

Oracle® Enterprise Repository
Oracle Registry Repository Exchange Utility
10g Release 3 (10.3)

July 2009

Copyright © 2008, 2009, 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 software or related documentation 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 RIGHTS Programs, software, databases, and related documentation and technical data delivered to U.S. Government customers are "commercial computer software" or "commercial technical data" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, the use, duplication, disclosure, modification, and adaptation shall be subject to the restrictions and license terms set forth in the applicable Government contract, and, to the extent applicable by the terms of the Government contract, the additional rights set forth in FAR 52.227-19, Commercial Computer Software License (December 2007). Oracle USA, Inc., 500 Oracle Parkway, Redwood City, CA 94065.

This software 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 which may create a risk of personal injury. If you use this software in dangerous applications, then you shall be responsible to take all appropriate fail-safe, backup, redundancy, and other measures to ensure the safe use of this software. Oracle Corporation and its affiliates disclaim any liability for any damages caused by use of this software in dangerous applications.

Oracle is a registered trademark of Oracle Corporation and/or its affiliates. Other names may be trademarks of their respective owners.

This software 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.

Contents

1. Getting Started With the Oracle Registry Repository Exchange Utility

What is the Oracle Registry Repository Exchange Utility?	1-2
Valid Metadata Entities	1-2
Example Use Cases	1-3
Related Documentation	1-4

2. Configuring the Oracle Registry Repository Exchange Utility

Installing and Configuring the Oracle Registry Repository Exchange Utility	2-2
Install the Oracle Registry Repository Exchange Utility	2-2
Import the Registry Repository Exchange Utility Solution Pack Into Oracle Enterprise Repository	2-2
Verify the Oracle Enterprise Repository UDDI System Settings	2-3
Configure the Service and SOA Business Entity Asset Types	2-4
Configuring the Oracle Registry Repository Exchange Utility Configuration File	2-4
Setting the Repository Connection Information	2-4
Setting the Registry Connection Information	2-5
Setting the Repository Query	2-5
Query by Name	2-6
Query by Registration Status	2-6
Query by Categorizations	2-6
Query by Endpoint Lifecycle Stage	2-6

Setting the Destination Registries	2-7
Setting the Registry Query	2-7
Setting the Source Registry	2-8
Configuring Oracle Enterprise Repository Categorizations in the UDDI Mappings File	2-8
Understanding the Oracle Registry Repository Exchange Utility's Property File	2-9
Encrypting the Configuration File Passwords	2-10

3. Using the Oracle Enterprise Repository Exchange Utility

Running the Oracle Registry Repository Exchange Utility	3-2
Invoking the Oracle Registry Repository Exchange Utility Using Workflows	3-3
Synchronizing Using Timer Based Workflows	3-4
Synchronizing When Events are Triggered	3-4
Example Use Cases	3-5
How the Exchanged Metadata Is Synchronized.	3-5
Synchronizing the Metadata Published from Oracle Enterprise Repository to Oracle Service Registry	3-5
Business Entities	3-6
Endpoint	3-6
Categorizations	3-6
Registration and Active Status	3-7
Sample Flow of Metadata from Oracle Enterprise Repository to Oracle Service Registry	3-9
Synchronizing the Metadata from Oracle Service Registry to Oracle Enterprise Repository	3-11
Business Entities	3-11
Endpoint	3-11
Categorizations	3-11

Sample Flow of Metadata from Oracle Service Registry to Oracle Enterprise Repository	3-12
Web Service Endpoint Management	3-14
Searching for Oracle Service Registry Exchanged Metadata in Oracle Enterprise Repository	3-15
Checking the Oracle Registry Repository Exchange Utility Log File	3-17
Known Issues	3-17
Resynchronizing Oracle Service Registry Services	3-17
Import of Oracle Service Bus WSDLs	3-17
Workaround of Solution	3-18

Getting Started With the Oracle Registry Repository Exchange Utility

This section contains information on the following subjects:

- [“What is the Oracle Registry Repository Exchange Utility?”](#) on page 1-2
- [“Example Use Cases”](#) on page 1-3
- [“Related Documentation”](#) on page 1-4

What is the Oracle Registry Repository Exchange Utility?

The Oracle Registry Repository Exchange Utility synchronizes Oracle Enterprise Repository and Oracle Service Registry bi-directionally so that the metadata from either of these products can flow in either direction through the utility. The following are the metadata entities that are handled by the utility.

The Oracle Registry Repository Exchange Utility is capable of:

- Publishing services and endpoints from design-time to the run-time environment using UDDI.
- Submitting newly discovered run-time services, and endpoints to the repository so that they can be managed and governed.
- Communicating service performance information that is deposited into the UDDI registry back into the repository to better inform prospective service consumers and portfolio managers.

Valid Metadata Entities

The following metadata entities are handled by the Oracle Registry Repository Exchange Utility:

- All UDDI enabled services and SCA Service types, as well any custom Services of any Oracle Enterprise Repository Asset Type developed by end users.
- Business Entities that provides Business Services. (See [“Synchronizing the Metadata Published from Oracle Enterprise Repository to Oracle Service Registry”](#) on page 3-5 and [“Synchronizing the Metadata from Oracle Service Registry to Oracle Enterprise Repository”](#) on page 3-11 for detailed information about how the Oracle Registry Repository Exchange Utility will arrive at the Business Entities.)
- Endpoint assets that are linked to the Services that provide access point to the services. For example, there can be multiple endpoints that can be tagged as *staging* or *production* and are mapped to the UDDI binding template appropriately.
- Oracle Enterprise Repository Categorizations are mapped to UDDI t-models. When a categorization is applied to a Service asset, an appropriate entry is added to the UDDI Category Bag and is linked to the appropriate taxonomy t-model. These t-models are also automatically loaded into Oracle Service Registry the first time they are encountered, or they can be loaded manually by the Oracle Enterprise Repository Exchange Utility (see the `-publish_tmodel` option in [Table 3-1](#)).

- Service Registration Status and Active Status are added as items to the Category Bag of the Business Service in Oracle Service Registry when the services are pushed to Oracle Service Registry.

Example Use Cases

Because Oracle Registry Repository Exchange Utility is bi-directional, it is able to support the following use cases:

- When a service in Oracle Service Registry is modified, such as when more endpoints are added to it (for example, for staging and production environments), the Oracle Registry Repository Exchange Utility can relate the service with one or more endpoints in Oracle Enterprise Repository.
- When Business Services are moved to different Business Entities in Oracle Enterprise Repository using asset relationships, the new relationship is also reflected in Oracle Service Registry when the Services are re-synchronized.
- When synchronizing Oracle Enterprise Repository metadata with Oracle Service Registry, the Oracle Registry Repository Exchange Utility merges any changes so that endpoint changes are preserved.
- Any two arbitrary Services that were published to Oracle Enterprise Repository and Oracle Service Registry can be linked using the Oracle Registry Repository Exchange Utility. For example, if a Service is published to the repository using the Oracle Enterprise Repository plug-ins for Eclipse and the same Service is published to Oracle Service Registry using Oracle Service Bus, these two services can be linked using the Oracle Registry Repository Exchange Utility. Once the Services are linked, they can be bi-directionally synchronized.
- The endpoints of matched services can be filtered based on the specified Asset Lifecycle of the endpoints, so that only the matched endpoints are published to the repository. This query is useful when there are separate registries: one that lists the staged endpoints and another that lists the production endpoints.

