Oracle® Health Sciences Information Manager

Policy Engine Installation and Configuration Guide

Release 2.0.1

E37112-02

October 2013

Oracle Health Sciences Information Manager (OHIM) leverages Integrating the Healthcare Enterprise (IHE) profiles, CONNECT reference architecture, and Oracle WebLogic to provide a broad range of international-standards-based web services to HIE applications in a management and performance optimized solution.

1 Getting Started

This section describes the minimum hardware and software requirements for installing Oracle Health Sciences Information Manager (OHIM) Policy Engine.

1.1 Hardware Requirements

The following are the hardware requirements for installing OHIM Policy Engine:

- 2 GB (2048 MB) of RAM
- 12 GB of Disk Space
- 16 GB of disk space for 64-bit

1.2 Software Requirements

The following are the software requirements for installing OHIM Policy Engine:

- Java 1.6 executable in path
- Oracle GlassFish Enterprise Server 2.1.1 Patch 14
- Oracle Enterprise Linux 5.5 or higher

Configuration Requirements

- Apache Ant 1.8.2 executable in path

```
PATH=$PATH:<install_dir>/apache-ant-1.8.2/bin
```

1.3 Downloading Oracle Health Sciences Information Manager Health Policy Engine

To download the Oracle Health Sciences Information Manager Health Policy Engine, perform the following tasks:

1. Navigate to http://edelivery.oracle.com.
2. Enter your Registration information, accept the Agreement Terms by selecting the check boxes, then click **Continue**.

3. From the **Select a Product Pack** drop-down menu, select **Health Sciences**.
4. From the **Platform** drop-down menu, select **Linux x86**.
5. Click **Go**.
6. Select **Oracle Health Sciences Information Manager Media Pack**.
7. Click **Continue**.
8. Click **Download** for the following and save the files to your system:
   - **Oracle Health Sciences Information Manager 2.0.1 Health Policy Engine**
9. Extract the files to view the *Oracle Health Sciences Information Manager Health Policy Engine Installation and Configuration Guide* and get the compressed tar file (`*.tgz`).

2 Installing the Policy Engine

Execute the following commands to install the Policy Engine:

1. `$ tar -zxvf ohim_hpe_installer.tgz`
2. `$ cd ohim_hpe_installer`
3. `$ java -jar ohim_hpe_installer.jar`
4. `$ cd <AS_INSTALLED_DIR>/bin`

> **Note:** AS_INSTALLED_DIR represents the directory where Oracle GlassFish Application server is installed.

5. `$ asadmin start-domain domain1`

To follow the prompts, refer to Appendix A: Running the Oracle Health Sciences Information Manager Policy Engine Installer.

3 Configuring the Policy Engine Components

3.1 Configuring the OpenSSO

To configure the OpenSSO, perform the following steps:

1. Open the following URL in a Web browser:

   `http://<POLICY_ENGINE_IP>:<HTTP_PORT>/opensso`

> **Note:** In the URL, ensure that you enter the IP address of system in stead of host name for <POLICY_ENGINE_IP>.

2. The Configurator starts and display the Configuration Options page. Click **Create Default Configuration**.
3. On the Default Configuration Options page, enter and confirm the following passwords:

- **Default User (amAdmin)**: `adminadmin`
- **Default Policy Agent (UrlAccessAgent)**: `password`

Note: You must change the default password after importing the configurations. For more information, refer to *Sun OpenSSO Enterprise 8.0 Administration Guide* at http://docs.oracle.com/cd/E19681-01/820-3885/index.html.

---

4. Click **Create Configuration** to continue.

   When the configuration is complete, the Configurator displays a link to the OpenSSO Enterprise Administration Console to perform any additional configuration required for your deployment.

5. (Optional step) If you have entered incorrect value for Policy Engine IP or Adapter host name during installation, you can use the following script to correct it. The information you enter here is updated in the import-svc-cfg.xml which is used by the script import-policyengine-svc-cfg.sh.

