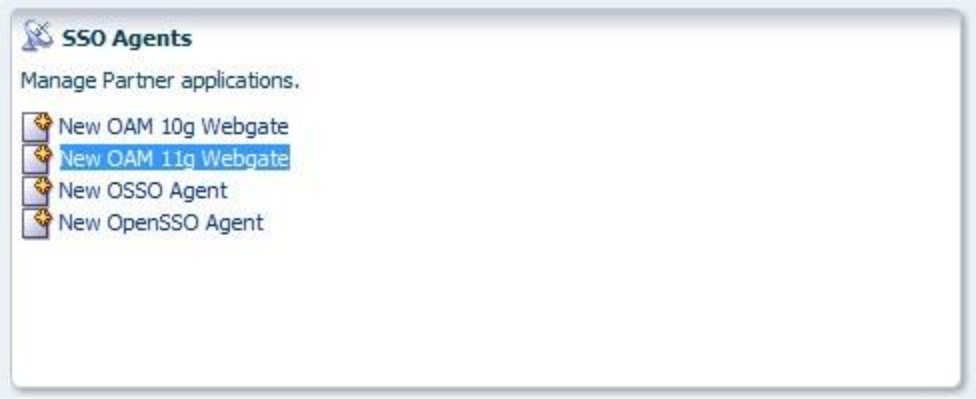
```
http://[OAM_HOST]:7001/oamconsole
```

Login using your Weblogic credentials and complete the following steps.

Step 1 - Create the 11g Webgate

You can create a new 11g Webgate by following these steps:

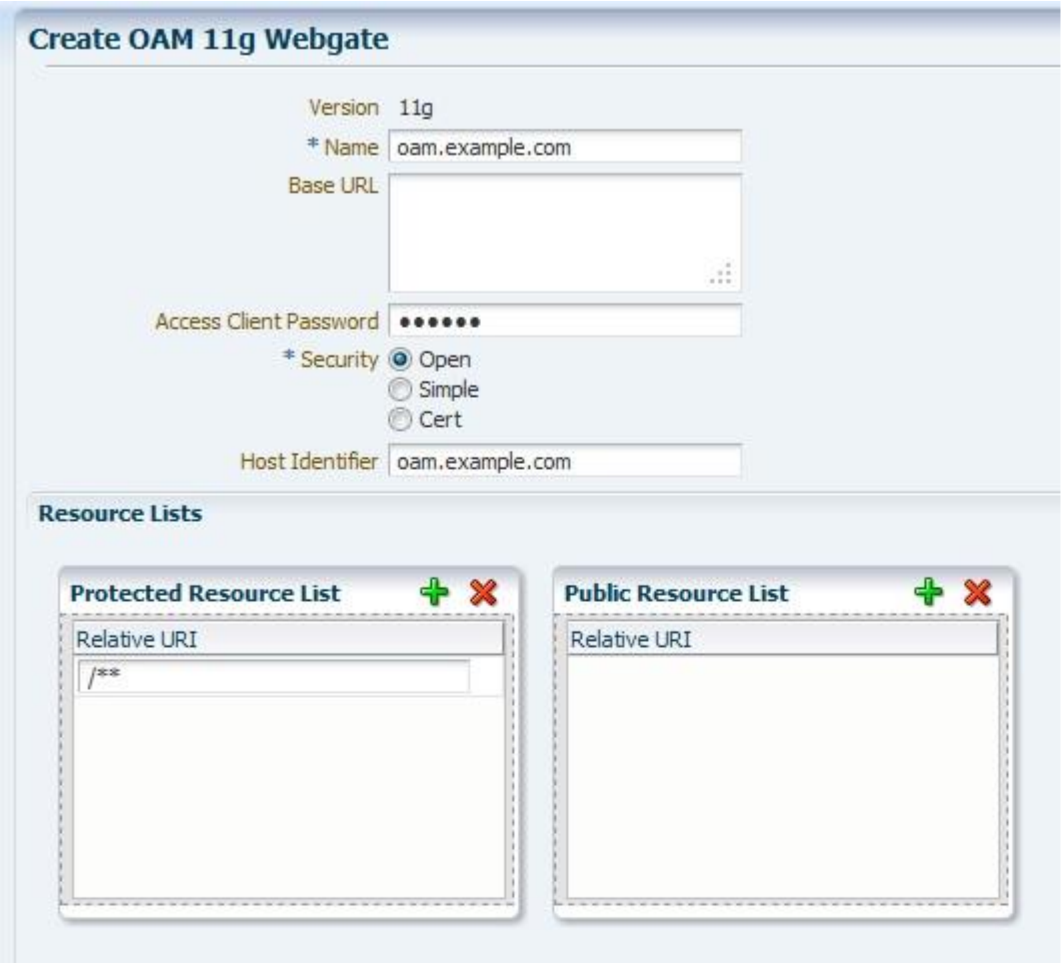
1. On the “Welcome” screen, click the **New OAM 11g Webgate** link.



Adding a new OAM 11g Webgate

2. Complete the following fields on the “Create OAM 11g Webgate” screen:

   ○ **Name**: Enter a unique name for this OAM 11g Webgate, for example, oam.example.com.
   ○ **Access Client Password**: Key in a suitable password for this Webgate.
   ○ **Host Identifier**: Enter the hostname of the machine on which your API Server and ASDK have been installed. In the screenshot below, we have elected to use the actual hostname (i.e. **Host Identifier**) as the **Name** of the new Webgate.



Create OAM 11g Webgate

Version 11g

* Name oam.example.com

Base URL

Access Client Password ●●●●●●

* Security ● Open ○ Simple ○ Cert

Host Identifier oam.example.com

Resource Lists

Protected Resource List

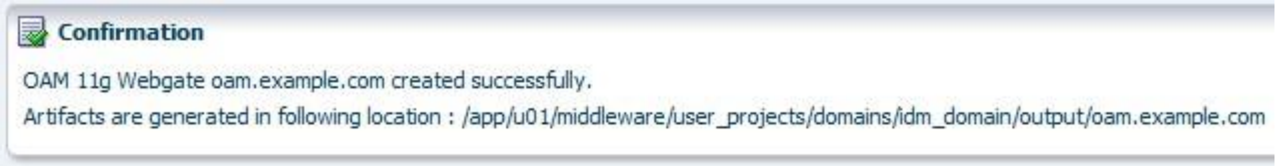
Relative URI
/**

Public Resource List

Relative URI

Creating an OAM 11g Webgate

3. Click the **Apply** button when you have completed the above configuration.
4. Note the location of the generated artifacts given in the confirmation message:



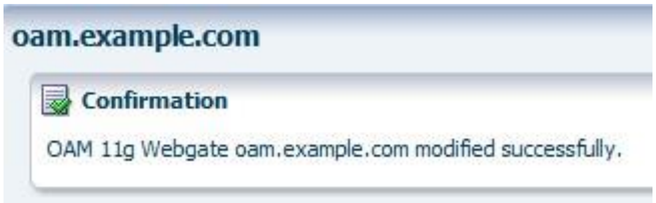
Confirmation

OAM 11g Webgate oam.example.com created successfully.