Note: The Oracle Registry Repository Exchange Utility can be invoked by the Oracle Enterprise Repository workflows, such that synchronization between Oracle Enterprise Repository and Oracle Service Registry can be triggered by an asset lifecycle stage change, or by any other event trigger. See [Oracle Enterprise Repository Configuring and Managing Advanced Registration Flows](#) for more information.

Related Documentation

- Oracle Enterprise Repository documentation set - provides the tools to manage and govern the metadata for any type of software asset, from business processes and services to patterns, frameworks, applications, components, and data services.
- Oracle Enterprise Repository on OTN – The home page for Oracle Enterprise Repository on Oracle Technology Network (OTN) is:

<http://www.oracle.com/technologies/soa/enterprise-repository.html>

- Architect Center: SOA Governance: Essential to Your Business - Learn how effective SOA governance is an essential element in any enterprise transformation strategy by reading the Architect Center: SOA Governance: Essential to Your Business documents at:

<http://www.oracle.com/technology/architect/soa/soagov/index.html>

- SOA Blog - Keep on top of the latest SOA blogs at:

<http://blogs.oracle.com/governance>

Configuring the Oracle Registry Repository Exchange Utility

This section contains the following topics:

- “Installing and Configuring the Oracle Registry Repository Exchange Utility” on page 2-2
- “Configuring the Oracle Registry Repository Exchange Utility Configuration File” on page 2-4
- “Configuring Oracle Enterprise Repository Categorizations in the UDDI Mappings File” on page 2-8
- “Understanding the Oracle Registry Repository Exchange Utility’s Property File” on page 2-9
- “Encrypting the Configuration File Passwords” on page 2-10

Installing and Configuring the Oracle Registry Repository Exchange Utility

Before you can use the Oracle Registry Repository Exchange Utility to publish and receive Oracle Enterprise Repository metadata to and from Oracle Service Registry, you must complete the following configuration steps.

Install the Oracle Registry Repository Exchange Utility

The `alrrxExchangeUtility.zip` is packaged in the `OER10.3SupInteg.Doc_Data.zip` file. This file includes the necessary Oracle Registry Repository Exchange Utility files for the Oracle Enterprise Repository and Oracle Service Registry integration, the Registry Repository Exchange Utility Solution Pack, and the documentation.

You can unzip the `alrrxExchangeUtility.zip` into the directory on your file system where Oracle Enterprise Repository is installed, typically `BEA_HOME\repository103`. When the zip file containing the Oracle Registry Repository Exchange Utility is unzipped to your file system, it creates the following structure.

```
alrrx <ExchangeUtility Tool Home>
|
|   datapack
|   docs
|   lib
```

Within the `<ExchangeUtility Tool Home>` directory, you will find the Oracle Registry Repository Exchange Utility files, such as the `alrrx.xml`, `UDDIMappings.xml`, `alrrx.properties`, `alrrx.bat`, and `encrypt.bat` files.

Import the Registry Repository Exchange Utility Solution Pack Into Oracle Enterprise Repository

The required Registry Repository Exchange Utility Solution Pack is bundled with the `alrrxExchangeUtility.zip`. Follow these steps to import the Registry Repository Exchange Utility Solution Pack (`OER103-RR-ExchangeUtility-Solution-Pack.zip`) into Oracle Enterprise Repository.

1. Unzip the `alrrxExchangeUtility.zip` file onto your file system.

2. Start the Oracle Enterprise Repository Import/Export tool, as described in the [Oracle Enterprise Repository Import/Export Guide](#).
3. Select the **Import** tab.
4. Navigate to the `<ExchangeUtility Tool Home>\datapack` directory.
5. Select `OER103-RR-ExchangeUtility-Solution-Pack.zip` as the target file to import into Oracle Enterprise Repository.
6. Click **Next**, and then click **Next** again to start the import process.
7. Click **Finish** to complete the process.

Verify the Oracle Enterprise Repository UDDI System Settings

Verify that the UDDI server and Web service plug-in are enabled.

1. Open the Oracle Enterprise Repository **Admin** page.
2. Click the **System Settings** option in the left pane.
3. Navigate to the **External Integration > UDDI** section.
Note: You can also search for **UDDI**.
4. Verify the following UDDI property values:
 - **UDDI/cmee.uddi.enabled** is set to **True** to enable the UDDI Web service plug-in.
 - **UDDI/cmee.uddi.server enabled** is set to **True** to enable the repository to act as a UDDI registry for certain applications.
 - **cmee.uddi.business.service.relationship** is set to **Provided by Business Entity** — the relationship between Service and Business Entity asset types.
 - **cmee.import.uddi.service.assettype** is set to **Service** — the Service asset type.
 - **cmee.uddi.default.business** is set to a **UDDI Node** — only when publishing services to Oracle Service Registry, when the asset is not linked to a Business Entity.
5. If necessary, click **Save**.

For more information on configuring System Settings, see the [Oracle Enterprise Repository Administration Guide](#).

Configure the Service and SOA Business Entity Asset Types

Using the Oracle Enterprise Repository Type Manager, enable the UDDI options for the Service and SOA Business Entity asset types, as follows:

1. Launch the Asset Editor by clicking **Edit/Manage Assets** on the Assets page.
2. Launch the Type Manager by selecting **Manage Types** on the **Actions** menu.
3. Under the Asset Types folder, select the **Service** asset type to edit its default configuration, as follows:
 - a. Change the UDDI setting to **UDDI Business Service Entity**.
 - b. Save the change by selecting **Save** on the File menu.
4. Under the Asset Types folder, select the **SOA Business Entity** asset type to edit its default configuration, as follows:
 - a. Change the UDDI setting to **UDDI Business Entity**.
 - b. Save the change by selecting **Save** on the File menu.
5. Exit the Type Manager, and then exit the Asset Editor.

For more information on using the Type Manager, see the [Oracle Enterprise Repository Registrar Guide](#).

Configuring the Oracle Registry Repository Exchange Utility Configuration File

This section describes how to configure the Oracle Registry Repository Exchange Utility configuration file for your environment.