   1. Navigate to config directory:

      ```
      $ cd <install_dir>/addons/policyengine/config
      ```

   2. Modify install_pe.properties file to include the Oracle GlassFish installation directory path as the value for the as.install.dir property.

   3. Execute the script:

      ```
      $ sh config-opesso-config.sh
      ```

6. Modify the ssoadm utility file permissions:

   ```
   $ cd <install_dir>/addons/policyengine/config/files/opensso/tools/bin

   $ chmod 740 ssoadm
   ```

7. Import the service configuration data into OpenSSO configuration data store:

   ```
   $ cd <install_dir>/addons/policyengine/config
   ```

   Modify install_pe.properties file to include the Oracle GlassFish installation directory path as the value for the as.install.dir property.

   ```
   $ sh import-policyengine-svc-cfg.sh
   ```

8. Enter `y` for **Directory Service contains existing data. Do you want to delete it? [y | N]** and press the Enter key.

   You will see the following message on the console:

   ```
   Please wait while we import the service configuration...
   ```

   Upon successful completion of the service configuration import, you will see the message:

   ```
   Service Configuration is imported.
   ```

9. Restart the application server:

   ```
   $ cd <AS_INSTALLED_DIR>/bin

   $ asadmin stop-domain domain1

   $ asadmin start-domain domain1
   ```

10. Create a test user:

    1. Open the following URL in a browser:

       `http://POLICY_ENGINE_IP:8080/opensso`

    2. Enter the following credentials to log in to the OpenSSO administration console:

       User Name: `amAdmin`

       Password: `adminadmin`

    3. Under the Access Control tab, click the / (Top level Realm) in which you are creating the user.

    4. Click the **Subjects** tab.

    5. Click **New**.

    6. Enter data for the following fields:

       Note: Delete the default values of the **Last Name** and **Full Name** fields before you enter data.

       ID: `user1`

       First Name: `One`

       Last Name: `User`

       Full Name: `User, One`

       Password: `password`

       Password (Confirm): `password`

       User Status: `Active`

    7. Click **OK**.

       user1 is displayed in the list of users on the **Subjects** tab.

    8. Click **Back to Access Control**.

    9. On the **Access Control** page, click **LOG OUT**.

    10. Click **OK** if another dialog box is displayed regarding the need to close other associated windows.

3.2 Installing Self-signed Certificates on Policy Engine

This is an optional step. To install the self signed certificate for testing:

1. Stop the application server:

   ```
   $ cd <AS_INSTALLED_DIR>/bin
   $ asadmin stop-domain domain1
   ```

2. Execute create-and-import-selfsigned-certs.sh:

   ```
   $ cd <install_dir>/addons/policyengine/config
   $ sh create-and-import-selfsigned-certs.sh
   ```

3. Copy the Adapter VM's certificate from scp hiauser@ADAPTER_HOST:<appserver_install_dir>/domains/domain1/config/<ADAPTER_HOSTNAME.cer> to <appserver_install_dir>/domains/domain1/config.
4. Execute import-others-cert.sh to import Adapter certificate:

```
$ sh import-others-cert.sh
```

5. Start the application server:

```
$ cd <AS_INSTALLED_DIR>/bin
$ asadmin start-domain domain1
```

3.3 Configuring CONNECT on HIG Adapter System for OpenSSO based Policy Engine

To configure CONNECT on HIG adapter system for OpenSSO based Policy Engine, perform the following steps:

1. Log in to the Adapter system as `hiauser` (password: `hiapass`).
2. Copy ada_gw_pe_config.zip from Policy Engine system to a local directory. For example,

```
scp user@POLICY_ENGINE_HOST:<install_
dir>/addons/policyengine/config/ada_gw_pe_config.zip .
```

3. Ensure that the application server is not running. If it is running, stop it using the following commands:

```
$ cd <AS_INSTALLED_DIR>/bin
$ asadmin stop-domain domain1
```

4. Navigate to ada_gw_pe_config.zip and unzip the file:

```
$ cd <FILE_PATH>
$ unzip ada_gw_pe_config.zip -d ada_gw_pe_config
$ cd ada_gw_pe_config
```

5. Modify install_pe.properties file to include the Oracle GlassFish installation directory path as the value for the as.install.dir property.
6. Execute config-adapter-policyengine.sh to configure the CONNECT Adapter which enables it to interact with openSSO-based Policy Engine:

```
$ sh config-adapter-policyengine.sh
```

7. Start the application server:

```
$ cd <AS_INSTALLED_DIR>/bin
$ asadmin start-domain domain1
```

3.4 Creating Consumer Preferences Document Using SoapUI

This section assumes the following have already been setup, and applications/services on the OHIG Adapter and Gateway are ready to test from the SoapUI project.

