

## **Oracle® Fusion Middleware**

Administrator's Guide for Oracle Enterprise Repository

12c Release 1 (12.1.3)

**E55954-03**

October 2015

Documentation for administrators that describes Oracle Enterprise Repository configuration and customization tasks.

Copyright © 2008, 2015, Oracle and/or its affiliates. All rights reserved.

Primary Author: Oracle Corporation

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, then 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 about 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 unless otherwise set forth in an applicable agreement between you and Oracle. 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, except as set forth in an applicable agreement between you and Oracle.

---

---

# Contents

<b>Preface</b> .....	xvii
Audience .....	xvii
Documentation Accessibility .....	xvii
Related Documents .....	xvii
Conventions .....	xvii
<b>What's New in This Guide</b> .....	xix
New and Changed Features for 12c Release 1 (12.1.3).....	xix
Other Significant Changes in this Document for 12c Release 1 (12.1.3) .....	xix
<b>1 Basic Configuration</b>	
1.1 Overview .....	1-1
1.2 Departments .....	1-2
1.2.1 Creating a New Department .....	1-2
1.2.2 Editing Department Information.....	1-2
1.2.3 Deleting a Department.....	1-3
1.3 Roles .....	1-3
1.3.1 User Role Descriptions.....	1-3
1.3.2 Creating a New Role .....	1-5
1.3.3 Editing a Role .....	1-6
1.3.4 Deleting a Role .....	1-7
1.4 Access Settings .....	1-8
1.4.1 Granting Permissions to a Role .....	1-8
1.4.2 Editing Permissions.....	1-10
1.4.3 Verifying User Permissions.....	1-10
1.4.4 Security Considerations .....	1-11
1.4.4.1 Access Setting and the Asset Editor.....	1-12
1.4.4.2 Permissions.....	1-12
1.5 Users .....	1-14
1.5.1 Creating a New User .....	1-14
1.5.2 Viewing User Information.....	1-15
1.5.3 User Search .....	1-15
1.5.4 Cloning a User.....	1-16
1.5.5 Editing User Information.....	1-16
1.5.6 Deleting a User.....	1-16

1.6	Projects.....	1-17
1.6.1	Viewing Project Details.....	1-17
1.6.2	Creating a Project.....	1-19
1.6.3	Editing a Project.....	1-22
1.6.4	Reassigning Users/Usage.....	1-22
1.6.5	Locating a Project.....	1-23
1.6.6	Closing a Project.....	1-23
1.6.7	Adding a Relationship to a Project.....	1-24
1.6.8	Deleting a Project.....	1-27
1.7	Sessions.....	1-27
1.7.1	Viewing Sessions.....	1-28
1.7.2	Deleting a Single Session.....	1-28
1.7.3	Deleting Multiple Sessions.....	1-28
1.8	Types.....	1-28
1.9	Accessing Oracle Enterprise Repository Diagnostics Page.....	1-29
1.10	Install Java Web Start.....	1-29
1.10.1	Download and Install Java Web Start.....	1-29
1.10.2	Troubleshooting Java Web Start.....	1-29
1.10.3	Configure Java Web Start on Linux.....	1-29

## 2 System Settings Overview

2.1	Access System Settings.....	2-1
2.2	Search for System Settings.....	2-2
2.3	Enable System Settings Properties.....	2-3
2.4	Move Settings to Database.....	2-3
2.5	System Settings in Oracle Enterprise Repository.....	2-3
2.5.1	Functional Settings.....	2-3
2.5.1.1	General User Interface.....	2-4
2.5.1.2	Search.....	2-9
2.5.1.3	Print Asset Detail via PDF.....	2-12
2.5.1.4	Projects and Navigator.....	2-12
2.5.1.5	Assets in Progress.....	2-15
2.5.1.6	Policy Management.....	2-15
2.5.1.7	Notifications.....	2-15
2.5.1.8	Access Control.....	2-17
2.5.1.9	Software File Identification (SFID).....	2-17
2.5.1.10	Use - Download.....	2-17
2.5.1.11	Custom Links.....	2-18
2.5.1.12	Reports.....	2-20
2.5.1.13	Miscellaneous.....	2-20
2.5.2	Server Settings.....	2-22
2.5.2.1	General.....	2-23
2.5.2.2	Server Resources.....	2-24
2.5.2.3	Email.....	2-25
2.5.2.4	Upload Area.....	2-26
2.5.2.5	Audit.....	2-27
2.5.2.6	Caching.....	2-27