Setting the Repository Connection Information

Open the `alrrx.xml` file located at `<ExchangeUtility Tool Home>` and modify the following XML section so that it points to your Oracle Enterprise Repository instance with the appropriate credentials.

```
<repository>
  <uri>http://localhost:7101/oer</uri>
<credentials>
  <user>admin</user>
```

```

    <password>admin</password>
  </credentials>
</repository>

```

where URI = Oracle Enterprise Repository URI, using the following format:

```
http://<host>:<port>/<Oracle Enterprise Repository web app name>
```

For security purposes, the password can be encrypted, as described in [“Encrypting the Configuration File Passwords” on page 2-10](#).

Setting the Registry Connection Information

The Oracle Registry Repository Exchange Utility can publish to one or more registries and can read from multiple registries (it requires separate transactions to read from each registry). The first step is to create one more `<registry>` node with the connection info, as shown here.

```

<registries>
  <registry name="alsr">
    <inquiryURI>http://localhost:7001/registry/uddi/inquiry</inquiryURI>
    <publishURI>http://localhost:7001/registry/uddi/publishing</publishURI>
    <securityURI>http://localhost:7001/registry/uddi/security</securityURI>
    <credentials>
      <user>admin</user>
      <password>admin</password>
    </credentials>
  </registry>
  <registry name="alsr2">
    <inquiryURI>http://localhost:7201/registry/uddi/inquiry</inquiryURI>
    <publishURI>http://localhost:7201/registry/uddi/publishing</publishURI>
    <securityURI>http://localhost:7201/registry/uddi/security</securityURI>
    <credentials>
      <user>admin</user>
      <password>admin</password>
    </credentials>
  </registry>
</registries>

```

Setting the Repository Query

The following configuration snippets demonstrate how to build a query to run against Oracle Enterprise Repository and receive the list of Services that should be published to Oracle Service Registry. This filters the services to be pushed to Oracle Service Registry in the form of a query. There are a number of ways that services can be queried and you can create one or more queries.

Query by Name

When the `<services>` element is configured, the service name specified is published to Oracle Service Registry. However, due to a limitation in the Oracle Enterprise Repository REX API, only one `<services>` element can be added.

```
<query>
  <repositoryQuery>
    <services>
      <service name="HelloWorld" />
    </services>
    <registrationStatus></registrationStatus>
    <serviceCategorizations type="AssetLifecycleStage" value="" />
    <endpointAssetLifecycleStatus></endpointAssetLifecycleStatus>
```

Query by Registration Status

When the `<registrationStatus>` element is configured, only the services with the specified Registration Status are published. For example, if this field is set to *Registered*, then only registered services will be published to Oracle Service Registry, while ignoring all other matched services that are not in this state.

```
<registrationStatus>Registered</registrationStatus>
```

Query by Categorizations

When the `<serviceCategorizations>` element is configured, only the services with the specified categorization will be published. For example, when using the following categorization, only the *Recommended* services will be published to Oracle Service Registry.

```
<serviceCategorizations type="classification" value="Recommended" />
```

Query by Endpoint Lifecycle Stage

When the `<endpointAssetLifecycleStatus>` element is configured, the endpoints of matched services can be filtered based on the specified Asset Lifecycle of the endpoints, and only the matched endpoints will be published to Oracle Service Registry. For example, if there are two endpoints attached to a service, one with the Asset Lifecycle Stage of *Stage 3 – Build* and one with *Stage 4 – Release*, only the endpoint with the *Stage 3 – Build Asset Lifecycle* is published.

```
<endpointAssetLifecycleStatus>Stage 3 -
Build</endpointAssetLifecycleStatus>
```


This query is useful when there are separate registries: one that lists the staged endpoints and another that lists the production endpoints.

Setting the Destination Registries

The following configuration snippet demonstrates how to use the `<destinationRegistries>` element to configure one or more destination registries where the matched Oracle Enterprise Repository Services will go. These registries are used when Services are picked from Oracle Enterprise Repository and are moved to Oracle Service Registry (i.e., Oracle Enterprise Repository > Oracle Service Registry).

```
<destinationRegistries>
  <destinationRegistry>alsr</destinationRegistry>
  <destinationRegistry>alsr2</destinationRegistry>
</destinationRegistries>
```

Setting the Registry Query

The following configuration snippet demonstrates how to use the `<registryQuery>` element to build a query to run against Oracle Service Registry and receive the list of services that need to be fetched from Oracle Service Registry and placed in Oracle Enterprise Repository.

```
<registryQuery>
  <businessEntities>
    <businessEntity name="Account Services" />
    <businessEntity />
  </businessEntities>
  <services>
    <service name="AddCustomerService%" />
  </services>
  <qualifiers>
    <qualifier>approximateMatch</qualifier>
  </qualifiers>
<sourceRegistry>alsr</sourceRegistry>
</registryQuery>
```

Follow these configuration guidelines:

- Make sure that the `businessEntities` `name` or `service` `name` value is not empty.
- For the `businessEntities` `name`, the *exact* name must be specified.

Configuring the Oracle Registry Repository Exchange Utility

- For the `service name`, at least one wildcard character should be used. For example, to get all services specify “%”.
- Search criteria for an Oracle Service Registry query is case-sensitive.

Services can be searched in the following way:

- Search by one or more Service names. The Service names can be a wildcard if the qualifier is approximate. For example, if the service name is “Google”, any service that starts with “Google” will be fetched and placed in Oracle Enterprise Repository.
- Select one or more Business Entity and all the services in those Business Entities will be fetched and placed in Oracle Enterprise Repository. The Business Entity name has to be exact; the wild card is supported only for Services and not for Business Entities.

Warning: If both Business Entity query and Service query are specified, the Business Entity query will override the Service query.

Setting the Source Registry

The `<sourceRegistry>` element tells the registry where the Services will be picked and placed in Oracle Enterprise Repository. This registry is used when Services are picked from Oracle Service Registry and they move to Oracle Enterprise Repository (i.e., Oracle Service Registry > Oracle Enterprise Repository).

```
<sourceRegistry>alsr</sourceRegistry>
```

Configuring Oracle Enterprise Repository Categorizations in the UDDI Mappings File

