

# Table of Contents

|   |           |
|---|-----------|
| <b>1 Oracle Access Manager Integration</b> .....      | <b>1</b>  |
| 1.1 Overview.....                                     | 1         |
| 1.2 Prerequisites.....                                | 1         |
| 1.3 Deployment.....                                   | 1         |
| 1.4 Integration.....                                  | 1         |
| 1.5 Authentication Process.....                       | 1         |
| <b>2 Oracle WebLogic</b> .....                        | <b>2</b>  |
| <b>3 Overview</b> .....                               | <b>3</b>  |
| <b>4 Prerequisites</b> .....                          | <b>4</b>  |
| <b>5 Baseline</b> .....                               | <b>5</b>  |
| <b>6 Architecture</b> .....                           | <b>6</b>  |
| <b>7 Installation</b> .....                           | <b>7</b>  |
| 7.1 Swivel Integration Configuration.....             | 7         |
| 7.2 Configuring the Swivel Authentication Portal..... | 7         |
| 7.3 Create private keys and certificates.....         | 8         |
| 7.4 Generating IdP metadata.....                      | 8         |
| 7.5 WebLogic Integration Configuration.....           | 9         |
| 7.6 Additional Installation Options.....              | 20        |
| <b>8 Verifying the Installation</b> .....             | <b>21</b> |
| <b>9 Uninstalling the Swivel Integration</b> .....    | <b>22</b> |
| <b>10 Troubleshooting</b> .....                       | <b>23</b> |
| 10.1 Enabling WebLogic debugging.....                 | 23        |
| 10.2 Error Messages.....                              | 23        |
| <b>11 Known Issues and Limitations</b> .....          | <b>24</b> |
| <b>12 Additional Information</b> .....                | <b>25</b> |

# 1 Oracle Access Manager Integration

## 1.1 Overview

This article provides a basis for integration with Oracle Access Manager 10g.

Client code is included which you can deploy in conjunction with your Oracle Access Manager and PINsafe environment.

## 1.2 Prerequisites

- Oracle Access Manager 10g
- PINsafe 3.8

Download [GenericIntegration.zip](#) to obtain the client code for this integration. The code contains Eclipse project settings and a pre-built version of the WAR file for deployment.

## 1.3 Deployment

Deploy the war file into the Apache Tomcat webapps folder of your existing PINsafe 3.8 installation.

If using a PINsafe appliance, you can use WinSCP. See the [WinSCP How To Guide](#) for further information on transferring files to a PINsafe appliance.

## 1.4 Integration

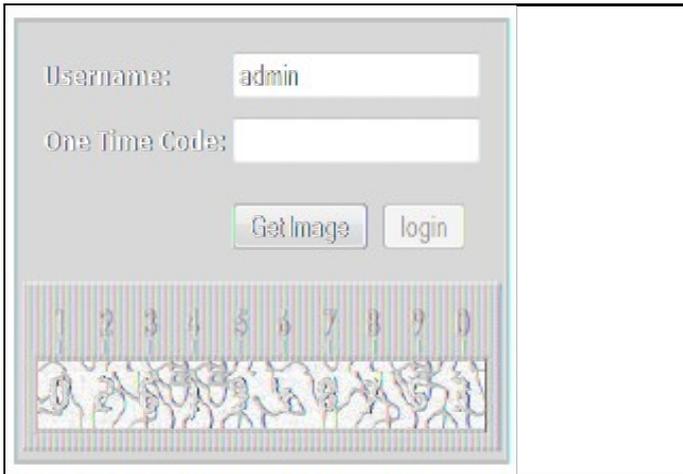


Image showing the Generic Integration login page

The integration requires a JSP containing username, password, OTC and the [TURING](#) image. Upon ?Logon? this is posted to a servlet which extracts the username, password and OTC from the request and calls the login (username, password, etc) of the PINsafeClient class (supplied in the jar). This method sends an HTTP request to PINsafe to allow a user to login or change their PIN.

Some pointers regarding the sample code provided:

- Marked in the code is the point in login.jsp where the creating of the OAM cookie described below should occur.
- WEB-INF/settings.xml contains the configuration to point to the PINsafe server, where to redirect upon success and whether or not to use a password.
- WEB-INF/settings2.xml contains a configuration for secondary server, for high availability purposes.

## 1.5 Authentication Process

During authentication, if the user is authenticated by PINsafe then depending on elements specific to your integration, you would either:

- Create an OAM cookie, which would require knowledge and availability of the OAM API;
- Redirect to the location for normal processing of the login (typically /oblix/login.cgi as detailed [on the Oracle website](#)). This would require a filter to stop anyone calling /oblix/login.cgi directly.