2.5.3	Enterprise Authentication Settings .....	2-27
2.5.3.1	General .....	2-28
2.5.3.2	Authentication Token .....	2-29
2.5.3.3	Password Settings.....	2-30
2.5.3.4	Plugin Login Settings.....	2-31
2.5.3.5	LDAP and Active Directory Settings.....	2-31
2.5.3.6	Single Sign-On (SSO) Settings .....	2-34
2.5.3.7	Container Login Settings .....	2-36
2.5.4	Open API Settings.....	2-36
2.5.4.1	Open API .....	2-36
2.5.5	External Integration Settings.....	2-37
2.5.5.1	UDDI .....	2-38
2.5.5.2	File Stores.....	2-38
2.5.5.3	Rational ClearCase .....	2-38
2.5.5.4	Eventing .....	2-38
2.5.6	Import / Export and Introspection Settings .....	2-39
2.5.6.1	Import / Export .....	2-39
2.5.6.2	WSDL and XSD.....	2-40
2.6	Known Issues.....	2-40

### 3 Customizing Oracle Enterprise Repository

3.1	Overview .....	3-1
3.1.1	Accessing the Oracle Enterprise Repository Home Page .....	3-1
3.1.2	Customizing the Oracle Enterprise Repository Home Page .....	3-2
3.2	Rebranding Oracle Enterprise Repository .....	3-2
3.2.1	Start with CSS.....	3-2
3.2.2	Customizing Skins .....	3-3
3.2.2.1	Choosing a Skin .....	3-3
3.2.2.2	Creating a Skin .....	3-3
3.2.2.3	Images .....	3-3
3.2.2.4	Activating Your Custom Skin.....	3-4
3.2.2.5	Installing a Oracle-supplied Custom Skin in Oracle Enterprise Repository .....	3-4
3.2.2.6	Using the Directory Structure.....	3-4
3.3	Role/Department-based Homepage Display .....	3-5
3.3.1	Installation .....	3-5
3.4	Install Oracle Enterprise Repository Solution Packs .....	3-6

### 4 Configuring E-Mail Notifications and Distribution Lists

4.1	Overview .....	4-1
4.1.1	Using E-Mail Notifications and E-Mail Distribution Lists .....	4-1
4.2	Setting Up E-Mail Notification .....	4-1
4.3	Managing E-mail Distribution Lists .....	4-2
4.3.1	Creating a Distribution List.....	4-2
4.3.2	Updating a Distribution List.....	4-3
4.3.3	When List Members Are Copied on E-mails .....	4-4
4.4	Working with E-Mail Templates .....	4-4

4.4.1	Viewing E-Mail Templates .....	4-6
4.4.2	Editing E-Mail Templates .....	4-7

## 5 Import Export Tool

5.1	Overview .....	5-1
5.2	Using the Import/Export Tool.....	5-2
5.2.1	Overview .....	5-3
5.2.2	The parameter.properties File.....	5-3
5.2.2.1	Supported Parameters in the parameter.properties File .....	5-3
5.2.2.2	Sample parameter.properties File .....	5-5
5.2.3	The Default Import Behavior .....	5-6
5.3	Starting the Import/Export Tool .....	5-7
5.3.1	Launching from Oracle Enterprise Repository .....	5-7
5.3.2	Launching from Outside of Oracle Enterprise Repository .....	5-8
5.3.3	Initial Startup.....	5-8
5.4	Export Items from Oracle Enterprise Repository .....	5-9
5.5	Importing Items into Oracle Enterprise Repository .....	5-12

## 6 Configuring Reporting with BI Publisher

6.1	Deploy Oracle Enterprise Repository Pre-configured BI Publisher Content.....	6-1
6.1.1	Configure BI Publisher with Oracle Enterprise Repository .....	6-2
6.2	Understanding the Security Model .....	6-2
6.2.1	The Administrator User .....	6-2
6.2.2	The OER User and OER_REPORTS Role .....	6-2
6.2.3	Set Local Time Zone for Report Footer.....	6-2
6.3	Configuring the Oracle Enterprise Repository JDBC Data Source.....	6-3
6.4	Configuring the BI Publisher Server - Cache Expiration .....	6-3
6.5	Configuring the Oracle Enterprise Repository System Settings to Enable Reporting.....	6-4
6.6	Working with Reports.....	6-5
6.6.1	Understanding Report Types.....	6-5
6.6.2	Adding Custom Reports.....	6-9
6.6.2.1	Overview .....	6-9
6.6.2.2	Configuration Steps.....	6-9
6.6.3	Running a Report.....	6-10