Prior to publishing assets to Oracle Service Registry, Oracle Enterprise Repository Categorizations are mapped in the UDDI Mappings file (`UDDIMappings.xml`) that is stored in the `<ExchangeUtility Tool Home>` directory, as shown in the following XML snippet:

```
<uddi:uddiSettings
xmlns:uddi="http://www.bea.com/aler/integration/config/uddi">
  <categorizationMappings>
    <categorizationType alerCategorizationTypeName="AssetLifecycleStage"
alerCategorizationTypeId="112"
uddiCategoryTModelKey="uddi:bea.com:aler:categorization:AssetLifecycleStage">
      <categorization alerCategorization="Stage 1 - Propose" uddiKeyName="Stage
1 - Propose" uddiKeyValue="Stage 1 - Propose" />
      <categorization alerCategorization="Stage 2 - Plan" uddiKeyName="Stage 2
- Plan" uddiKeyValue="Stage 2 - Plan" />
    </categorizationType>
  </categorizationMappings>
</uddi:uddiSettings>
```

```

    <categorization alerCategorization="Stage 3 - Build" uddiKeyName="Stage 3
- Build" uddiKeyValue="Stage 3 - Build" />
    <categorization alerCategorization="Stage 4 - Release" uddiKeyName="Stage
4 - Release" uddiKeyValue="Stage 4 - Release" />
    <categorization alerCategorization="Stage 5 - Target For Retirement"
uddiKeyName="Stage 5 - Target For Retirement" uddiKeyValue="Stage 5 - Target For
Retirement" />
    <categorization alerCategorization="Stage 6 - Retirement"
uddiKeyName="Stage 6 - Retirement" uddiKeyValue="Stage 6 - Retirement" />

```

An Oracle Enterprise Repository Categorization will be honored in Oracle Service Registry only if a corresponding mapping is present in the UDDI Mappings file; otherwise, the Categorization will simply be ignored. Therefore, if a new asset Categorization is created in Oracle Enterprise Repository, you must regenerate the UDDI Mappings file for that Categorization to be honored in Oracle Service Registry.

Understanding the Oracle Registry Repository Exchange Utility's Property File

This section describes the properties in the Property file (`alrrx.properties`) file that is stored in the *<ExchangeUtility Tool Home>* directory. Some properties already exist in the Oracle Enterprise Repository System Settings and some of the properties are new for the Oracle Registry Repository Exchange Utility.

- `cmee.uddi.service.endpoint.relationship=Deployment-Deployed` — relationship between Service and Endpoint.
- `cmee.import.uddi.business.assettype=SOA - Business Entity` — Business Entity asset type.
- `cmee.import.uddi.accesspoint.assettype=Endpoint: Web Service` — Endpoint asset type.
- `cmee.import.uddi.artifactwsdl.relationship=Sync-Defines` — relationship between Service and WSDL artifact.
- `cmee.import.uddi.receiver.batch.size=100` — Only when receiving from Oracle Service Registry, this determines the batch size.

Caution: The following properties will only be used if the corresponding property is not set in the Oracle Enterprise Repository System Settings. If the Oracle Enterprise Repository System Settings property is configured, that setting will override the property in the `alrrx.properties` file.

Configuring the Oracle Registry Repository Exchange Utility

- `cmee.uddi.business.service.relationship=BusinessService` — relationship between Service and Business Entity asset types.
- `cmee.import.uddi.service.assettype=Service` — Service asset type.
- `cmee.uddi.default.business=A UDDI Node` — only when publishing services to Oracle Service Registry, when the asset is not linked to a Business Entity.

For more information about other Oracle Enterprise Repository System Settings, see the [Oracle Enterprise Repository Administration Guide](#).

Encrypting the Configuration File Passwords

For enhanced security, the password encryption tool (`encrypt.bat`) allows you to encrypt the passwords that are stored in the Oracle Registry Repository Exchange Utility configuration (`alrrx.xml`) file.

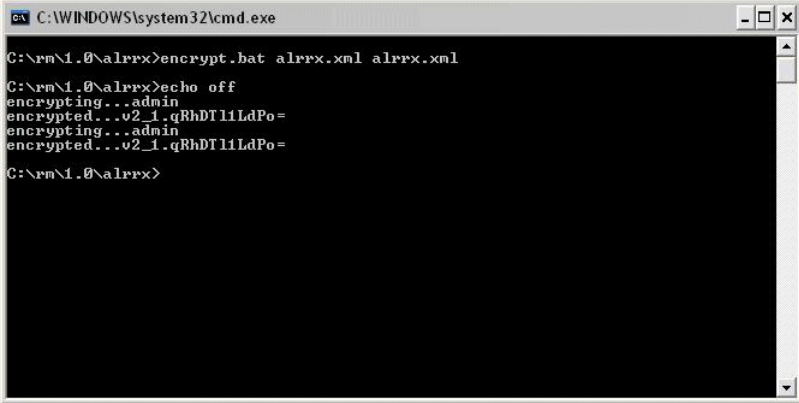
1. Navigate to the `<ExchangeUtility Tool Home>` directory.
2. From a command prompt, run the password encryption tool as follows:

```
> encrypt.bat alrrx.xml alrrx.xml
```

where:

`alrrx.xml` = the Oracle Registry Repository Exchange Utility configuration file

Figure 2-1 Encrypt Password Tool



```
C:\WINDOWS\system32\cmd.exe
C:\rpm\1.0\alrrx>encrypt.bat alrrx.xml alrrx.xml
C:\rpm\1.0\alrrx>echo off
encrypting...admin
encrypted...v2_1.qRhDT11LdPo=
encrypting...admin
encrypted...v2_1.qRhDT11LdPo=
C:\rpm\1.0\alrrx>
```

Using the Oracle Enterprise Repository Exchange Utility

This section contains the following topics:

- [“Running the Oracle Registry Repository Exchange Utility”](#) on page 3-2
- [“How the Exchanged Metadata Is Synchronized”](#) on page 3-5
- [“Searching for Oracle Service Registry Exchanged Metadata in Oracle Enterprise Repository”](#) on page 3-15
- [“Checking the Oracle Registry Repository Exchange Utility Log File”](#) on page 3-17
- [“Known Issues”](#) on page 3-17

Running the Oracle Registry Repository Exchange Utility

The Oracle Registry Repository Exchange Utility is run from a command-prompt, using the following syntax:

```
> alrrx.bat <options>
```

[Table 3-1](#) defines the configuration options available when running the Oracle Registry Repository Exchange Utility.

Table 3-1 Oracle Registry Repository Exchange Utility Command-line Options

Option	Required	What it does...
<code>-mode <mode></code>	Yes	Configures the utility to run in either <code>publish</code> or <code>receive</code> mode. <ul style="list-style-type: none"> To publish Services to Oracle Service Registry from Oracle Enterprise Repository, use: <code>alrrx.bat -mode publish</code> To receive Services from Oracle Service Registry into Oracle Enterprise Repository, use: <code>alrrx.bat -mode receive</code>
<code>-config <File Name></code>	Optional	Passes the <code>alrrx.xml</code> file as a parameter. Example usage: <code>alrrx.bat -mode publish -config C:\rm\uddi\alrrx.xml</code> If this parameter is omitted, the configuration XML file will be loaded from the current directory where the <code>alrrx.bat</code> file is located using the system's Classpath.
<code>-map <Dir Name></code>	Optional	Generates a <code>UDDIMappings.xml</code> file by connecting to Oracle Enterprise Repository and populating it with the t-models based on the configured Categorizations. Also, this loads the Oracle Registry Repository Exchange Utility configuration using the <code>-config</code> parameter from the default location. You can customize this file if the t-model already exists in Oracle Service Registry and map an Oracle Enterprise Repository categorization to a t-model. Example usage: <code>alrrx.bat -map c:/rm/uddi</code> For more information about the <code>UDDIMappings.xml</code> file, see “Configuring Oracle Enterprise Repository Categorizations in the UDDI Mappings File” on page 2-8.