- OpenSSO Instance has been installed and configured on Policy Engine system
- Oracle GlassFish Application Server on Policy Engine system is up and running

- OHIG Gateway and Adapter are configured to interact with Policy Engine system for authentication/authorization services
- Oracle GlassFish Application Servers on OHIG Gateway system, and OHIG Adapter system are up and running
- A test system with SoapUI application installed on it

1. If Oracle GlassFish Application Server is not running on any of the systems, start it by using the following commands:
   a. `$ cd <AS_INSTALLED_DIR>/bin`
   b. `$ asadmin start-domain domain1`
2. Launch the SoapUI application on the test machine.
3. Copy the SoapUI project file from Policy Engine system, to a directory on the test system. For example,

   ```
   scp user@POLICY_ENGINE_HOST:<install_
   dir>/addons/policyengine/config/files/opensso/soapui/AdapterPEPWS-soapu
   i-project.xml.
   ```
4. From the **File** menu, click **Import Project**. This will display the Select soapUI Project Files window.
5. Enter *<FILEPATH>*`/AdapterPEPWS-soapui-project.xml` as the filename.

   Note: `FILE_PATH` represents the absolute path to which the `AdapterPEPWS-soapui-project.xml` file has been copied.

6. Click **Open**. The `AdapterPEPWS-soapui-project.xml` file is imported into your soapUI application.
7. Open the test by selecting **AdapterPEPWS** -> **Adapter PIP Binding Soap** -> **StorePtConsent** -> **StorePatientConsent1**.

   Note: While testing using the default CONNECT Adapter provided Master Patient Index (`mpi.xml`), use the Patient ID: `D123401`.

   To ensure that the patient consent is not changed during SoapUI testing, make the following changes to the endpoint URL. Perform the following for this step (**StorePatientConsent1**) and step 11 (**StorePatientConsent2**).

   Note: If you use a database-based repository, you do not need to change the endpoint URL for either step.

   To update `<AS_INSTALLED_DIR>/domains/domain1/config/nhin/internalConnectionInfo.xml` from the OHIG Adapter and Gateway servers, replace:

   ```
   <businessService serviceKey="uddi:testnhincnode:adapterxdsbdocrepository"
   businessKey="uddi:testnhieonenode:1.1">
   <name xml:lang="en">adapterxdsbdocrepository</name>
   <bindingTemplates>
   ```

```
<bindingTemplate bindingKey="uddi:testnhincnode:adapterxdsbdocrepository"
serviceKey="uddi:testnhincnode:adapterxdsbdocrepository">
<accessPoint useType="endPoint">http://<hig_adapter_
IP>:8080/CONNECTAdapterDocReposSoap12/AdapterDocRepository2Soap12Service</acces
sPoint>
<categoryBag>
<keyedReference tModelKey="uddi:nhin:versionofservice" keyName=""
keyValue="1.0"/>
<keyedReference tModelKey="CONNECT:adapter:apilevel" keyName=""
keyValue="LEVEL_a0"/>
</categoryBag>
</bindingTemplate>
</bindingTemplates>
<categoryBag>
<keyedReference tModelKey="uddi:nhin:standard-servicenames"
keyName="adapterxdsbdocrepository" keyValue="adapterxdsbdocrepository"/>
</categoryBag>
</businessService>
```