## 7 Using the Type Manager

7.1	Overview .....	7-1
7.2	Launching the Type Manager .....	7-2
7.2.1	Type Manager Views .....	7-2
7.2.1.1	The Editor .....	7-2
7.2.1.2	The Viewer.....	7-4
7.3	Creating a New Type.....	7-4
7.4	Configuring Tabs to Support Registration Workflow .....	7-5
7.4.1	Adding Tabs .....	7-5
7.4.2	Change the Position of a Tab .....	7-6
7.4.3	Editing Tabs.....	7-7

7.4.4	Deleting Tabs .....	7-8
7.5	Tab Elements .....	7-9
7.5.1	Working with Elements .....	7-14
7.6	Adding Images for Type Manager Icons .....	7-15
7.6.1	Icon Image Requirements .....	7-15
7.6.2	View/Select Available Type Icons .....	7-15
7.7	Unique ID .....	7-16
7.7.1	Adding the Unique ID Element to a Type .....	7-17
7.7.2	Information Necessary for Custom Reporting .....	7-18
7.8	Common Metadata Viewers .....	7-19
7.8.1	XPath Common Metadata Viewer .....	7-19
7.8.2	XSLT Common Metadata Viewer .....	7-20
7.9	Viewing and Editing Type XML Schemas .....	7-20
7.9.1	Configuring XML Schemas .....	7-20
7.9.2	Using the Export Schema Feature .....	7-20
7.9.3	Using the Type XML Editor .....	7-21

## 8 Configuring and Using Automated Harvesting

8.1	Getting Started with Harvester .....	8-1
8.1.1	Prerequisites .....	8-1
8.1.2	Harvester Functionality .....	8-2
8.1.3	Artifacts/Products Version Matrix .....	8-3
8.2	Configuring the Harvester .....	8-3
8.2.1	Obtaining the Harvester .....	8-4
8.2.2	Configuring the Harvester for the Command Line .....	8-6
8.2.2.1	Setting Repository Connection Information for the Command Line .....	8-6
8.2.2.2	Selecting the Artifacts to Harvest for the Command Line .....	8-6
8.2.3	Invoking the Harvester Using the Repository.Submit Ant Task .....	8-10
8.2.3.1	Specifying Parameters for the repository.submit Ant Task .....	8-10
8.2.3.2	Specifying Parameters as Nested Elements .....	8-11
8.2.3.3	Running the Harvester from Ant .....	8-14
8.2.3.4	Using the Third-Party Tasks .....	8-15
8.2.4	Invoking Harvester from WLST .....	8-15
8.2.5	Runtime Harvesting Details .....	8-17
8.2.5.1	Harvesting from Oracle SOA Suite Server .....	8-17
8.2.5.2	Harvesting Web Services from WebLogic Server .....	8-19
8.2.5.3	Harvesting Web Services from Oracle Service Bus .....	8-20
8.2.5.4	Harvesting Deployed Services .....	8-20
8.2.6	Performing Optional Harvester Configuration .....	8-21
8.2.6.1	Oracle Metadata Service URLs .....	8-21
8.2.6.2	Classpath URLs .....	8-22
8.2.7	Configuring Logging for the Harvester .....	8-22
8.2.8	Transaction Handling in Harvester .....	8-22
8.2.9	Advanced Configuration .....	8-22
8.3	Harvesting of Artifacts .....	8-24
8.3.1	Harvesting Artifacts .....	8-24
8.3.1.1	BPEL .....	8-26