Table 3-1 Oracle Registry Repository Exchange Utility Command-line Options (Continued)

Option	Required	What it does...
<code>-publish_tmodel</code> <File Name>	Optional	<p>Publishes all the t-models configured in the <code>UDDIMappings.xml</code> file to Oracle Service Registry. If a t-model already exists in Oracle Service Registry, it will be skipped. You need to manually delete the existing t-models if you want the Oracle Registry Repository Exchange Utility to populate those t-models. Also, this loads the Oracle Registry Repository Exchange Utility configuration using the <code>-config</code> parameter from the default location.</p> <p>Example usage: <code>alrrx.bat -publish_tmodel c:/rm/UDDIMappings.xml</code></p> <p>For more information about the <code>UDDIMappings.xml</code> file, see “Configuring Oracle Enterprise Repository Categorizations in the UDDI Mappings File” on page 2-8.</p>
<code>-link <asset_id></code> <service_key>	Optional	<p>Links a Service in Oracle Enterprise Repository to a Service in Oracle Service Registry together so that the Oracle Registry Repository Exchange Utility can treat them as the same service. This may be required when the Service in Oracle Enterprise Repository and a service in Oracle Service Registry are the same but were published to Oracle Enterprise Repository and Oracle Service Registry using different tools. For example, AquaLogic SOA Management could have published a service to Oracle Service Registry and the SAM plug-in could have published the same service to Oracle Enterprise Repository. After they are linked, they can be synchronized by the utility.</p> <p>Example usage: <code>alrrx.bat -link 50822 uddi:systinet.com:demo:hr:employeesList</code></p> <p>Note: Use this option with caution when linking two services.</p>

Invoking the Oracle Registry Repository Exchange Utility Using Workflows

By using workflows to invoke the Oracle Registry Repository Exchange Utility, you can automate the synchronization of Oracle Enterprise Repository and Oracle Service Registry.

For more information about the `workflow.xml` file and configuring workflows, refer to the [Oracle Enterprise Repository Configuring and Managing Advanced Registration Flows](#) manual.

Synchronizing Using Timer Based Workflows

A timer based workflow can be used to synchronize Oracle Enterprise Repository from Oracle Service Registry, or to synchronize Oracle Service Registry from Oracle Enterprise Repository.

Table 3-2 shows the names and descriptions of the timer based workflows:

Table 3-2 Timer Based Workflows

Workflow Name	Description
autoSyncAlerToUddi	Moves services from Oracle Enterprise Repository to Oracle Service Registry.
autoSyncUddiToAler	Moves services from Oracle Service Registry to Oracle Enterprise Repository.

The timer can be configured to wake up based on the timerInterval settings in the workflow.xml file. The following example shows the autoSyncAlerToUddi and autoSyncUddiToAler workflows configured in the workflow.xml file:

```
<automation>
  <autoSyncAlerToUddi configFileName="AlerToUddiSyncAlrrxConfig.xml"
    mappingFileName="AlerToUddiSyncAlrrxMapping.xml" timerInterval="2d" />
  <autoSyncUddiToAler configFileName="UddiToAlerSyncAlrrxConfig.xml"
    mappingFileName="UddiToAlerSyncAlrrxMapping.xml" timerInterval="3d" />
</automation>
```

Synchronizing When Events are Triggered

Individual services and their metadata can be moved to Oracle Service Registry by wiring the events that get triggered when these services are registered or a lifecycle of a service is changed.

The PublishAssetToUDDI workflow can be used to move the services. The PublishAssetToUDDI workflow can be wired to any event trigger, depending on the requirements.

For example, the following configuration moves the service when the Asset Lifecycle stage is QA:

```
<state
name="urn:com:bea:aler:events:type:CategorizationChanged:AssetLifecycleStage"
value="QA"
  <action>PublishAssetToUddi</action>
```



```
<alrrxConfigFileName>alrrx4LCC_QA.xml</alrrxConfigFileName>  
<alrrxMappingFileName>uddiMapping4LCC_QA.xml</alrrxMappingFileName>  
</state>
```

Also, note that the configuration related to where the registry is running is configured using the `alrrxConfigFileName` setting, and that the mapping file related to the categorization is configured using the `alrrxMappingFileName` setting.

You could have different configuration files configured to point to different registries. Depending on the lifecycle, the services can be moved to different registries based on the Asset Lifecycle triggers.

Example Use Cases

Oracle Enterprise Repository supports the following use cases for automated workflows:

- Automated workflows can automate synchronizing Oracle Enterprise Repository and Oracle Service Registry based on a configured timer interval.
- When different registries are running to categorize services based on the Lifecycle, services can be published automatically to these registries by wiring the Asset Lifecycle event trigger to publish to the relevant registry.
- Services can be automatically published to Oracle Service Registry if a service is submitted to Oracle Enterprise Repository. When a service is submitted to Oracle Enterprise Repository, it triggers an event which will be processed by the workflows. The workflows will move this to Oracle Service Registry automatically, if it is configured to do so.

How the Exchanged Metadata Is Synchronized

This section describes how metadata is synchronized when assets are exchanged between Oracle Enterprise Repository and Oracle Service Registry.

Synchronizing the Metadata Published from Oracle Enterprise Repository to Oracle Service Registry

This section describes how metadata is synchronized when publishing assets from Oracle Enterprise Repository to Oracle Service Registry.

Note: When synchronizing a service to Oracle Service Registry that was previously synchronized, Oracle Service Registry does not show the updated values if an Oracle Service Registry browser instance is already open. Therefore, all the Oracle Service

Registry browser instances need to be restarted to see the updated values. For more information, see [“Known Issues” on page 3-17](#).

Business Entities

Check if the Service asset being published has a related (by configured relation) Business Entity asset, as follows:

- If yes, the existing asset to newly created Service asset using configured relation.
- If no, get default Business Entity name from configuration, as follows:
 - Check if default business entity asset is configured in the Oracle Enterprise Repository System Settings.
 - If yes, use that as the default Business Entity.
 - If no, use the default Business Entity name from the `alrrx.properties` file.