Artifacts are generated in following location : /app/u01/middleware/user_projects/domains/idm_domain/output/oam.example.com

Confirmation message indicating the location of the auto-generated artifacts

5. The complete configuration for the new Webgate is now displayed. You will need to enter a non-null value in the **Logout Target URL** field, for example, “end_url”.
6. Click **Apply** one more time to save the new **Logout Target URL**. You should see another confirmation message acknowledging the modification:



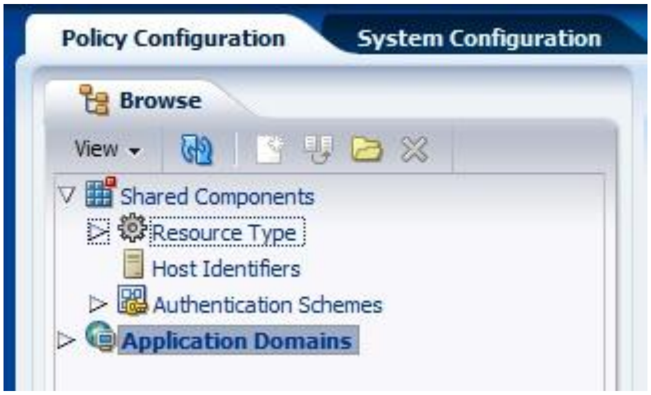
OAM 11g Webgate oam.example.com modified successfully.

Confirmation message to acknowledge the modification

Step 2 - Configure the Authentication Policy

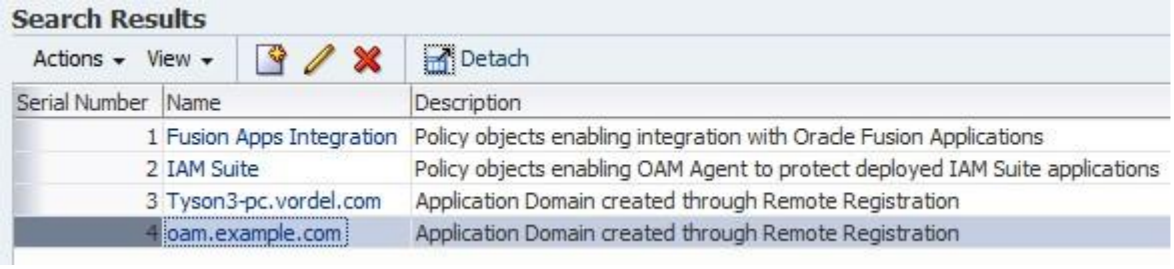
The next step is to update the Authentication Policy for the new Webgate:

1. Double-click on the **Application Domains** node in the tree view under the **Policy Configuration** tab:



Double-click the Application Domains node

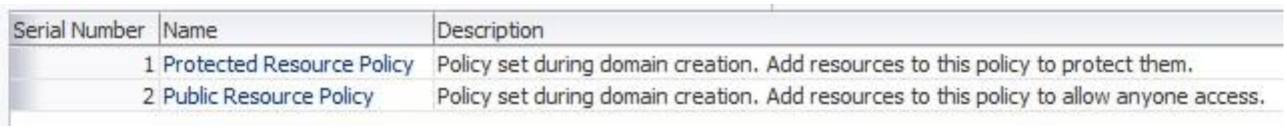
1. Click the **Search** button in the **Search** box. It is not necessary to enter anything in the **Search** field.
2. You will be able to see your newly created 11g Webgate in the list of Application Domains.



Serial Number	Name	Description
1	Fusion Apps Integration	Policy objects enabling integration with Oracle Fusion Applications
2	IAM Suite	Policy objects enabling OAM Agent to protect deployed IAM Suite applications
3	Tyson3-pc.vordel.com	Application Domain created through Remote Registration
4	oam.example.com	Application Domain created through Remote Registration

List of Application Domains showing the newly created 11g Webgate selected

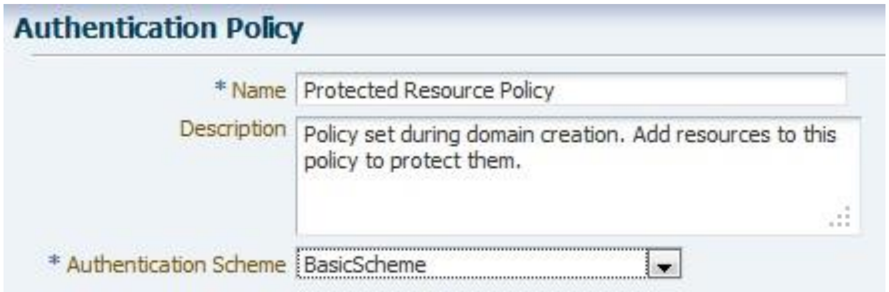
1. Click on the newly created “oam.example.com” Webgate link in the table.
2. Open the **Authentication Policies** tab to display the current list of policies.



Serial Number	Name	Description
1	Protected Resource Policy	Policy set during domain creation. Add resources to this policy to protect them.
2	Public Resource Policy	Policy set during domain creation. Add resources to this policy to allow anyone access.

Current list of Authentication Policies

3. Click the “Protected Resource Policy” link in the table.
4. Change the **Authentication Scheme** field from “LDAPScheme” to “BasicScheme”.



Authentication Policy

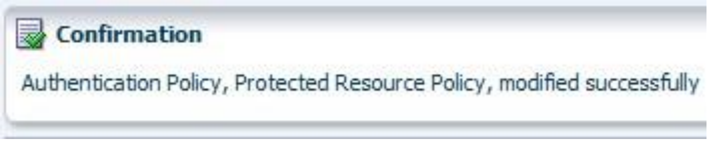
* Name: Protected Resource Policy

Description: Policy set during domain creation. Add resources to this policy to protect them.