## 2 Oracle WebLogic

## 3 Overview

This document outlines the integration of Oracle WebLogic with Swivel using SAML with Swivel as an Identity Provider (IdP). It assumes that the Identity Provider and SAML Swivel Demo app are installed on the same Swivel appliance.

Swivel can provide Two Factor authentication with SMS, Token, Mobile Phone Client and strong Single Channel Authentication TURing, Pinpad or in the Taskbar.

To use the Single Channel Image such as the TURing Image, the Swivel server must be made accessible. **The client requests the images from the Swivel server, and is usually configured using a NAT (Network Address Translation)**, often with a proxy server. The Swivel virtual appliance or hardware appliance is configured with a proxy port to allow an additional layer of protection.

## 4 Prerequisites

Oracle WebLogic

Swivel 3.9 onwards

[Swivel AuthenticationPortal.zip](#). The file containing the IdP and login page to authenticate using Swivel.

[Swivel SAMLswivelDemo.zip](#). A simple app which sits on the service provider server to demonstrate how a user needs to be authenticated.

## 5 Baseline

Swivel 3.9, 3.10

Oracle WebLogic 12.1.1

## 6 Architecture

Swivel is configured as an Identity Provider, see the following [Oracle Documentation](#).

# 7 Installation

To implement the solution there are several steps:

- Setup up the Identity Provider (IdP) (Authentication Portal)
- Generate the IdP metadata (which is used to create the relationship between the IdP and Service Provider).
- Setup the service provider (the federation service and its association with the Idp)
- Create a user within PINSafe and Weblogic
- Install the demonstration application
- Test the solution

## 7.1 Swivel Integration Configuration

### 7.1.1 Configuring Swivel for Agent XML Authentication

The IdP is usually deployed on the Swivel hardware or virtual appliance, and a default localhost Agent is usually pre-configured. To make any changes to this see [Agents How to Guide](#)

### 7.1.2 Configuring Swivel for Single Channel Images

If Swivel Single Channel images are to be used for authentication, then the following guide can be used.

[Single Channel How To Guide](#)

### 7.1.3 Configuring Swivel for Dual Channel Authentication

If Swivel Dual Channel authentication methods are to be used, refer to the following guide:

[Transport Configuration](#)

## 7.2 Configuring the Swivel Authentication Portal

Download and extract the AuthenticationPortal.war file from the AuthenticationPortal.zip and copy this file using [WinSCP](#) to /usr/local/tomcat/webapps2 where a folder called AuthenticationPortal should appear.

Within the AuthenticationPortal folder, there will be folder called WEB-INF, with the settings.xml file (/usr/local/tomcat/webapps2/WEB-INF/settings.xml). Right click settings.xml and either Edit the file or Open in another editor such as Notepad++.

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE properties SYSTEM "http://java.sun.com/dtd/properties.dtd">
<properties>
<entry key="pinsafessl">false</entry>
<entry key="pinsafeserver">localhost</entry>
<entry key="pinsafecontext">pinsafe</entry>
<entry key="pinsafesecret">secret</entry>
<entry key="pinsafeport">8080</entry>
<entry key="imagessl">false</entry>
<entry key="imageserver">localhost</entry>
<entry key="imagecontext">pinsafe</entry>
<entry key="imageport">8080</entry>
<entry key="selfsigned">>true</entry>
<entry key="serviceProviderEndpointURL">https://login.salesforce.com/?saml=02HKiPoin4nQspKPHoScmudQmsKtM.qRKnViSBCmh05IC52m5VptCNw0.p</entry>
<entry key="audience">https://saml.salesforce.com</entry>
<entry key="certificateIssuer">SAML_SP</entry>
<entry key="publicKeyFilePath">/keys/pinsafe/ssl/dsapubkey.der</entry>
<entry key="privateKeyFilePath">/keys/pinsafe/ssl/dsaprivkey.der</entry>
<entry key="certificateFilePath">/keys/pinsafe/ssl/dsacert.pem</entry>
</properties>
```

**pinsafessl** Communication between the IdP and Swivel. If SSL is used on the Swivel server set this to true, otherwise false. For a Swivel Hardware or Virtual appliance this should be changed to false when using port 8181 if Swivel is deployed in webapps2.

**pinsafeserver** Communication between the IdP and Swivel. Where the IdP is installed on the same server as Swivel this should be set to localhost.

**pinsafecontext** Communication between the IdP and Swivel. This is the install context and is usually pinsafe.

**pinsafesecret** Communication between the IdP and Swivel. By default a Swivel hardware or virtual appliance uses this value as the shared secret.

**pinsafeport** Communication port between the IdP and Swivel. For a Swivel Hardware or Virtual appliance this should be changed to 8181 if Swivel is deployed in webapps2 and uses a non SSL connection.

**imagessl** Communication between the IdP and User. If SSL is used on the Swivel server set this to true, otherwise false.

**imageserver** Communication between the IdP and User. If SSL is used on the Swivel server set this to true, otherwise false. By default a Swivel hardware or virtual appliance uses SSL.

**imagecontext** Communication between the IdP and User. This is the install context and is usually pinsafe.

**imageport** Communication between the IdP and User. For a Swivel Hardware or Virtual appliance this should be changed to 8443 although 443 or other port can also be used.

**selfsigned** Communication between the IdP and User. If SSL is used on the Swivel server with a self signed certificate then set this to true, otherwise false. By default a Swivel hardware or virtual appliance uses SSL with a self signed certificate.

**serviceProviderEndpointURL** the Published Site URL, defined in Setting up the Service Provider. Example:<https://192.168.10.10/saml2>

**audience**

**certificateIssuer** SAML\_SP

**publicKeyFilePath** path to the public key usually /keys/pinsafe/ssl/dsapubkey.der

**privateKeyFilePath** path to the private key usually /keys/pinsafe/ssl/dsaprivkey.der

**certificateFilePath** path to the certificate usually /keys/pinsafe/ssl/dsacert.pem

## 7.3 Create private keys and certificates

Communication between Oracle and the Swivel instance is secure through the use of certificates.

### 7.3.1 Creating DSA Private Key

DSA key generation involves two steps, and can be done through the command line on a Swivel virtual appliance or hardware appliance:

1.