8.3.1.2	WSDL .....	8-27
8.3.1.3	WADL .....	8-27
8.3.1.4	XSD .....	8-28
8.3.1.5	XSL.....	8-28
8.3.1.6	Harvesting Oracle SOA Suite .....	8-28
8.3.1.7	JCA Adapters .....	8-33
8.3.1.8	Applications via Oracle BSE (WSDL/SOAP).....	8-34
8.3.1.9	Proxy Service or Business Service .....	8-35
8.3.1.10	Oracle Service Bus (OSB) Project.....	8-36
8.3.1.11	BPMN .....	8-37
8.3.2	Detecting Duplicate Artifacts.....	8-38
8.3.3	Downloading Harvested Artifacts .....	8-38
8.3.4	Searching Harvested Assets .....	8-38
8.3.4.1	Using Consumption Method .....	8-38
8.3.4.2	Using Metadata.....	8-39
8.3.5	Previewing of the Created Assets in Harvester .....	8-40
8.3.6	Best Practices .....	8-41
8.3.6.1	Recommended Privileges for Harvesting.....	8-41
8.3.6.2	Do Not Override the Namespace Parameter.....	8-41
8.3.6.3	Namespaces in WSDL Files.....	8-41
8.3.6.4	Harvesting Completed Work.....	8-42
8.3.6.5	Harvesting and Maintenance Releases of XSD .....	8-42
8.3.6.6	Harvesting Metadata throughout the Asset Lifecycle .....	8-42
8.3.6.7	Downloading WSDL Artifacts.....	8-43
8.3.6.8	Harvesting Retired Endpoints .....	8-43
8.3.6.9	Harvesting SOA Projects .....	8-43
8.3.6.10	Harvesting a File in the Harvester Folder.....	8-43
8.3.6.11	Running Harvester and Exchange Utility on the Same Computer .....	8-43
8.3.6.12	Harvesting Composite Service WSDL.....	8-44
8.3.6.13	Understanding Versioning of Assets.....	8-44
8.3.7	Known Issues .....	8-45
8.3.7.1	Asset Types Must be Present in the System .....	8-45
8.3.7.2	Two Versions of an Asset Type .....	8-45
8.3.7.3	Do Not Delete the Harvester-Specific Metadata Entries in a in a Harvested Asset ... 8-45	
8.3.7.4	Supported Remote Server Types.....	8-45
8.3.7.5	Using Incorrect Encrypted Password .....	8-46
8.3.7.6	View in Repository Option .....	8-47

## 9 Using the Asset Editor

9.1	Overview .....	9-1
9.2	Launching the Asset Editor .....	9-1
9.3	Configuring System Options.....	9-2
9.3.1	Configure Acceptable Value Lists .....	9-2
9.3.2	Configure Artifact Stores.....	9-3
9.3.3	Selecting a New Artifact Store .....	9-5
9.3.4	Configure Categorizations .....	9-5

9.3.5	Configure Relationships .....	9-7
9.3.5.1	Configure System-Supplied Relationships .....	9-8
9.3.6	Configure Rejection Reasons.....	9-8
9.3.7	Configure Vendors .....	9-9
9.4	Advanced Asset Submission.....	9-10
9.5	Asset/Compliance Template/Policy Migration .....	9-11
9.5.1	Migrating an Asset/Compliance Template/Policy.....	9-12
9.6	Understanding the Asset Registration Process.....	9-13
9.6.1	Accepting a Submitted Asset .....	9-14
9.6.2	Registering an Asset .....	9-15
9.6.2.1	Approving an Asset's Tabs .....	9-15
9.6.2.2	Completing the Registration Process.....	9-17
9.6.3	Adding a Note to an Asset .....	9-17
9.7	Understanding Asset Status .....	9-17
9.7.1	Assigning or Changing an Asset's Status.....	9-18
9.8	Deleting an Asset .....	9-18

## **10 Configuring Oracle Enterprise Repository to use External Authentication Tooling**

10.1	Overview .....	10-1
10.1.1	LDAP/Active Directory .....	10-1
10.1.2	Enable LDAP Integration System Properties.....	10-2
10.1.3	Modify LDAP/Active Directory Properties .....	10-2
10.1.4	Security Considerations .....	10-5
10.1.4.1	Use Case Sample Scenarios .....	10-5
10.1.5	LDAP Property Examples .....	10-7
10.2	Single Sign-On .....	10-7
10.2.1	Overview.....	10-8
10.2.2	Configure Oracle Enterprise Repository For Use With Single Sign-On Authentication ..	10-8
10.2.2.1	Enable Single Sign-On Integration System Properties.....	10-8
10.2.2.2	Using the Oracle Enterprise Repository SSO Integration with Basic Authentication	10-12
10.2.2.3	Modify Application Property Files Manually .....	10-12
10.2.3	Advanced Options.....	10-13
10.2.3.1	Creating/Assigning Default Roles for New Users.....	10-14
10.2.3.2	Create New Users/Allow Unapproved Users.....	10-15
10.2.3.3	Enable Unapproved/New User Login.....	10-15
10.2.3.4	New User Notification.....	10-15
10.2.3.5	Syncing Departments.....	10-15
10.2.3.6	Syncing Roles .....	10-16
10.2.3.7	Enable Debug Logging .....	10-17
10.3	Container Managed Setup .....	10-17
10.3.1	Overview .....	10-18
10.3.2	Configure Oracle Enterprise Repository for Container Managed Authentication	10-18
10.3.3	Modify the Web Application's Web.xml File to Allow for Container Authentication.....	10-19