* Authentication Scheme: BasicScheme

Change the Authentication Scheme from “LDAPScheme” to “BasicScheme”

5. Click the **Apply** button to save the change.
6. You will see a confirmation message indicating that the update was successful.



Confirmation

Authentication Policy, Protected Resource Policy, modified successfully

Confirmation of the update to the “Protected Resource Policy” Authentication Policy

Step 4 - Copy the Webgate Artifacts to the API Server Machine

The next step is to copy the auto-generated Webgate artifacts generated in Step 1 above. As notified in the confirmation message, the artifacts were generated in the following location after creating the 11g Webgate:

```
/app/u01/middleware/user_projects/domains/idm_domain/output/oam.example.com
```

A directory listing on this location shows that 2 files were generated when the 11g Webgate was created: the “ObAccessClient.xml” and “cwallet.sso” files.

```
[oracle@oam oam.example.com]$ ls
cwallet.sso  ObAccessClient.xml
```

Both of these files must be copied from the OAM machine to the machine on which you have installed the ASDK and are running the API Server. They must be copied to the ASDK_HOME/ config directory alongside the “jps-config.xml” file.

```
C:\Oracle\AccessServerSDK11\config>dir
10/12/2012  12:09 <DIR>      .
10/12/2012  12:09 <DIR>      ..
10/12/2012  12:06             3,149 cwallet.sso
27/11/2012  12:06             1,426 jps-config.xml
10/12/2012  12:11             3,043 ObAccessClient.xml
```

Step 5 - Modify the API Server Classpath

The API Server's classpath must be extended to include the ASDK jars. To achieve this, create a jvm.xml file at the following location:

```
<APISERVER_INSTALL_DIR>/conf/jvm.xml
```

Edit this `jvm.xml` so that its contents are as follows. Make sure to set the value of the ASDK_HOME environment variable to the location where you installed the ASDK.

```
<ConfigurationFragment>

 <!-- OAM ASDK Settings -->
 <Environment name="ASDK_HOME" value="C:\Oracle\AccessServerSDK11" />
<ClassDir name="$ASDK_HOME" />
 <VMArg  name="-
Doracle.security.jps.config=$ASDK_HOME/config/jpsconfig.xml"/>
</ConfigurationFragment>
```

Start the API Server

Assuming an API instance name of "APIServer1" belonging to a group called "Group1", you can start the API server using the following command depending on your target platform, where [APISERVER_HOME] refers to the root of your API Server installation:

On Windows:

```
[APISERVER_HOME]\Win32\bin>startinstance -n "APIServer1" -g "Group1"
```

On UNIX:

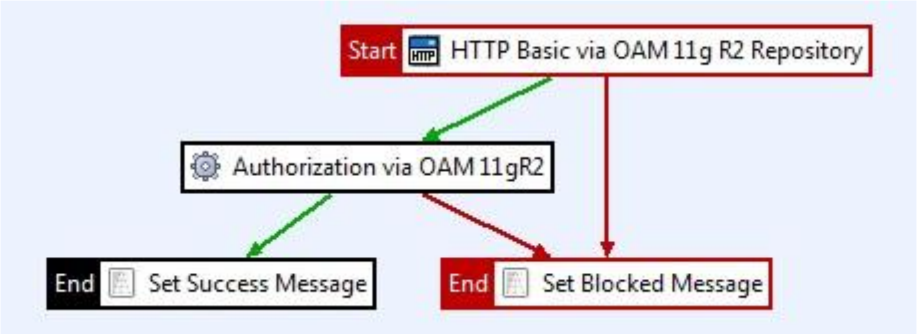
```
[APISERVER_HOME]/posix/bin>startinstance -n "APIServer1" -g "Group1"
```

Configure the API Server to Authenticate and Authorize against OAM

This section explains how to configure the API Server to delegate authentication and authorization decisions to Oracle Access Manager. The following steps are required:

- Configure the **OAM Authentication Repository**
- Configure a **HTTP Basic Authentication** filter to authenticate users against the OAM Authentication Repository
- Configure the **OAM Authorization** filter.
- Configure some appropriate success and failure messages.

The resulting policy created in the Gateway will appear as follows:


Start
HTTP Basic via OAM 11g R2 Repository
Authorization via OAM 11gR2
End
Set Success Message
End
Set Blocked Message

The complete policy showing the HTTP Basic Authentication, OAM 11g Authorization, and Set Message filters

Step 1 - Configure the OAM Authentication Repository

To configure the **OAM Authentication Repository**, perform the following steps:

1. Start the Policy Studio.
2. Expand the **External Connections** node in the tree view on the left hand side.
3. Expand the **Authentication Repository** node.
4. Click on the **Oracle Access Manager Repositories**.
5. Click on the **Add New Repository** link in the main window.
6. Configure the following fields in the **Authentication Repository** window:

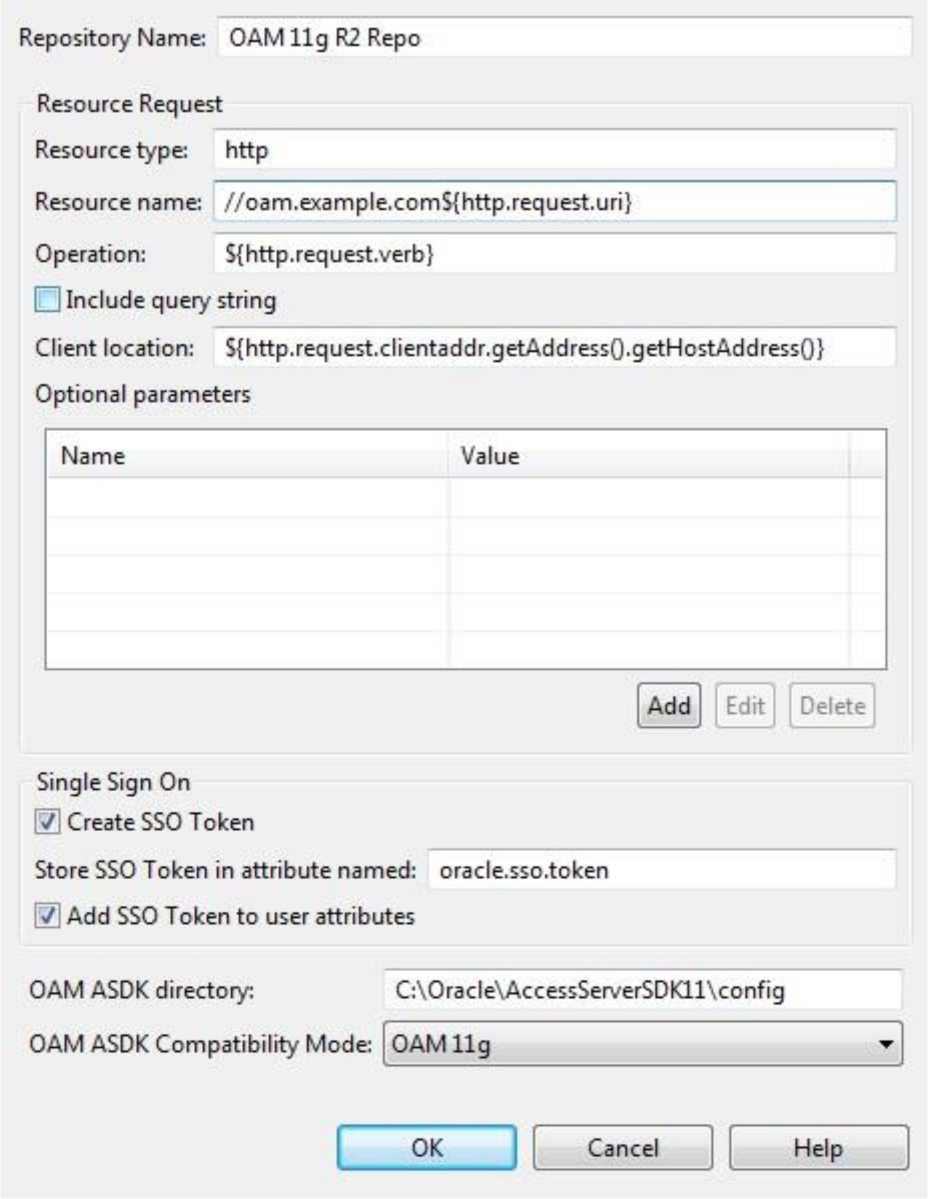
Field Name	Configuration
Name	Enter a suitable name for this Authentication Repository, such as “OAM 11g R2 Repo” when connecting to an 11gR2 OAM.
Resource Type	Enter “http” as the type of resource.

Resource Name	If you are running against a 10g OAM server, simply enter “${http.request.uri}” in the field provided. If you are running against an 11g OAM server, you must also enter the hostname in the resource, for example, “//oam.example.com${http.request.uri}”. Make sure to enter the exact same hostname in the **Resource Name** field as you did for the **Host Identifier** field when creating the OAM 11g Webgate, for example, “oam.example.com”. The value entered here is case-sensitive.
Operation	Enter “${http.request.verb}”, which will take the HTTP verb from the incoming request.
Client Location	The Selector Expression, “${http.request.clientaddr.getAddress().getHostAddress()}” will get the IP address of the client from the request.
Create SSO Token	Select “Yes” to create an SSO token for the authenticated user.
Store SSO token in attribute named	The SSO token will be stored in a message attribute called “oracle.sso.token”.
OAM ASDK Directory	The **OAM ASDK Directory** will differ depending on the ASDK version you have installed. ***OAM ASDK 10g (10.1.4.3.0):*** Assuming you have installed to the default location, enter: `C:\Program Files x86)\NetPoint\AccessServerSDK` ***OAM ASDK 11gR2 (11.1.2.0.0):*** Assuming you have followed the installation instructions in the Prerequisites section, enter: `C:\Oracle\AccessServerSDK11\config`

OAM ASDK Compatibility Mode	If you are running against an 11g OAM server, select “OAM 11g”. On the other hand, if you are running against a 10g OAM server,
	select “OAM 10g”.

7. Click OK to complete the configuration.

The following screenshot shows the OAM Authentication Repository configured to talk to an 11g OAM server:



Repository Name: OAM 11g R2 Repo

Resource Request

Resource type: http

Resource name: //oam.example.com${http.request.uri}

Operation: ${http.request.verb}

☐ Include query string

Client location: ${http.request.clientaddr.getAddress().getHostAddress()}

Optional parameters

Name	Value

Single Sign On

☑ Create SSO Token

Store SSO Token in attribute named: oracle.sso.token

☑ Add SSO Token to user attributes

OAM ASDK directory: C:\Oracle\AccessServerSDK11\config

OAM ASDK Compatibility Mode: OAM 11g

The complete OAM Authentication Repository configuration

Step 2 - Create a New Policy

Create a new policy in Policy Studio called, for example, “OAM 11gR2 Authentication and Authorization”. We will add the OAM authentication and authorization filters here.

Step 3 - Add the HTTP Basic Authentication Filter

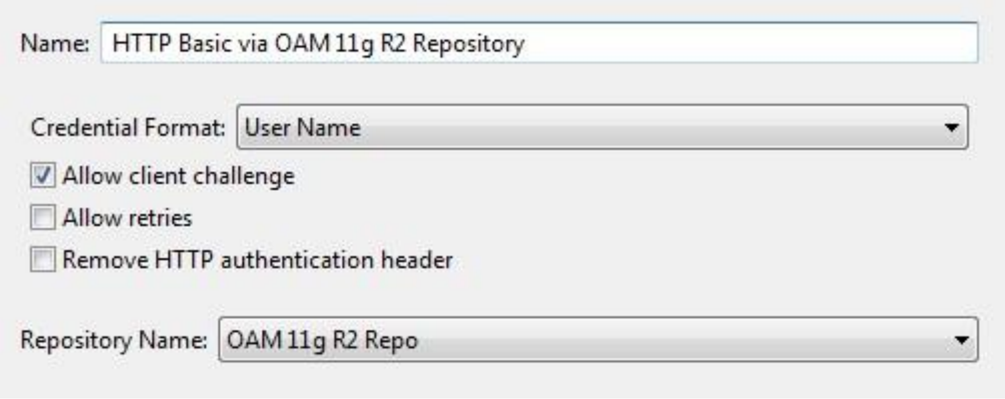
We will now create a **HTTP Basic Authentication** filter and configure it to authenticate users against the **OAM Authentication Repository** created in Step 1 earlier.