```
openssl dsaparam -out dsaparam.pem 2048
```

2.

```
openssl gendsa -out dsaprivkey.pem dsaparam.pem
```

The first step creates a DSA parameter file, `dsaparam.pem`, which in this case instructs OpenSSL to create a 2048-bit key in Step 2. The `dsaparam.pem` file is not itself a key, and can be discarded after the public and private keys are created. The second step actually creates the private key in the file `dsaprivkey.pem` which should be kept secret.

Export the key into a DER (binary) format. You can do so with the following steps:

1.

```
openssl dsa -in dsaprivkey.pem -outform DER -pubout -out dsapubkey.der
```

2.

```
openssl pkcs8 -topk8 -inform PEM -outform DER -in dsaprivkey.pem -out dsaprivkey.der -nocrypt
```

Step 1 extracts the public key into a DER format. Step 2 converts the private key into the pkcs8 and DER format. Once you've done this, you can use this public (`dsapubkey.der`) and private (`dsaprivkey.der`) key pair.

### 7.3.2 Creating a Certificate

Once you have your key pair, it's easy to create an X.509 certificate. The certificate holds the corresponding public key, along with some metadata relating to the organization that created the certificate. Follow this step to create a self-signed certificate from either an RSA or DSA private key:

```
openssl req -new -x509 -key dsaprivkey.pem -out dsacert.pem
```

After you answer a number of questions, the certificate will be created and saved as `dsacert.pem`.

The created keys, `dsapubkey.der` and `dsapubkey.der` need to be copied to the keys folder or wherever specified within `settings.xml`

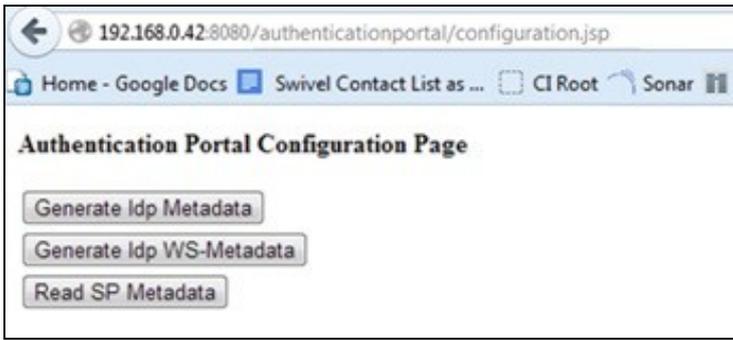
The **dsacert.pem** certificate needs to be uploaded to the GoogleApps server.

## 7.4 Generating IdP metadata

SAML metadata is generated by the IdP to simplify the mapping process between itself and the Service Provider.

The `AuthenticationPortal` folder should be located under `/usr/local/tomcat/webapps2`. In order to gain access to the Authentication Portal webpage, you must navigate to `https://<IPAddress>:8443/AuthenticationPortal/configuration.jsp`. (case sensitive).

This will display the configuration page as shown below. From here you should press `?Generate Idp Metadata?`.

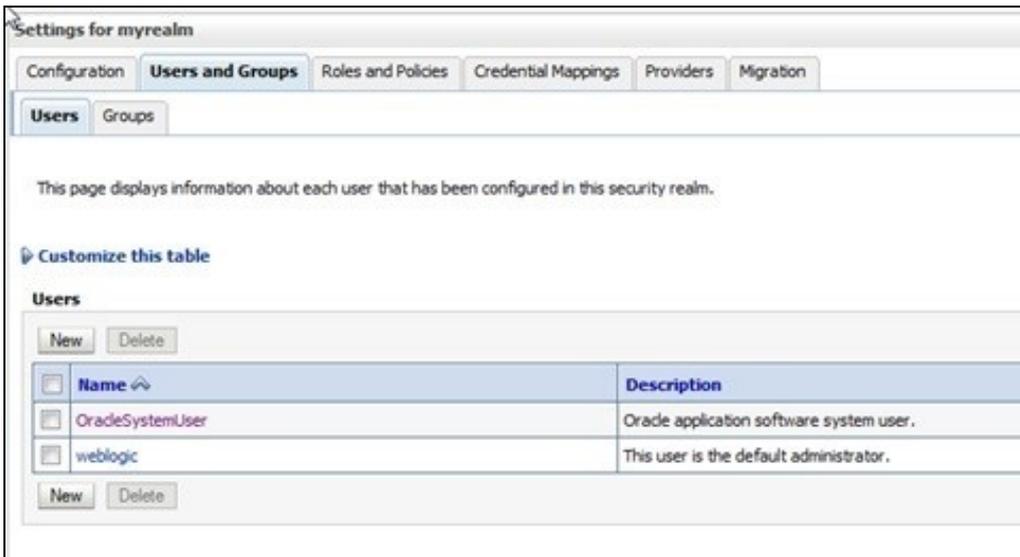


If successful, the metadata will be written to the root of the web application with the message "Metadata successfully written to" and the full path and filename displayed. Make a note of the destination which will be used later when configuring the Service Provider.

## 7.5 WebLogic Integration Configuration

### 7.5.1 Configure a WebLogic User

On the WebLogic Administration console main menu select Security Realms, select myrealm then select Users and Groups and then the Users tab to show following main screen:



Select New then input a username, this should match to the e-mail address of a Swivel user test user. Enter a dummy password as this will not be used by this integration, then press OK to save.

**Create a New User**

OK Cancel