## 11 Configuring Advanced Role-based Access Control

11.1	Overview .....	11-1
11.2	Security Considerations .....	11-2
11.2.1	Access Settings and the Asset Editor .....	11-2
11.2.2	Enabling Custom Access Settings .....	11-2
11.3	Basic Concepts .....	11-3
11.3.1	Roles.....	11-3
11.3.2	Access Settings .....	11-4
11.4	Access Definitions.....	11-4
11.5	Process Overview.....	11-6
11.6	Granting and Denying Permissions .....	11-7
11.6.1	Grant/Deny Access to Specific Assets.....	11-8
11.6.1.1	Prerequisites .....	11-8
11.6.1.2	Create the Asset CAS .....	11-9
11.6.2	Grant/Deny Access to Specific Download Files Within an Asset.....	11-13
11.6.2.1	Prerequisites .....	11-13
11.6.2.2	Create the File CAS .....	11-13
11.7	Configuring Access Settings for Existing Roles.....	11-14
11.7.1	User Roles and Default Privileges.....	11-14
11.7.2	Access Options .....	11-16
11.7.2.1	Option I: Grant or deny access to specific assets using CAS .....	11-17
11.7.2.2	Option II: Grant or deny access to specific files and assets using CAS.....	11-21
11.8	Role-based Access Control Use Cases.....	11-25
11.8.1	Use Case 1: Expose Web Services to Customers and Trading Partners .....	11-26
11.8.1.1	Benefit.....	11-26
11.8.1.2	Overview .....	11-26
11.8.1.3	Solution .....	11-27
11.8.1.3.1	Prerequisite.....	11-27
11.8.1.3.2	Create and Assign the Roles .....	11-27
11.8.1.3.3	Organize the Metadata .....	11-28
11.8.1.3.4	Create the Relationships.....	11-29
11.8.1.3.5	Apply the Relationship.....	11-29
11.8.1.3.6	Results .....	11-31
11.8.1.3.7	Validation Test.....	11-31
11.8.2	Use Case 2: Manage Intellectual Property in an Outsourced Environment .....	11-32
11.8.2.1	Benefits .....	11-32
11.8.2.2	Overview .....	11-33
11.8.2.3	Solution .....	11-33
11.8.2.3.1	Solution for Outsourced Development Teams .....	11-33
11.8.2.3.2	Solution for Export Controls.....	11-36
11.8.3	Use Case 3: Establish a Federated Repository .....	11-38
11.8.3.1	Benefit.....	11-38
11.8.3.2	Overview .....	11-38
11.8.3.3	Solution .....	11-38
11.8.3.3.1	Prerequisite.....	11-38
11.8.3.3.2	Create the Role.....	11-38
11.8.3.3.3	Create the Custom Access Settings.....	11-39

11.8.3.3.4	Validation Test.....	11-42
11.8.4	Use Case 4: Manage the Asset Lifecycle.....	11-43
11.8.4.1	Benefit.....	11-43
11.8.4.2	Overview .....	11-44
11.8.4.2.1	Asset Lifecycle Stages .....	11-44
11.8.4.2.2	General Configuration.....	11-45
11.8.4.2.3	Create the Roles .....	11-45
11.8.4.2.4	Create the Custom Access Settings.....	11-45
11.8.4.3	Configuring the Use Case Solutions .....	11-50
11.8.4.3.1	Requirements Gathering Stage Solution.....	11-50
11.8.4.3.2	Design and Development Stage Solution .....	11-52
11.8.4.3.3	Beta Release Stage Solution.....	11-52
11.8.4.3.4	Release Stage Solution .....	11-53
11.8.4.3.5	Scheduled for Retirement Stage Solution .....	11-53
11.8.4.3.6	Retired Stage Solution .....	11-53
11.8.4.4	Asset Lifecycle Management Solution Validation Tests.....	11-53
11.8.5	Use Case 5: Limit Access to Source Code Files to Asset Production Teams .....	11-55
11.8.5.1	Benefit.....	11-55
11.8.5.2	Overview .....	11-55
11.8.5.3	Solution .....	11-55
11.8.5.3.1	Create the Roles .....	11-55
11.8.5.3.2	Create the Customer Access Settings .....	11-55
11.8.5.3.3	Results .....	11-58
11.8.5.3.4	Validation Test.....	11-59
11.8.6	Use Case 6: Grant Browse-only Repository Access to Specific Groups.....	11-60
11.8.6.1	Benefit.....	11-60
11.8.6.2	Overview .....	11-60
11.8.6.3	Solution .....	11-60
11.8.6.4	Validation Test .....	11-61