1. Open the newly created “OAM 11gR2 Authentication and Authorization” policy.
2. Drag a **HTTP Basic** filter from the **Authentication** category in the palette and drop it onto the canvas and configure it as follows:

Field Name	**Configuration**
Name	Enter a suitable name for this filter, such as “HTTP Basic via OAM 11g R2 Repository”.
Credential Format	Select **“**User Name” from the dropdown.
Allow Client Challenge	Check the “Allow client challenge” checkbox.
Repository Name	Select “OAM 11gR2 Repo” from the dropdown.

3. Click OK.

The completed configuration for the **HTTP Basic Authentication** filter appears as follows:



Name: HTTP Basic via OAM 11g R2 Repository

Credential Format: User Name

☑ Allow client challenge

☐ Allow retries

☐ Remove HTTP authentication header

Repository Name: OAM 11g R2 Repo

The configuration for the HTTP Basic Authentication filter

4. Right-click on the filter and select the **Set as Start** menu option.

Step 4 - Add the OAM Authorization Filter

The next step is to add the **OAM Authorization** filter, which will authorize authenticated users against OAM. Complete the following steps to configure the **OAM Authorization** filter:

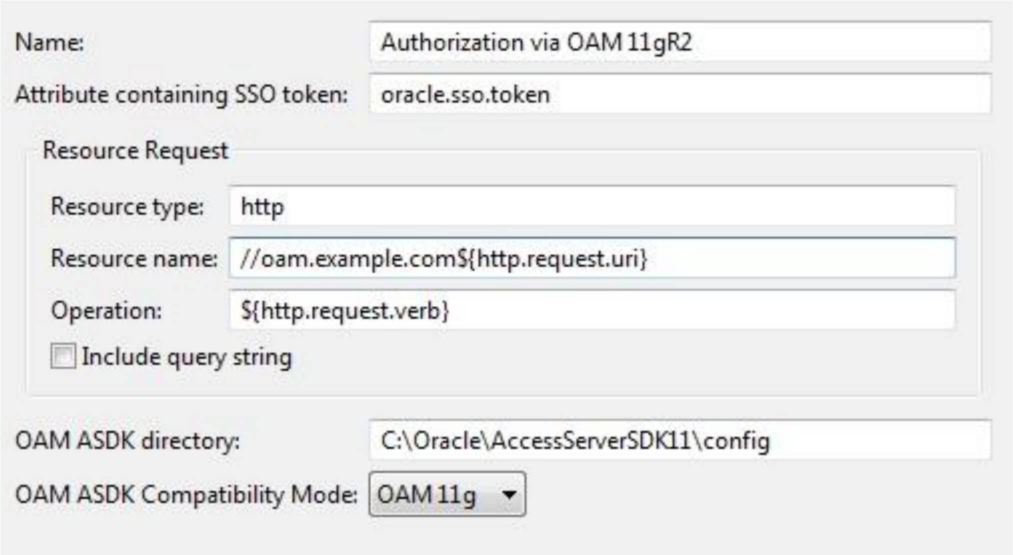
1. From the **Oracle Access Manager** category in the palette of the Policy Studio, drag the **Authorization** filter and drop it onto the **HTTP Basic Authentication** filter created in Step 3 above. By dropping a filter directly on to another filter, the new filter will be automatically connected to the first filter with a *success* path.
2. Configure the fields on the filter as follows:

Field Name	Configuration
Name	Enter a suitable name, such as “Authorization via OAM 11gR2”.
Attribute Containing SSO Token	Enter the name of the message attribute configured in the Authentication Repository earlier where the SSO token is stored, i.e. “oracle.sso.token”.
Resource Type	Enter a resource type of “http”.
Resource Name	If you are running against a 10g OAM server, simply enter “${http.request.uri}” in the field provided. If you are running against an 11g OAM server, you must also enter the hostname in the resource, for example, “//oam.example.com${http.request.uri}”. Make sure to enter the exact same hostname in the **Resource Name** field as you did for the **Host Identifier** field when creating the OAM 11g Webgate, for example, “oam.example.com”. The value entered here is case-sensitive.
Operation	You can use the “${http.request.verb}” Selector Expression to take the HTTP method from the incoming request.

OAM ASDK Directory	The **OAM ASDK Directory** will differ depending on the ASDK version you have installed. ***OAM ASDK 10g (10.1.4.3.0):*** Assuming you have installed to the default location, enter: `C:\Program Files (x86)\NetPoint\AccessServerSDK`
	OAM ASDK 11gR2 (11.1.2.0.0): Assuming you have followed the installation instructions in the Prerequisites section, enter: `C:\Oracle\AccessServerSDK11\config`
OAM ASDK Compatibility Mode	If you are running against an 11g OAM server, select "OAM 11g". On the other hand, if you are running against a 10g OAM server, select "OAM 10g".

3. Click OK.

The following screenshot shows the **OAM Authorization** filter configured to talk to an 11g R2 OAM server:



Name: Authorization via OAM 11gR2

Attribute containing SSO token: oracle.sso.token

Resource Request

Resource type: http

Resource name: //oam.example.com${http.request.uri}

Operation: ${http.request.verb}

□ Include query string

OAM ASDK directory: C:\Oracle\AccessServerSDK11\config

OAM ASDK Compatibility Mode: OAM 11g

The configuration for the OAM Authorization filter

Step 5 - Add the Success Message Filter

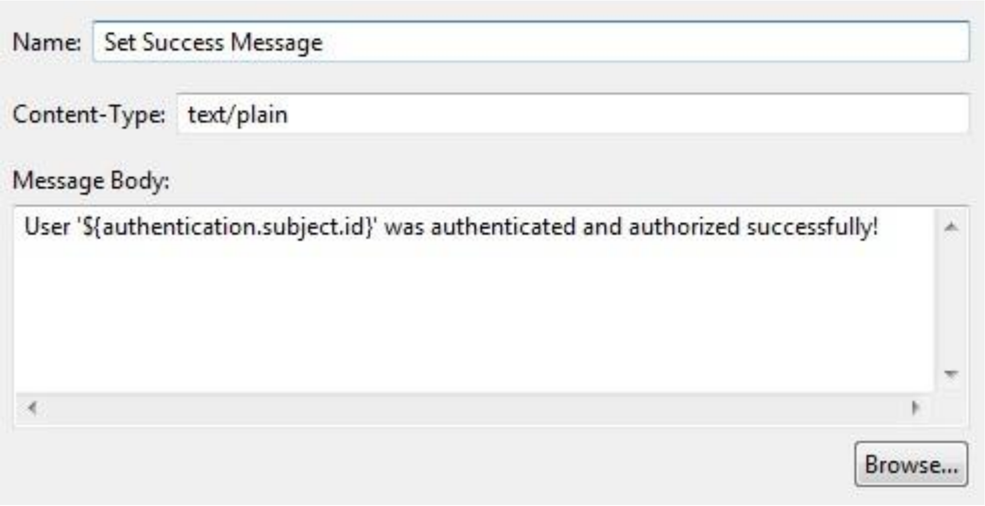
To display a “success” message to the user after successfully authorizing the user we can add a **Set Message** filter as follows:

1. Drag a **Set Message** filter from the **Conversion** category in the palette and drop it onto the **OAM Authorization** filter created in Step 4 above.
2. Configure the following fields on this filter:

Field Name	Configuration
Name	Enter “Set Success Message” in the text field.
Content-type	Enter “text/plain” as the content-type of the message to return to the client.
Message Body	Enter the following message to return to the client: “User '${authentication.subject.id}' was authenticated and authorized successfully!”

3. Click OK.

The configuration for the **Set Success Message** filter should now look like this:



Name: Set Success Message

Content-Type: text/plain

Message Body:

User '${authentication.subject.id}' was authenticated and authorized successfully!

Setting the success message using the Set Message filter

Step 6 - Add the Failure Message Filter

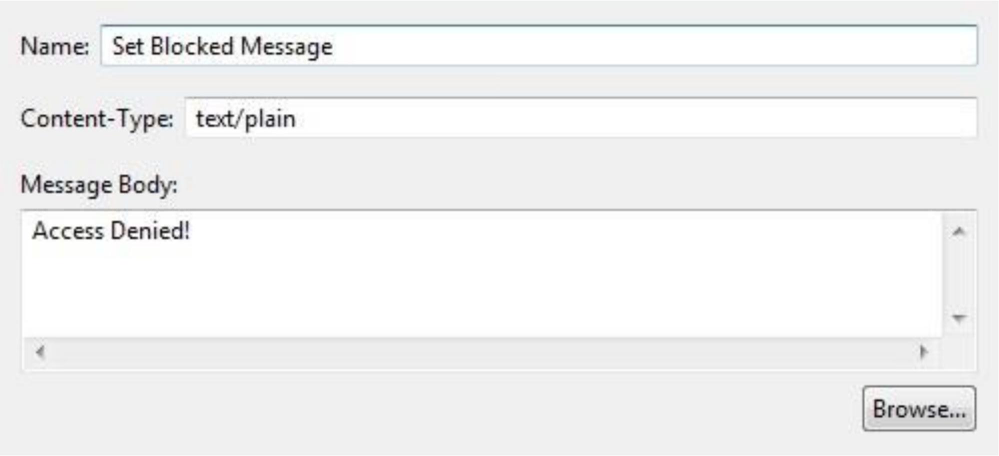
If OAM fails to authenticate and/or authorize the user we should return an appropriate error message to the client. To display a “failure” message to the client after an unsuccessful authentication/authorization event we can add another **Set Message** filter as follows:

1. Drag a **Set Message** filter from the **Conversion** category in the palette and drop it onto the **OAM Authorization** filter. Since this filter already has the "Set Success Message" filter connected on its *success* path, the new **Set Message** filter will be automatically added on its *failure* path.
2. Configure the following fields on this filter:

Field Name	Configuration
Name	Enter "Set Blocked Message" in the text field.
Content-type	Enter "text/plain" as the content-type of the message to return to the client.
Message Body	Enter the following message to return the client: "Access Denied!".

3. Click OK.

The configuration for the **Set Blocked Message** filter should now look like this:



Name: Set Blocked Message

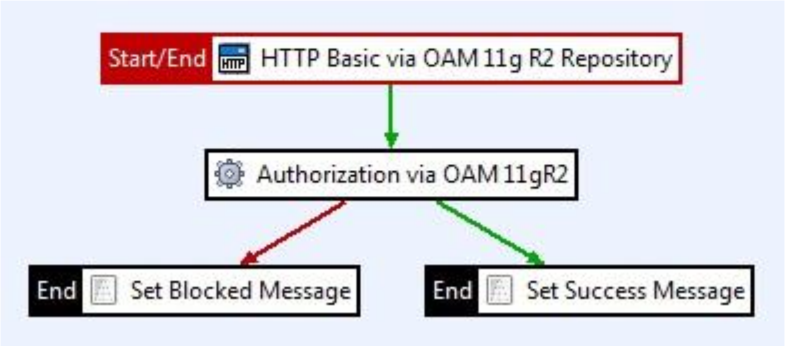
Content-Type: text/plain

Message Body:

Access Denied!

Setting the failure message using another Set Message filter

The following screenshot shows the policy we have configured so far:

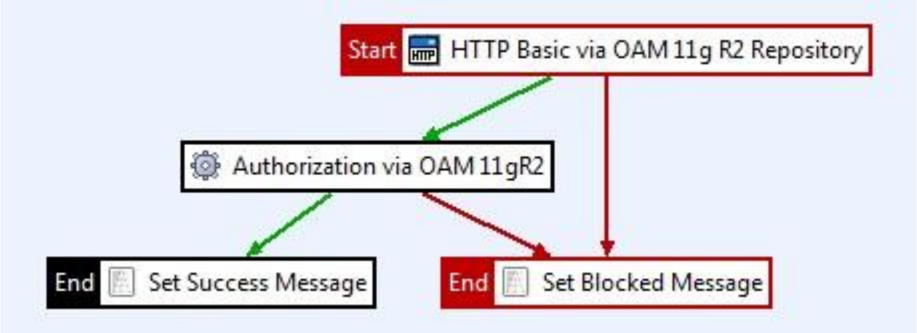


Start/End
HTTP Basic via OAM 11g R2 Repository
Authorization via OAM 11gR2
End
Set Blocked Message
End
Set Success Message

Policy showing the Authentication, Authorization, and Set Message filters

For completion, it would be useful to connect the “Set Blocked Message” filter to the “HTTP Basic via OAM 11g R2 Repository” filter along its failure path so that we get an appropriate failure messages when authentication AND authorization fails.