**User Properties**

The following properties will be used to identify your new User.

\* Indicates required fields

What would you like to name your new User?

\* **Name:** a.douglas@swivelsecui

How would you like to describe the new User?

**Description:** My test user

Please choose a provider for the user.

**Provider:** DefaultAuthenticator ▼

The password is associated with the login name for the new User.

\* **Password:** ●●●●●●

\* **Confirm Password:** ●●●●●●

OK Cancel

### 7.5.2 Setting up the Service Provider

On the WebLogic Administration console main menu select Environment, Servers then select AdminServer(admin). Then select Configuration, Federation Services and SAML 2.0 General to get the following screen:

Settings for AdminServer

Configuration Protocols Logging Debug Monitoring Control Deployments Services Security Notes

General Cluster Services Keystores SSL **Federation Services** Deployment Migration Tuning Overload Health Monitoring Server Sta

SAML 1.1 Source Site SAML 1.1 Destination Site **SAML 2.0 General** SAML 2.0 Identity Provider SAML 2.0 Service Provider

Save Publish Meta Data

This page configures the general SAML 2.0 per server properties

— General —

 Replicated Cache Enabled

— Site Info —

Contact Person Given Name:

Contact Person Surname:

Contact Person Type:  ▼

Contact Person Company:

Contact Person Telephone Number:

Contact Person Email Address:

Organization Name:

Organization URL:

Published Site URL:

Entity ID:

Published Site URL should be your WebLogic URL + /saml2 and the Entity ID should be SAML\_SP to match up other aspects of the configuration. Ensure that under the Bindings option, Recipient Check Enabled is not checked and is therefore disabled. Enter other details as appropriate then press Save.

Then, from the same screen, select SAML 2.0 Service Provider to get the following screen:

Settings for AdminServer

Configuration Protocols Logging Debug Monitoring Control Deployments Services Security Notes

General Cluster Services Keystores SSL **Federation Services** Deployment Migration Tuning Overload Health Monitoring Serv

SAML 1.1 Source Site SAML 1.1 Destination Site SAML 2.0 General SAML 2.0 Identity Provider **SAML 2.0 Service Provider**

Save

This page configures the SAML 2.0 per server service provider properties

Enabled

Always Sign Authentication Requests

Force Authentication

Passive

Only Accept Signed Assertions

Authentication Request Cache Size:

Authentication Request Cache Timeout:

POST One Use Check Enabled

POST Binding Enabled

Artifact Binding Enabled

Preferred Binding:

Default URL:

Save

Ensure the checkboxes are set as above and for the Default URL enter the path to the SAMLSwivelDemo. Press Save. Making sure that the Published Site URL is your WebLogic URL and by adding /saml2. E.g. <http://192.168.10.10/saml2> - This is your serviceProviderEndpointURL.

Going back to the section Setting up the IdP, you can go back to the settings.xml and add for example:

```
<entry key="serviceProviderEndpointURL">https://192.168.10.10/saml2</entry>
```

?

### 7.5.3 Specifying the IdP

On the WebLogic Administration console main menu select Security Realms, select myrealm then select Providers and Authentication to show following main screen:

Settings for myrealm

Configuration Users and Groups Roles and Policies Credential Mappings **Providers** Migration

Authentication Password Validation Authorization Adjudication Role Mapping Auditing Credential Mapping Certification Path Keystores

An Authentication provider allows WebLogic Server to establish trust by validating a user. You must have one Authentication provider in a security realm, and you can configure multiple Authentication providers in a security realm. Different types of Authentication providers are designed to access different data stores, such as LDAP servers or DBMS. You can also configure a Realm Adapter Authentication provider that allows you to work with users and groups from previous releases of WebLogic Server.

Customize this table

Authentication Providers

New Delete Reorder

| Name  | Description                          | Version |
|---|--------------------------------------|---------|
| <input checked="" type="checkbox"/> DefaultAuthenticator    | WebLogic Authentication Provider     | 1.0     |
| <input checked="" type="checkbox"/> DefaultIdentityAsserter | WebLogic Identity Assertion Provider | 1.0     |

New Delete Reorder

Showing 1 to 2 of 2 Previous | Next

Select New to create a SAML2IdentityAsserter and name it SAML2IdentityAsserter as shown here:

Home > Providers > SAML > Summary of Security Realm > myrealm > Users and Groups > Realm Roles > Credential Mappings > Providers

Create a New Authentication Provider

OK Cancel

Create a new Authentication Provider

The following properties will be used to identify your new Authentication Provider.  
\* Indicates required fields

The name of the authentication provider.

\* Name: SAML2IdentityAsserter

This is the type of authentication provider you wish to create.

Type: SAML2IdentityAsserter

OK Cancel

Pressing OK will take you to the following screen.

Settings for myrealm

Configuration Users and Groups Roles and Policies Credential Mappings Providers Migration

Authentication Password Validation Authorization Adjudication Role Mapping Auditing Credential Mapping Certification Path Keystones

An Authentication provider allows WebLogic Server to establish trust by validating a user. You must have one Authentication provider in a security realm, and you can configure multiple Authentication providers in a security realm. Different types of Authentication providers are designed to access different data stores, such as LDAP servers or DBMS. You can also configure a Realm Adapter Authentication provider that allows you to work with users and groups from previous releases of WebLogic Server.

Customize this table

Authentication Providers

New Delete Reorder Showing 1 to 3 of 3 Previous Next

| Name                    | Description   | Version |
|-------------------------|---|---------|
| DefaultAuthenticator    | WebLogic Authentication Provider  | 1.0     |
| DefaultIdentityAsserter | WebLogic Identity Assertion provider  | 1.0     |
| SAML2IdentityAsserter   | SAML 2.0 Identity Assertion Provider. Supports Security Assertion Markup Language v2.0. | 1.0     |

New Delete Reorder Showing 1 to 3 of 3 Previous Next

At this point you need to activate the changes. One way you can do this is from the main menu select Environment, select Servers then select AdminServer(admin). Then select Control. Select the checkbox next to AdminServer(admin) and Shutdown. Then restart the server and logon to the admin console.