Endpoint

Check if the Service asset being published has one or more related endpoint assets, as follows:

- If yes, create UDDI Binding Templates if this is the first time. If this was synchronized before, update the existing Binding Templates. Use the Endpoint URI in the Endpoint asset to arrive at the UDDI Access point. Derive the port from the WSDL that is attached to the `File Info` of the endpoint asset.
- If no, create the Binding Templates based on the WSDL that is attached to the `File Info` of the Service asset if the WSDL contains the port info.
- If Asset Life cycle is attached to the endpoint and if the Endpoint Asset Life cycle Query is used, filter the endpoints based on the Asset Lifecycle. For example, you can publish the staging endpoint to the Staging Registry and publish the production endpoint to the production registry.

Categorizations

Check if Categorizations are applied to Service and Endpoint assets, as follows:

- If yes, load the Categorization mapping for each of the applied Categorizations from the `UDDIMappings.xml` file.
 - If the mapping is found, add an entry to the Category Bag, either to the Business Service or Binding Template.

- If the t-model is not found in the Oracle Service Registry, automatically create the t-models.
- If no, the Categorization will not be applied to the Service in Oracle Service Registry.

Registration and Active Status

The Registration and Active Status are added to Category Bags. [Figure 3-1](#) illustrates how references appear in Oracle Service Registry.

Figure 3-1 Oracle Service Registry t-model Categories for a WSDL Service

The screenshot displays the 'Categories' section for a WSDL service. It features a table with three columns: tModel, key name, and key value. Below the table is a section for 'Category groups' which is currently empty.

tModel	key name	key value
uddi-org:wSDL:types	uddi.org:wSDL:types	service
uddi-org:xml:localName	uddi.org:xml:localName	GoogleSearchService
uddi-org:xml:namespace	uddi.org:xml:namespace	urn:GoogleSearch
bea-com:aler:uuid	bea-com:aler:uuid	940e2081-7fb3-11dd-9a50-ffbe5fcb0906
bea-com:aler:ActiveStatus		Active
bea-com:aler:RegistrationStatus		Unsubmitted

Category groups
No category groups found.

[Figure 3-2](#) illustrates how two endpoints that are linked to a service in Oracle Enterprise Repository appear in Oracle Service Registry.

Figure 3-2 WSDL Bindings in Oracle Service Registry

The screenshot displays the 'Bindings' section for a WSDL service. It features a table with three columns: type, detail, and description. There is one binding entry listed.

type	detail	description
access point	view http://api.google.com/search/beta2	

Figure 3-3 and Figure 3-4 illustrate how the different entities and their relationships appear in the Oracle Enterprise Repository Navigator.

Figure 3-3 Entity Relationship in Oracle Enterprise Repository Navigator

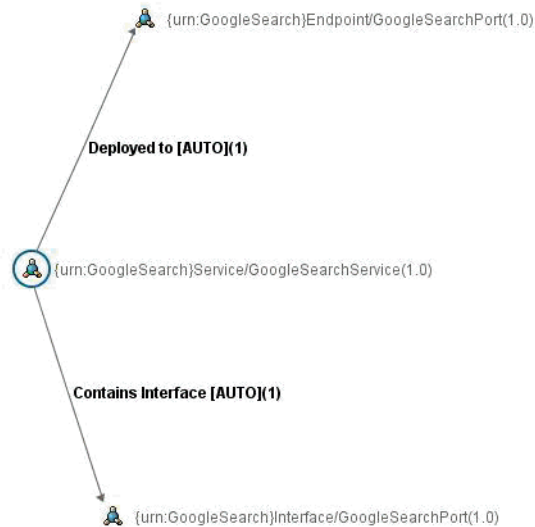


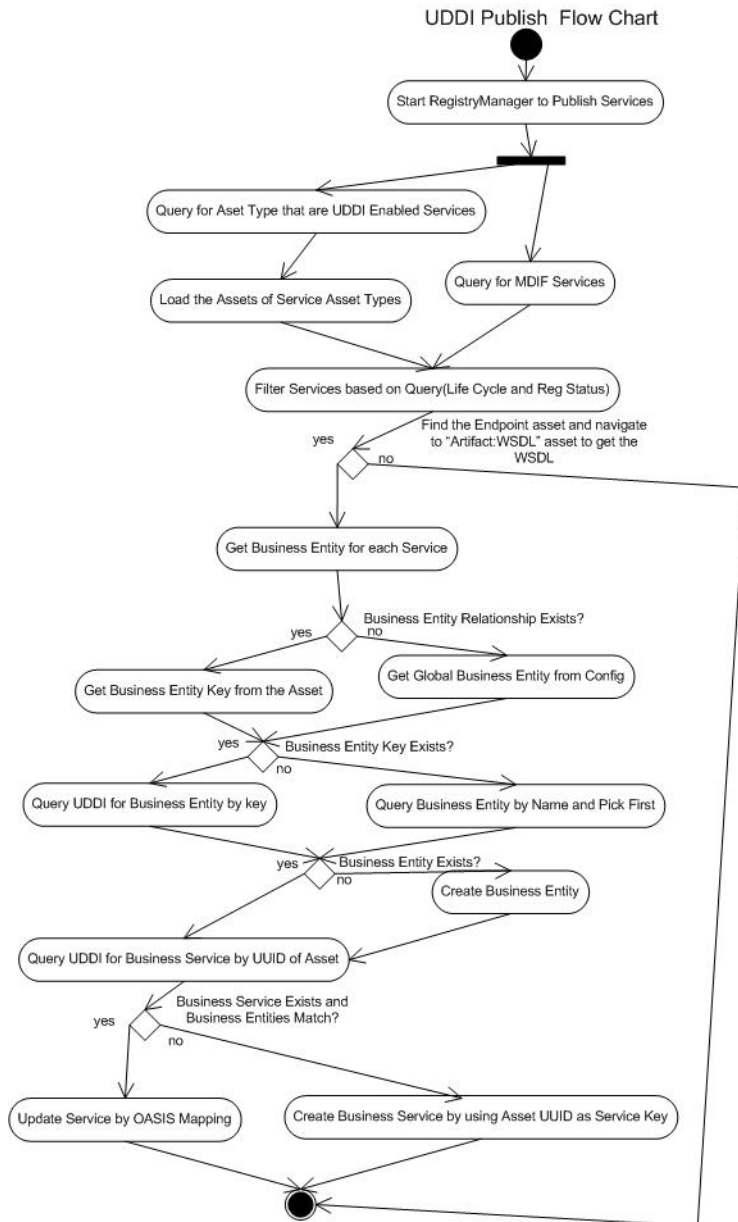
Figure 3-4 Another Entity Relationship in Oracle Enterprise Repository Navigator



Sample Flow of Metadata from Oracle Enterprise Repository to Oracle Service Registry

Figure 3-5 illustrates the Oracle Enterprise Repository > Oracle Service Registry metadata synchronization described in this section.

Figure 3-5 Flow of Metadata Published from Oracle Enterprise Repository to Oracle Service Registry



Synchronizing the Metadata from Oracle Service Registry to Oracle Enterprise Repository

This section describes how metadata is synchronized when receiving assets into the repository from Oracle Service Registry.