## 12 Password Encryption

12.1	Generating Encrypted Passwords .....	12-1
12.1.1	Encrypted Passwords Used in Oracle Enterprise Repository Deployment.....	12-2
12.1.2	Other Passwords .....	12-2
12.2	Encrypting the Configuration File Passwords.....	12-2
12.2.1	Harvester Configuration File .....	12-3
12.2.2	Exchange Utility Configuration File .....	12-3
12.2.3	Workflow Configuration File.....	12-4
12.2.4	Export to API Catalog Configuration File.....	12-4

## 13 Configuring Oracle Enterprise Repository Express Workflows

13.1	Introduction to Oracle Enterprise Repository Express Workflow.....	13-1
13.2	Configuring Express Workflows .....	13-2
13.2.1	Step 1:..... Install the OER Express Workflows SOA Composite Application	13-3
13.2.2	Step 2: Configure OER Event Manager to deliver events to the deployed SCA .....	13-4
13.2.2.1	Configuring Web Service Endpoints .....	13-4

13.2.2.2	Enabling the Event Manager.....	13-5
13.2.3	Step 3: Define and Configure your Express Workflows .....	13-6
13.3	Running the Sample Workflow .....	13-7
13.4	Modifying the Sample Workflow .....	13-8
13.4.1	Modifying the Express Workflows Configuration File.....	13-8
13.4.1.1	Producing Projects.....	13-8
13.4.1.2	Approval Flows .....	13-9
13.4.1.3	Event-to-Action Mappings.....	13-10

## 14 Configuring Prescriptive Reuse

14.1	Overview .....	14-1
14.2	Creating a Compliance Template Type .....	14-2
14.3	Creating a Compliance Template .....	14-5
14.4	Apply a Compliance Template to a Project.....	14-6
14.5	Using Assets Associated with Compliance Templates (Prescribed Assets).....	14-7
14.6	Running a Project Compliance Report.....	14-9

## 15 Integration with Code Compliance Inspector

15.1	Overview .....	15-1
15.1.1	Overview of Steps for Generating CCI Reports and Publishing them to OER .....	15-2
15.2	Checking Compliance and Synchronizing Report Data in OER.....	15-3
15.2.1	Configuring an OER Artifact Store for CCI Reports .....	15-3
15.2.2	Configuring the cci-oesynch.properties File.....	15-3
15.2.3	Running the Check Compliance Command .....	15-4
15.2.4	Harvesting the SOA or AIA Projects into OER .....	15-5
15.2.5	Running the CCI-OER Synch Command .....	15-5
15.2.6	Copying the Files .....	15-6
15.3	Configuring Reports for Access from the OER Reports Menu .....	15-7
15.3.1	Reviewing the Reports .....	15-10

## 16 Configuring Policy Management

16.1	Overview .....	16-1
16.2	Activating the Policy Management Feature.....	16-2
16.3	Creating a Policy Type .....	16-3
16.4	Adding the Policy Assertion Element to the Policy Type.....	16-3
16.5	Creating a Policy .....	16-5
16.6	Adding the Applied Asset Policies Element to a Type .....	16-6
16.7	Applying a Policy to an Asset.....	16-7
16.8	Viewing Policy Status.....	16-9
16.9	Validating Policy Assertions .....	16-9
16.10	Running the Oracle Enterprise Repository Policy Reports.....	16-12

## 17 Enterprise Manager Integration Utility

17.1	Overview .....	17-1
17.1.1	Prerequisites .....	17-1
17.1.2	Obtaining the Enterprise Manager Integration Utility.....	17-2