Return to the same screen and select the SAML2IdentityAsserter.

Settings for myrealm

Configuration Users and Groups Roles and Policies Credential Mappings Providers Migration

Authentication Password Validation Authorization Adjudication Role Mapping Auditing Credential Mapping Certification Path

An Authentication provider allows WebLogic Server to establish trust by validating a user. You must have one Authentication provider in a security realm, and you can configure multiple Authentication providers in a security realm. Different types of Authentication providers are designed to access different data stores, such as LDAP servers or DBMS. You can also configure a Realm Adapter Authentication provider that allows you to work with users and groups from previous releases of WebLogic Server.

Customize this table

Authentication Providers

New Delete Reorder

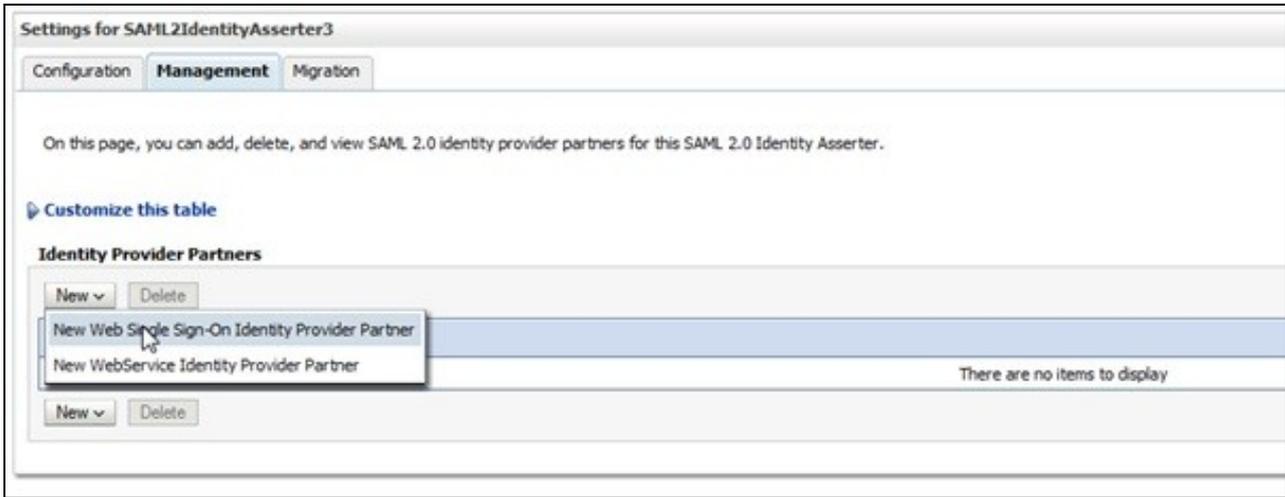
| Name                    | Description   |
|-------------------------|---|
| DefaultAuthenticator    | WebLogic Authentication Provider  |
| DefaultIdentityAsserter | WebLogic Identity Assertion provider  |
| SAML2IdentityAsserter   | SAML 2.0 Identity Assertion Provider. Supports Security Assertion Markup Language v2.0. |

New Delete Reorder

Then select Management to get the screen below:



Select New and New Web Single Sign-On Identity Provider Partner as shown below:



Select New then locate and select the IdP metadata as shown below. Press OK to save

### Create a SAML 2.0 Web Single Sign-on Identity Provider Partner

OK Cancel

**Partner Properties**

Use this page to:

- Enter the name of your new Single Sign-on Identity Provider partner
- Specify the name and location of the SAML 2.0 metadata file that you received from this new partner

\* Indicates required fields

Please specify the name of the partner.

\* **Name:**

Please specify the name of the file containing the partner metadata document.

**Path:**

**Recently Used Paths:**

- C:\Oracle\Middleware\user\_projects\domains\base\_domain
- C:\Users\adouglas\workspace3.9.2\SAMLOracle
- C:\

**Current Location:** 192.168.0.42 | C: | Users | adouglas | workspace3.9.2 | SAMLOracle

- .externalToolBuilders
- .settings
- .svn
- build
- src
- WebContent
- build.xml
- example.xml
- generatedIdPMetadata.xml

OK Cancel

Thus will take you to the following screen:

### Settings for SAML2IdentityAsserter

Configuration **Management** Migration

On this page, you can add, delete, and view SAML 2.0 identity provider partners for this SAML 2.0 Identity Asserter.

[Customize this table](#)

**Identity Provider Partners**

New Delete

| <input type="checkbox"/> | Name ↕               |
|--------------------------|----------------------|
| <input type="checkbox"/> | WebSSO-IdP-Partner-0 |

New Delete

? Select WebSSO-IdP-Partner-0 which will take you to the following screen:

Settings for SAML2IdentityAsserter

General Site Info Single Sign-On Signing Certificate Transport Layer Client Certificate Single Sign-On Service Endpoints Artifact Resolution Service Endpoints

Save

Configures a SAML 2.0 Web Single Sign-on Identity Provider Partner's General Properties

The parameters that can be set on this Administration Console page can also be accessed programmatically via the Java interfaces that are identified in this help topic. For

— Overview —

Name: WebSSO-IdP-Partner-0

Enabled

Description:

— Authentication Requests —

Identity Provider Name Mapper Class Name:

Issuer URI: SAML\_SP

Virtual User

Redirect URIs:

/SAMLswivelDemo/\*

Ensure Enabled and Virtual User are checked and that Redirect URIs is set to /SAMLswivelDemo/\*. Press Save to save your settings.