Business Entities

Check if the Service asset being received exists and is related to a Business Entity asset, as follows:

- If yes, the same Business Entity relationship will be used.
- If no, the Business Entity on the Oracle Service Registry side will be used. If the Business Entity does not exist in Oracle Enterprise Repository, it will be created.

If the Service asset is newly created, get the default Business Entity asset type for UDDI Business from the Oracle Enterprise Repository configuration:

- If found in the System Settings
- If not, from the `alrrx.properties` file

Check if an asset exists with that name and type, as follows:

- If yes, simply relate that existing asset to newly created service asset using configured relation.
- If no, create a new asset and relate it to newly created service asset using configured relation.

Endpoint

Check if the Service asset being received has one or more Endpoint assets related, as follows:

- If yes, create check if the Endpoint assets are the same as the existing Binding Templates using the UUID. If they are same, update the Endpoints.
- If no, create the Endpoints for each Binding Template and relate them to the Service asset.

Categorizations

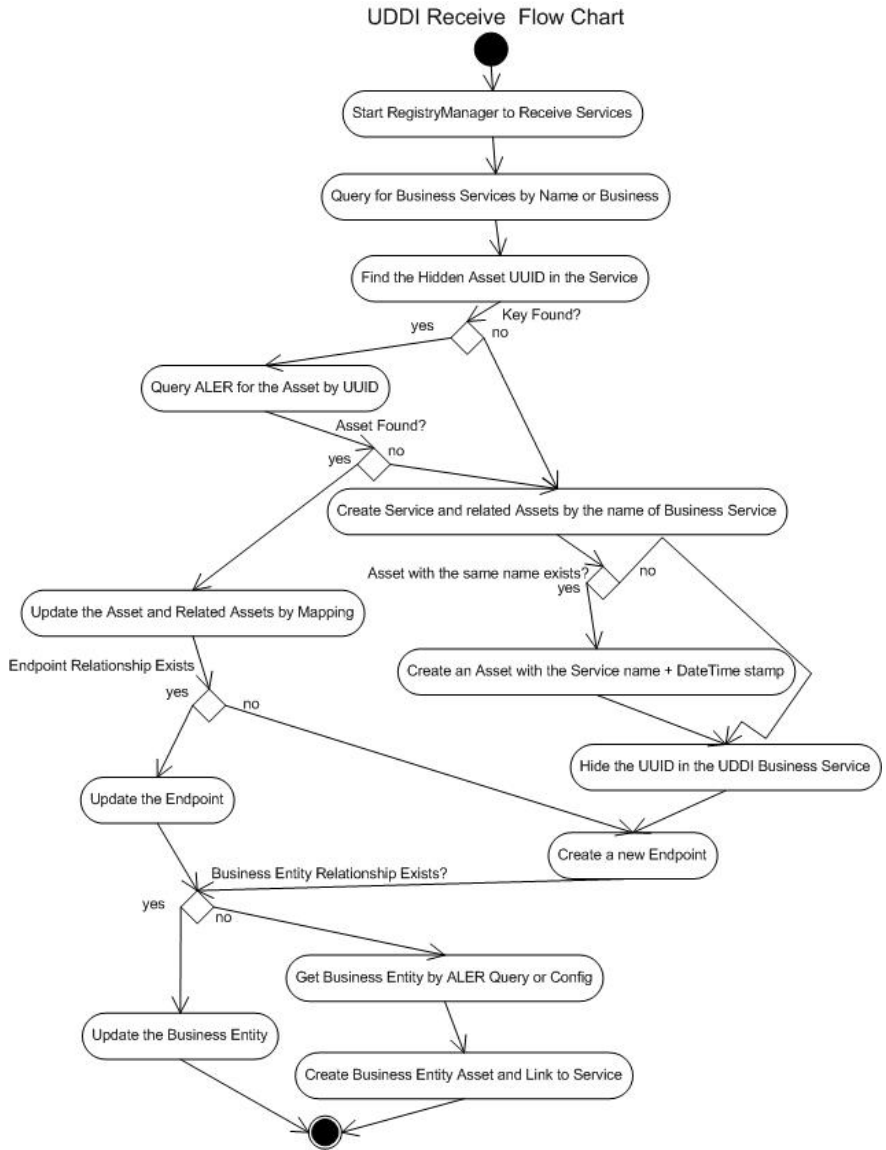
Check if Categorizations are present in the Category Bag, as follows:

- If yes, load the Categorization mapping for each of the applied Categorizations from the `UDDIMappings.xml` file.
- If the mapping is found, set the Categorizations of the Service asset.

Sample Flow of Metadata from Oracle Service Registry to Oracle Enterprise Repository

[Figure 3-6](#) illustrates the Oracle Service Registry > Oracle Enterprise Repository metadata synchronization described in this section.

Figure 3-6 Flow of Metadata Received from Oracle Service Registry



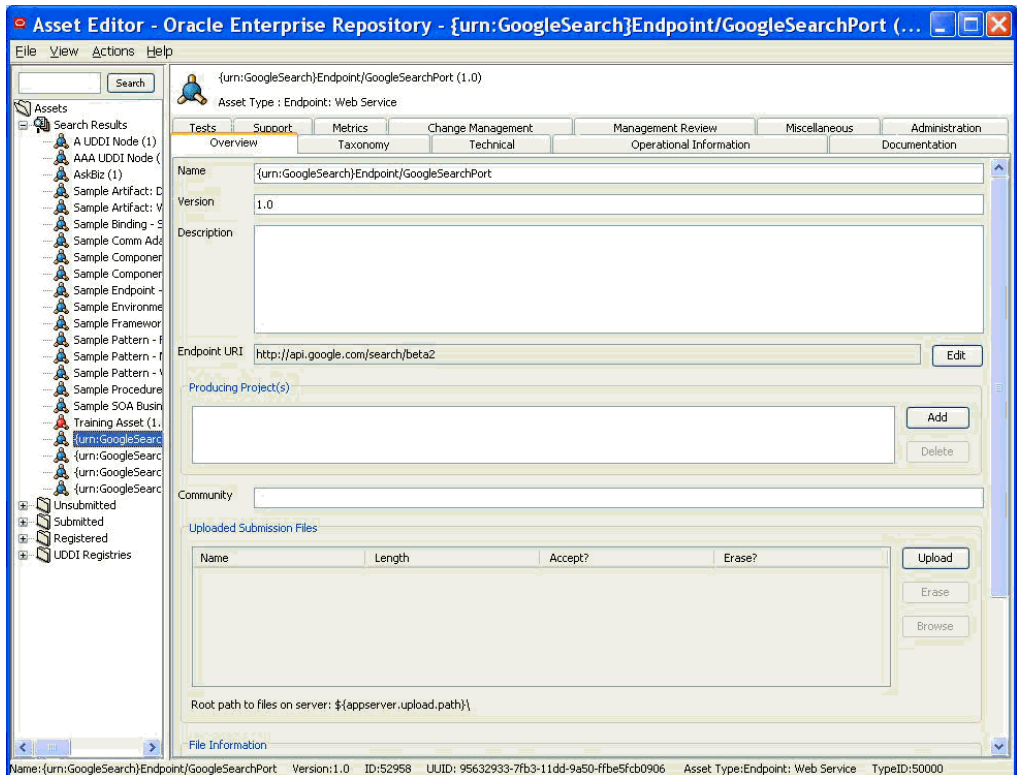
Web Service Endpoint Management

Use the Oracle Enterprise Repository Asset Editor to add the new `Endpoint:Web Service` asset type so that the endpoint information can be published to Oracle Service Registry.