with

```
<businessService serviceKey="uddi:testnhincnode:adapterxdsbdocrepository"
businessKey="uddi:testnhieonenode:1.1">
<name xml:lang="en">adapterxdsbdocrepository</name>
<bindingTemplates>
<bindingTemplate bindingKey="uddi:testnhincnode:adapterxdsbdocrepository"
serviceKey="uddi:testnhincnode:adapterxdsbdocrepository">
<accessPoint useType="endPoint">http://<hig_adapter_
IP>:8080/CONNECTAdapter/DocumentRepository_Service</accessPoint>
<categoryBag>
<keyedReference tModelKey="uddi:nhin:versionofservice" keyName=""
keyValue="1.0"/>
<keyedReference tModelKey="CONNECT:adapter:apilevel" keyName=""
keyValue="LEVEL_a0"/>
</categoryBag>
</bindingTemplate>
</bindingTemplates>
<categoryBag>
<keyedReference tModelKey="uddi:nhin:standard-servicenames"
keyName="adapterxdsbdocrepository" keyValue="adapterxdsbdocrepository"/>
</categoryBag>
</businessService>
```

After performing this update, restart the OHIG Adapter and Gateway Oracle GlassFish servers.

8. In the StorePatientConsent1 window, using the edit current option, set the endpoint URL for the request by using the correct IP address of OHIG Adapter.
9. Execute the test by clicking the **Submit request to specified endpoint URL** (Green arrow) near the top left corner of the StorePatientConsent1 window.
10. Execute the test **AdapterPEPWS** -> **AdapterPIPBindingSoap** -> **RetrievePtConsentByPtId** -> **RetrievePatientConsent** to verify that the document was stored successfully.
11. Update the patient preference by modifying the **StorePatientConsent2** (**AdapterPEPWS** -> **AdapterPIPBindingSoap** -> **StorePtConsent** -> **StorePatientConsent2**) SOAP request where you use "false" for the "optIn"

element, and include the policyOID element, which can be found in the response of the **RetrievePatientConsent** request.

The modified request looks like:

```
<soapenv:Envelope xmlns:soapenv="http://www.w3.org/2003/05/soap-envelope"
xmlns:urn="urn:gov:hhs:fha:nhinc:common:nhinccommonadapter">
<soapenv:Header/>            <soapenv:Body>
<urn:StorePtConsentRequest>          <urn:patientPreferences>
<urn:patientId>0000000000</urn:patientId>
<urn:assigningAuthority>1.1</urn:assigningAuthority>
<urn:optIn>false</urn:optIn>          <urn:fineGrainedPolicyMetadata>
<urn:policyOID>20.200.20.31</urn:policyOID>
</urn:fineGrainedPolicyMetadata>           </urn:patientPreferences>
</urn:StorePtConsentRequest>         </soapenv:Body>
</soapenv:Envelope>
```

Execute the modified **StorePatientConsent2** request. This will update the patient's preference.

12. Open the test **AdapterPEPWS** -> **AdapterPEPBindingSoap** -> **CheckPolicy** -> **DocumentQueryIn**. Use "false" for the "optIn" element.

    Note: While testing using the default CONNECT Adapter provided Master Patient Index (`mpi.xml`), for the `resource-id` attribute, use the string `D123401` as the attribute value.

13. In the DocumentQueryIn window, using the edit current option, edit the endpoint URL for the request by using the IP address of OHIG Adapter system.
14. Execute the test by clicking the **Submit request to specified endpoint URL** (Green arrow) near the top left corner of the DocumentQueryIn window. You will observe "Deny" in the response.
15. Execute the SOAP request **AdapterPEPWS** -> **AdapterPIPBindingSoap** -> **StorePtConsent** -> **StorePatientConsent2**. This time use "true" for the "optIn" element. This will again update the patient's preference.
16. Execute the test again **AdapterPEPWS** -> **AdapterPEPBindingSoap** -> **CheckPolicy** -> **DocumentQueryIn**. This time you will observe “Permit” in the response.