### 7.5.4 Credential Mapping Provider

On the WebLogic Administration console main menu select Security Realms, myrealm then select Providers and Authentication to show following main screen:

Settings for myrealm

Configuration Users and Groups Roles and Policies Credential Mappings Providers Migration

Authentication Password Validation Authorization Adjudication Role Mapping Auditing Credential Mapping Certification Path Keystores

A Credential Mapping provider allows WebLogic Server to log into a remote system on behalf of a subject that has already been authenticated. You must have one Credential Mapping providers in a security realm.

Customize this table

Credential Mapping Providers

New Delete Reorder

| Name   | Description                          |
|--|--------------------------------------|
| <input type="checkbox"/> DefaultCredentialMapper | WebLogic Credential Mapping Provider |

New Delete Reorder

Select New and then enter a name of SAML2CredentialsMapper and select type of SAML2CredentialsMapper as below (then Press OK to save):

**Create a New Credential Mapping Provider**

OK Cancel

**Create a new Credential Mapping Provider**

The following properties will be used to identify your new Credential Mapping Provider.

\* Indicates required fields

The name of the Credential Mapping Provider.

\* **Name:** SAML2CredentialsMap

This is the type of credential mapping provider you wish to create.

**Type:** SAML2CredentialMapper

OK Cancel

Select SAML2CredentialsMapper then configuration and Provider Specific. For the Issuer URI enter SAML\_SP as shown below (then press Save):

Settings for SAML2CredentialsMapper

Configuration Management Migration

Common Provider Specific

Save

Use this page to configure provider-specific information for this SAML 2.0 Credential Mapping provider.

Issuer URI: SAML\_SP

Name Qualifier:

Default Time To Live: 120

Default Time To Live Offset: -5

Web Service Assertion Signing Key Alias:

Web Service Assertion Signing Key Pass Phrase:

Please type again To confirm:

Name Mapper Class Name:

Generate Attributes

Save

### 7.5.5 Setting up the demo application

On the WebLogic Administration console main menu select Deployments to get the main screen looking as such:

**Summary of Deployments**

**Control** | Monitoring

This page displays a list of Java EE applications and stand-alone application modules that have been installed to this domain. Installed applications are listed in the table below. To view the details of an application, first selecting the application name and using the controls on this page.

To install a new application or module for deployment to targets in this domain, click the Install button.

[Customize this table](#)

**Deployments**

Install | Update | Delete | Start v | Stop v

| <input type="checkbox"/>      | Name ^ | State | Health | Type |
|-------------------------------|--------|-------|--------|------|
| There are no items to display |        |       |        |      |

Install | Update | Delete | Start v | Stop v

Select Install then locate the WAR file for the SAMLswivelDemo as such:

**Install Application Assistant**

Back | Next | Finish | Cancel

**Locate deployment to install and prepare for deployment**

Select the file path that represents the application root directory, archive file, exploded archive directory, or application module descriptor that you wish to install. The file path is displayed in the Path field.

**Note:** Only valid file paths are displayed below. If you cannot find your deployment files, upload your file(s) and/or confirm that your application contains the files.

**Path:** C:\Users\adouglas\workspace3.9.2\SAMLSwivelDemo\build\SAMLSwivelDemo.war

**Recently Used Paths:**

- C:\Users\adouglas\workspace3.9.2\SAMLSwivelDemo\build
- C:\Users\adouglas\workspace3.9.2\SAMLDemo\build