To do this, click on the **Failure Path** item at the top of the palette to select it. Then simply click on the “HTTP Basic via OAM 11g R2 Repository” filter and then click on the “Set Blocked Message filter to connect the failure path. The final policy now appears as follows:

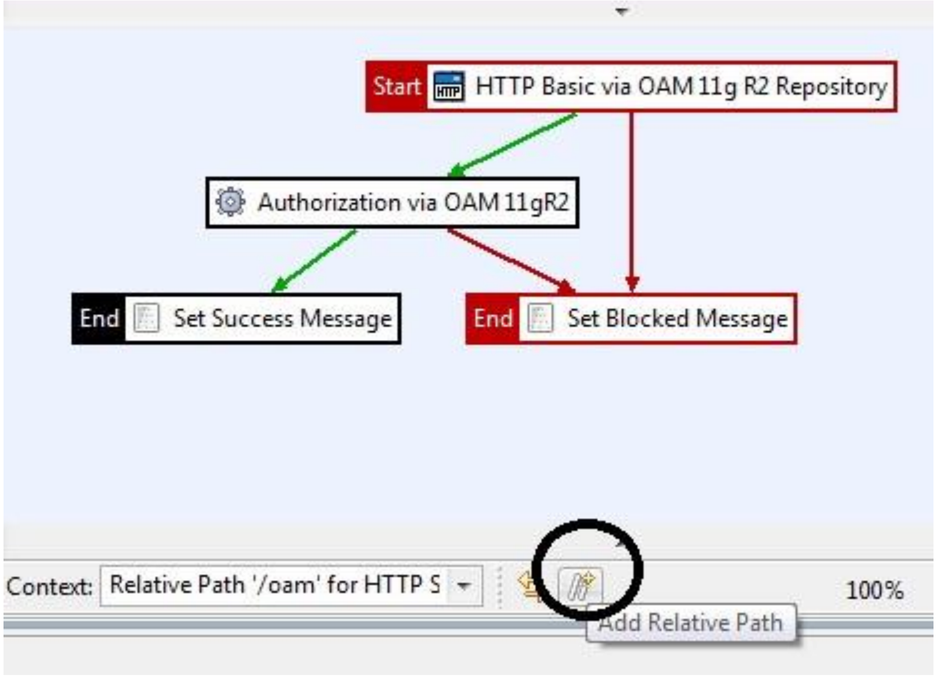

Start
HTTP Basic via OAM 11g R2 Repository
Authorization via OAM 11gR2
End
Set Success Message
End
Set Blocked Message

The complete OAM 11g authentication and authorization policy

Step 7 - Add a Relative Path for the OAM Authentication and Authorization Policy

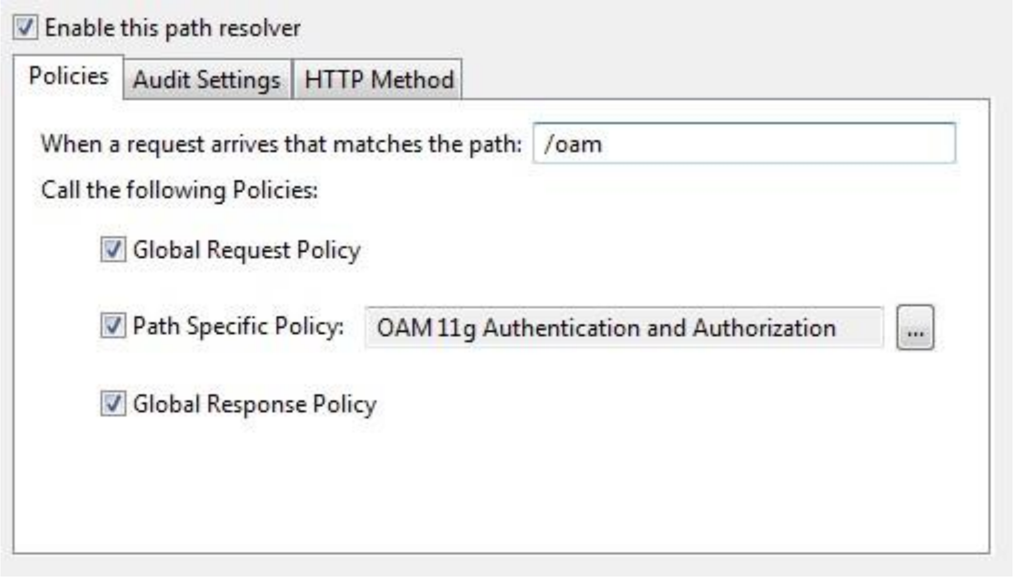
In order for the API Server to invoke the new policy for certain requests we need to create a **Relative Path** and map it to the policy. All requests received on this path will be automatically forwarded to the policy for processing.

To add a **Relative Path** for this policy click on the **Add Relative Path** button in the toolbar beneath the canvas. The **Add Relative Path** button is circled in the following screenshot:


Start
HTTP Basic via OAM 11g R2 Repository
Authorization via OAM 11gR2
End
Set Success Message
End
Set Blocked Message
Context:
Relative Path '/oam' for HTTP S
Add Relative Path
100%

Adding a Relative Path to map incoming requests to the newly created OAM policy

Enter the path on which the API Server will receive requests for this policy in the field provided in the Resolve Path to Policies dialog:



☑ Enable this path resolver

When a request arrives that matches the path: /oam

Call the following Policies:

☑ Global Request Policy

☑ Path Specific Policy: OAM 11g Authentication and Authorization

☑ Global Response Policy

Configuring the mapping from Relative Path to a Path Specific Policy

Enter a relative path of “/oam” in the field provided. You can see that this path is automatically mapped to the “OAM Authentication and Authorization” policy created earlier in this section.

Step 8 - Deploy the Policy

To push the configuration changes to the live API Server instance you must *deploy* the new policy. You can do this by clicking the **F6** button.