3.5 Validating CONNECT on OHIG Adapter

To validate the CONNECT software on the OHIG Gateway and Adapter machines after Adapter is configured to use openSSO Policy Engine:

1. Ensure that Oracle GlassFish Application Server is up and running on Policy Engine, Gateway, and Adapter machines using the following commands:
   a. `$ cd <AS_INSTALLED_DIR>/bin`
   b. `$ asadmin start-domain domain1`
2. Validate the configuration using the sample universal client distributed with the Adapter:
   a. Launch the application by navigating to the following URL:

```
http://<adapter_host_ip>:8080/
CONNECTUniversalClientGUI/
```

The authentication page is displayed asking for user account details.

b. Enter a valid user name and password (`user1/password`).

c. Click the **Login** button.

- If the account details are correct, the Universal Client GUI Main page has the **patient search** tab enabled, while the rest of the tabs are disabled.

- If the provided account details are incorrect, you will be prompted to enter the correct account details again.

d. Search for a patient with the last name: “Younger”.

e. If the installation is correct, this returns a page with the PatientId for the patient.

f. Click the PatientId hyperlink for additional details on the patient.

g. The **Document** tab is now enabled and you can search for patient documents by date range. Search for date range 08/01/2000 to 08/01/2010.

h. Click on the document URL to retrieve the document.

4 Appendix A: Running the Oracle Health Sciences Information Manager Policy Engine Installer

```
$ cd <install_dir>
$ java -jar ohim_hpe_installer.jar
Oracle HIM HPE Installer 2.0.1.0
-- Command
Choose option install_command (usage, version, install)
$ install
-- Enter the IP address of the Policy Engine system.
Enter policyengine_ip [127.0.0.1]
$ 10.145.240.58
-- Enter the IP address of the Connect adapter host.
Enter connectadapter_host [localhost]
$ 10.149.56.38
-- Stop glassfish
Choose option stop_glassfish ([yes], no)
$ yes
```

5 Documentation Accessibility

For information about Oracle's commitment to accessibility, visit the Oracle Accessibility Program website at http://www.oracle.com/pls/topic/lookup?ctx=acc&id=docacc.

Access to Oracle Support

Oracle customers have access to electronic support through My Oracle Support. For information, visit http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info or visit http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs if you are hearing impaired.

Oracle Health Sciences Information Manager Policy Engine Installation and Configuration Guide, Release 2.0.1
E37112-02

Copyright © 2010, 2013, Oracle and/or its affiliates. All rights reserved.

This software and related documentation are provided under a license agreement containing restrictions on use and disclosure and are protected by intellectual property laws. Except as expressly permitted in your license agreement or allowed by law, you may not use, copy, reproduce, translate, broadcast, modify, license, transmit, distribute, exhibit, perform, publish, or display any part, in any form, or by any means. Reverse engineering, disassembly, or decompilation of this software, unless required by law for interoperability, is prohibited.

The information contained herein is subject to change without notice and is not warranted to be error-free. If you find any errors, please report them to us in writing.

If this is software or related documentation that is delivered to the U.S. Government or anyone licensing it on behalf of the U.S. Government, the following notice is applicable:

U.S. GOVERNMENT END USERS: Oracle programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, delivered to U.S. Government end users are "commercial computer software" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, use, duplication, disclosure, modification, and adaptation of the programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, shall be subject to license terms and license restrictions applicable to the programs. No other rights are granted to the U.S. Government.

This software or hardware is developed for general use in a variety of information management applications. It is not developed or intended for use in any inherently dangerous applications, including applications that may create a risk of personal injury. If you use this software or hardware in dangerous applications, then you shall be responsible to take all appropriate fail-safe, backup, redundancy, and other measures to ensure its safe use. Oracle Corporation and its affiliates disclaim any liability for any damages caused by use of this software or hardware in dangerous applications.

Oracle and Java are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

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. AMD, Opteron, the AMD logo, and the AMD Opteron logo are trademarks or registered trademarks of Advanced Micro Devices. UNIX is a registered trademark of The Open Group.

This software or hardware and documentation may provide access to or information on content, products, and services from third parties. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to third-party content, products, and services. Oracle Corporation and its affiliates will not be responsible for any loss, costs, or damages incurred due to your access to or use of third-party content, products, or services.