1. From the File menu, select **New** to create a new `Endpoint:Web Service` asset type.
2. Add a description if necessary.
3. Select the **Taxonomy** tab to relate the asset to the Service asset by:
 - a. Scroll down to the Relationships section and click **Add**.
 - b. On the Add Relationship dialog box, select the **Provides Access to** Relationship Type.
 - c. Click **OK**.
 - d. Click **Approve** to save the change.
4. Select the **Overview** tab (shown in [Figure 3-7](#)), and do the following:
 - a. In the Endpoint URI field, click **Add** to enter the **URI** for the asset. Then click **OK** to add the Endpoint URI.
 - b. In the File Information field, click **Add** to enter the effective **WSDL** that contains the port. Then click **OK** to add the WSDL.

Note: You can omit this step if the WSDL attached to the Service contains the port information.
 - c. Click **Approve** to save the change.

Figure 3-7 Configure EndPoint:Web Service Asset Type



5. Optionally, you can set the Asset Lifecycle to this asset if the Endpoint needs to be filtered by the Asset Lifecycle.

For more information on using the Asset Editor to manage assets, see the [Oracle Enterprise Repository Registrar Guide](#).

Searching for Oracle Service Registry Exchanged Metadata in Oracle Enterprise Repository

The Oracle Registry Repository Exchange Utility tags each published and received Service with information that can be used for querying, as follows:

- Publish or Receive Date and Time
- Source and Destination Registries.

To query published/received Service information, use the Oracle Enterprise Repository More Search Options feature, as follows:

1. In the Oracle Enterprise Repository Web console, open the **Assets** page.
2. In the Search box in the sidebar, click the **More Search Options** link.
The More Search Options dialog box opens.
3. Select the **Filter by Additional Criteria** check box to reveal the additional filtering criteria options.
4. On the Select a Field drop-down, select the **internal.alrr.exchange.store** option.
5. In the field next to the **Add** button, enter the date the metadata was exchanged, as shown in [Figure 3-8](#).

Figure 3-8 More Search Options Dialog Box

More Search Options

Enter Search String: Search

Type: All Types Registration Status: Registered

Filter by Categorizations

Filter by Additional Criteria

Select a Field: internal.alrr.exchange.store

Enter an XPath: (optional) ttings/registries/registry/instanceDetails/publishDateTime

Contains

10-27-2007 Add

Rerr

6. Click the **Search** button at the bottom of the dialog box.

For more information on using the Oracle Enterprise Repository search options, see the [Oracle Enterprise Repository User Guide](#).

Checking the Oracle Registry Repository Exchange Utility Log File

The Oracle Registry Repository Exchange Utility uses `log4j` for logging the detailed tasks performed. The log file is stored in the `<ExchangeUtility Tool Home>` directory. The logging options can be changed by updating the file `log4fl.properties` file located in the `<ExchangeUtility Tool Home>` directory.

Known Issues

This section describes the known issues when using the Oracle Registry Repository Exchange Utility.

Resynchronizing Oracle Service Registry Services

When synchronizing a Service to Oracle Service Registry that was previously synchronized, there is known issue where Oracle Service Registry does not show the updated values if an Oracle Service Registry browser instance is already open. Therefore, all the Oracle Service Registry browser instances need to be closed to see the updated values.

Import of Oracle Service Bus WSDLs

Oracle Enterprise Repository WSDL import is not currently capable of supporting the import of an XSD into a WSDL document using the WSDL import mechanism. This is considered improper use of the WSDL import element by the industry. An example of the improper usage of the WSDL import element to import an XSD, along with an example of the correct way to import XSD into a WSDL is included. Note that Oracle Enterprise Repository WSDL import does support the importing of WSDL into a WSDL document using the WSDL import element.

Oracle Service Bus currently generates WSDLs that incorrectly import XSD using the WSDL import element. This causes a problem in the AL suite due to the fact that these Oracle Service Bus WSDLs can be parsed and submitted to Oracle Enterprise Repository properly by the common Eclipse tooling; however, the Oracle Registry Repository Exchange Utility is not capable of parsing the WSDLs when migrating them back to Oracle Service Registry.

Workaround of Solution

Example of improper usage of the WSDL import element to import XSD:

```
<?xml version='1.0' encoding='UTF-8'?>
<definitions name='OrderProcessing'
targetNamespace='http://avitek.com/orderprocessing/definitions'
xmlns:tns='http://avitek.com/orderprocessing/definitions'
xmlns:po='urn:iwaysoftware:ibse:jul2003:createPO'
xmlns:por='urn:iwaysoftware:ibse:jul2003:createPO:response'
xmlns:pos='urn:iwaysoftware:ibse:jul2003:POStatus'
xmlns:soap='http://schemas.xmlsoap.org/wsdl/soap/'
xmlns='http://schemas.xmlsoap.org/wsdl/'>
    <import namespace='urn:iwaysoftware:ibse:jul2003:createPO'
location='bapipo.xsd' />
    <import namespace='urn:iwaysoftware:ibse:jul2003:createPO:response'
location='bapipor.xsd' />
    <import namespace='urn:iwaysoftware:ibse:jul2003:POStatus'
location='POStatus.xsd' /> .
```

Example of proper usage of the XSD import element to import XSD:

```
<?xml version='1.0' ?>
<wsdl:definitions targetNamespace='urn:listing2'
xmlns:tns='urn:listing2'
xmlns:listing3='urn:listing3'
xmlns:wsdl='http://schemas.xmlsoap.org/wsdl/'
xmlns:xsd='http://www.w3.org/2001/XMLSchema'
xmlns:soap='http://schemas.xmlsoap.org/wsdl/soap/'>
  <wsdl:types>
    <xsd:schema targetNamespace='urn:listing2'
xmlns:listing3='urn:listing3'
xmlns:xsd='http://www.w3.org/2001/XMLSchema'>
      <xsd:import namespace='urn:listing3'
schemaLocation='listing3.xsd' />
```