Testing the Integration Steps

Having completed the integration steps, we can now test the setup using the **Curl** testing utility. Assuming you are running the API Server on a machine called “apiserver” on the default port of 8080, you can send a POST request to the newly created policy on the API Server using HTTP basic authentication with the following command:

```
> curl --user weblogic:weblogic --data "test=data" http://apiserver:8080/oam
User 'weblogic' was authenticated and authorized successfully!
```

We can see that the success message has been returned by the API Server meaning that the "weblogic" user has been successfully authenticated and authorized by OAM to access the resource. A quick look at the API Server's trace output shows that the "weblogic" user has been authenticated and authorized to access the "//oam.example.com/oam" resource.

```
DEBUG 11/Dec/2012:13:07:44.090 [0794] run circuit "OAM 11g Authentication
and Authorization"...
DEBUG 11/Dec/2012:13:07:44.094 [0794] run filter [HTTP Basic via OAM 11g R2
Repository] {
DEBUG 11/Dec/2012:13:07:44.094 [0794] Check user name via Oracle Access
Manager
DEBUG 11/Dec/2012:13:07:44.338 [0794]    Creating ResourceRequest with
resType: 'http', resName: '//oam.example.com/oam, operation: 'POST'.
DEBUG 11/Dec/2012:13:07:44.339 [0794]     Successfully created
ResourceRequest
DEBUG 11/Dec/2012:13:07:44.543 [0794]    Login succeeded to OAM for user
weblogic
DEBUG 11/Dec/2012:13:07:44.544 [0794] } = 1, filter [HTTP Basic via OAM 11g
R2 Repository]
DEBUG 11/Dec/2012:13:07:44.544 [0794] Filter [HTTP Basic via OAM 11g R2
Repository] completes in 450 milliseconds.
DEBUG 11/Dec/2012:13:07:44.545 [0794] run filter [Authorization via OAM
11gR2] {
DEBUG 11/Dec/2012:13:07:44.545 [0794]    Creating ResourceRequest with
resType: 'http', resName: '//oam.example.com/oam, operation: 'POST'.
DEBUG 11/Dec/2012:13:07:44.545 [0794]     Successfully created
ResourceRequest
DEBUG 11/Dec/2012:13:07:44.545 [0794]    Authz for resource:
oracle.security.am.asdk.ResourceRequest@33aa7b
DEBUG 11/Dec/2012:13:07:44.638 [0794]
      User 'uid=weblogic,ou=people,ou=myrealm,dc=idm_domain' is authorized
for resource: //oam.example.com/oam
DEBUG 11/Dec/2012:13:07:44.638 [0794] } = 1, filter [Authorization via OAM
11g R2]
DEBUG 11/Dec/2012:13:07:44.639 [0794] Filter [Authorization via OAM 11gR2]
completes in 94 milliseconds.
DEBUG 11/Dec/2012:13:07:44.639 [0794] run filter [Set Success Message] {
DEBUG 11/Dec/2012:13:07:44.639 [0794]    The content type of the converted
message is text/plain
DEBUG 11/Dec/2012:13:07:44.640 [0794]    handle type text/plain with
factory class com.vordel.mime.Body$1
DEBUG 11/Dec/2012:13:07:44.640 [0794]    Added converted message is added
to the whiteboard
DEBUG 11/Dec/2012:13:07:44.640 [0794] } = 1, filter [Set Success Message]
DEBUG 11/Dec/2012:13:07:44.640 [0794] Filter [Set Success Message] completes
in 1 milliseconds. DEBUG 11/Dec/2012:13:07:44.641 [0794] ..."OAM 11g
Authentication and Authorization" complete.
```

Now let's see what happens when we authenticate with a user that hasn't been configured in OAM. To demonstrate this, we can send up a request using the credentials of the default API Server "admin" user:

```
> curl --user admin:changeme --data "test=data" http://apiserver:8080/oam
Access Denied!
```

If we take a quick look at the API Server's trace output once more, we can see that the **11g Authorization** filter has blocked the authorization request:

```
DEBUG   11/Dec/2012:14:35:42.331 [0ad4] run circuit "OAM 11g Authentication
and Authorization"...
DEBUG   11/Dec/2012:14:35:42.331 [0ad4] run filter [HTTP Basic via OAM 11g
R2 Repository] {
DEBUG   11/Dec/2012:14:35:42.331 [0ad4] Check user name via Oracle Access
Manager
DEBUG   11/Dec/2012:14:35:42.335 [0ad4] Creating ResourceRequest with
resType: 'http', resName: '//Tyson3-pc.vordel.com/oam, operation: 'POST'.
DEBUG   11/Dec/2012:14:35:42.336 [0ad4] Successfully created
ResourceRequest
ERROR   11/Dec/2012:14:35:42.486 [0ad4] Login failed to Oracle Access
Manager for user admin
ERROR   11/Dec/2012:14:35:42.487 [0ad4] java exception:
com.vordel.circuit.authn.VordelAuthNException: Login failed

...

DEBUG   11/Dec/2012:14:35:42.533 [0ad4] } = 0, filter [HTTP Basic via OAM
11g R2 Repository]
DEBUG   11/Dec/2012:14:35:42.534 [0ad4] Filter [HTTP Basic via OAM 11g R2
Repository] completes in 203 milliseconds.
ERROR   11/Dec/2012:14:35:42.534 [0ad4] The message [Id-
71222fbb50c744be03d40000] logged Failure at 12.11.2012 14:35:42,534 with log
description: HTTP basic authentication failed
DEBUG   11/Dec/2012:14:35:42.535 [0ad4] run filter [Set Blocked Message] {
DEBUG   11/Dec/2012:14:35:42.535 [0ad4] The content type of the converted
message is text/plain
DEBUG   11/Dec/2012:14:35:42.535 [0ad4] handle type text/plain with
factoryclass com.vordel.mime.Body$1
DEBUG   11/Dec/2012:14:35:42.536 [0ad4] Added converted message is added to
the whiteboard
DEBUG   11/Dec/2012:14:35:42.536 [0ad4] } = 1, filter [Set Blocked Message]
DEBUG   11/Dec/2012:14:35:42.536 [0ad4] Filter [Set Blocked Message]
completes in 1 milliseconds.
```

```
DEBUG   11/Dec/2012:14:35:42.536 [0ad4] ..."OAM 11g Authentication and
Authorization" complete.
```

Conclusion

This document outlines a simple demonstration to configure the API Gateway to authenticate and authorize a user’s request for a particular resource against a 10g, 11gR1, or 11gR2 Oracle Access Manager server.

This configuration can be part of a larger policy, including features such as XML threat detection, XML Signature, XML Encryption, and conditional routing, amongst many others. These features are out of scope of this document but are covered in other documents which can be obtained from Oracle Technology Network.