- C:\Users\adouglas\workspace3.9.2\SimpleDemo\build

**Current Location:** 192.168.0.42 \ C: \ Users \ adouglas \ workspace3.9.2 \ SAMLSwivelDemo \ build

SAMLSwivelDemo.jar

SAMLSwivelDemo.war

Back | Next | Finish | Cancel

Click Next, Next then Finish (using all the default options) to result in the following Screen:

**Messages**

- ✔ All changes have been activated. No restarts are necessary.
- ✔ The deployment has been successfully installed.

---

**Summary of Deployments**

**Control** | Monitoring

This page displays a list of Java EE applications and stand-alone application modules that have been installed to this domain, first selecting the application name and using the controls on this page.

To install a new application or module for deployment to targets in this domain, click the Install button.

[Customize this table](#)

**Deployments**

Install | Update | Delete | Start ▾ | Stop ▾

| <input type="checkbox"/> | Name ^   |
|--------------------------|--|
| <input type="checkbox"/> |  SAML Swivel Demo |

Install | Update | Delete | Start ▾ | Stop ▾

The Demo should now be accessible.

## 7.6 Additional Installation Options

## 8 Verifying the Installation

Open a web browser and enter the URL for the root of the demo. In this case: <http://weblogicserverURL:7001/SAMLSwivelDemo>

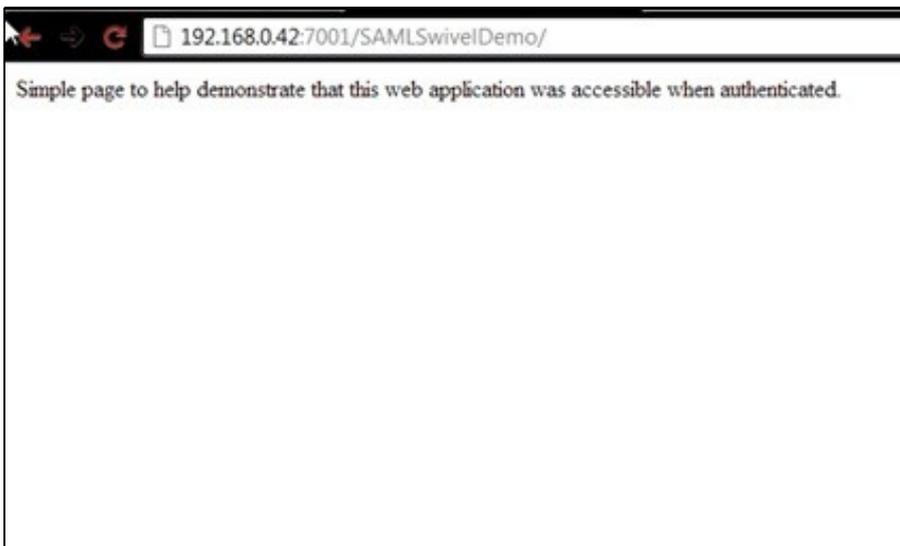
This will direct the user to the identity provider's login page as such:



A screenshot of a web browser showing the login page for SWIVEL. The browser's address bar displays the URL `192.168.0.42:8080/authenticationportal/identity_provider.jsp`. The page features the SWIVEL logo, which consists of a yellow cube icon and the text "SWIVEL the power of knowing". Below the logo, there are three input fields labeled "Username:", "Password:", and "OTC:". At the bottom of the form, there are two buttons: a blue "Login" button and a red "Refresh Image" button.

As per standard logon, enter the username and password (if required), start the session, enter the OTC and press ?Login?

If successful you will be authenticated and redirected to the SAMLDemo page as such:



## 9 Uninstalling the Swivel Integration

# 10 Troubleshooting

Check the Swivel logs

## 10.1 Enabling WebLogic debugging

To enable SAML logging On the WebLogic Administration console main menu select AdminServer->Configuration->Debug->Weblogic->Security->SAML2 and enable.

Now you can go to Diagnostics ->Log files ->ServerLog to view what is happening.

## 10.2 Error Messages

```
javax.security.auth.login.LoginException: [Security:090377]Identity Assertion Failed, weblogic.security.spi.IdentityAssertionException: [Security:090377]Identity Assertion Failed, weblogic.security.spi.IdentityAssertionException: [Security:096537]Assertion is not yet valid (NotBefore condition). at com.bea.common.security.internal.service.IdentityAssertionServiceImpl.assertIdentity(IdentityAssertionServiceImpl.java:89)
```

This has been seen where the time on the Swivel server is ahead of the WebLogics server. Ensure they both have the same time.

```
<BEA-000000> <[Security:096552]Illegal destination: https://<server_name>:<port>/saml2/sp/acs/post of assertion response.>
```

This is due to the Recipient destination value not matching the local (SP) assertion consumer URL. On the Weblogic Console => Environment => Servers => AdminServer => Configuration => Federation Services => SAM 2.0 General => disable ?Recipient Check Enabled? checkbox.

## 11 Known Issues and Limitations

## 12 Additional Information

For assistance in the Swivel installation and configuration please firstly contact your reseller and then email Swivel Secure support at [support@swivelsecure.com](mailto:support@swivelsecure.com).