17.1.3	High Level Use Cases.....	17-2
17.2	Using the Enterprise Manager Integration Utility .....	17-3
17.2.1	Running from Command Line .....	17-3
17.2.2	Scheduling from Enterprise Manager .....	17-4
17.2.3	Metric Publishing.....	17-5
17.2.3.1	Oracle Service Bus .....	17-5
17.2.3.2	BPEL PM .....	17-5
17.2.3.3	WebLogic Web Services.....	17-6
17.2.3.4	Oracle SOA Suite 11g .....	17-6
17.2.3.5	Artifacts Creation .....	17-6
17.2.3.6	Endpoint Creation .....	17-6
17.2.3.7	Metrics to Update .....	17-6
17.3	Configuring the Enterprise Manager Integration Utility .....	17-7
17.3.1	Setting the Repository and Enterprise Manager Connection Information for the Command-line Utility 17-7	
17.3.2	Advanced Configuration .....	17-8
17.3.2.1	Metric Mappings.....	17-8
17.3.2.2	Target Finders .....	17-9
17.3.2.3	Logging .....	17-10
17.4	Encrypting the Configuration File Passwords.....	17-10
17.5	Known Issue .....	17-11
17.5.1	Using Incorrect Encrypted Password .....	17-11

## 18 Configuring Metrics

18.1	Overview .....	18-1
18.1.1	Metrics Elements.....	18-1
18.1.2	Adding a Reusable Metrics Tab.....	18-2
18.2	Validate the Elements on the Asset Type .....	18-3
18.3	Running Queries Against the Database.....	18-3
18.3.1	dbqueries.....	18-5

## 19 Configuring Assets in Progress

19.1	Overview .....	19-1
19.2	Viewing Assets in Progress .....	19-2
19.3	Enabling Assets in Progress .....	19-3
19.3.1	Set Default Asset Status .....	19-4
19.4	Disabling Assets in Progress .....	19-4
19.5	Enabling Viewing of Assets in Progress Through Oracle JDeveloper .....	19-4

## 20 Configuring Contract Management

20.1	Overview .....	20-1
20.1.1	Prerequisites .....	20-1
20.2	Import the Contract Management Solution Pack.....	20-2
20.3	Instructions for Service Providers .....	20-2
20.3.1	Create the Contract.....	20-2
20.3.2	Edit the Contract .....	20-2

20.3.3	Create Relationships.....	20-2
20.3.3.1	Terms of Use.....	20-2
20.4	Instructions for Service Consumers .....	20-3

## 21 Configuring Oracle Enterprise Repository to Exchange Metadata with the Oracle Service Registry

21.1	Getting Started With the Oracle Registry Repository Exchange Utility .....	21-1
21.1.1	What is the Oracle Registry Repository Exchange Utility? .....	21-1
21.1.1.1	Valid Metadata Entities .....	21-2
21.1.1.2	Prerequisites .....	21-2
21.1.2	Example Use Cases .....	21-2
21.2	Configuring the Oracle Registry Repository Exchange Utility .....	21-3
21.2.1	Installing and Configuring the Oracle Registry Repository Exchange Utility .....	21-3
21.2.1.1	Install the Oracle Registry Repository Exchange Utility .....	21-3
21.2.1.2	Configure Using the Asset Type Name.....	21-4
21.2.2	Installing Oracle Service Registry .....	21-4
21.2.3	Configuring the Oracle Registry Repository Exchange Utility Configuration File	21-4
21.2.3.1	Setting the Repository Connection Information .....	21-5
21.2.3.2	Setting the Registry Connection Information.....	21-5
21.2.3.3	Setting the Repository Query.....	21-6
21.2.3.3.1	Query by Name .....	21-6
21.2.3.3.2	Query by Registration Status.....	21-6
21.2.3.3.3	Query by Categorizations .....	21-7
21.2.3.3.4	Query by Filters .....	21-7
21.2.3.4	Setting the Destination Registries .....	21-9
21.2.3.5	Setting the Registry Query .....	21-10
21.2.3.6	Setting the Source Registry .....	21-11
21.2.4	Configuring Oracle Enterprise Repository Categorizations in the UDDI Mappings File 21-11	
21.2.5	Configuring the tModelKey UDDI Setting .....	21-11
21.2.6	Understanding the Oracle Registry Repository Exchange Utility's Property File	21-15
21.2.7	XU Support .....	21-16
21.2.7.1	XU Support for Manually-Created Services .....	21-17
21.2.7.2	XU Support for Non-WSDL Services Harvested from Oracle Service Bus.....	21-17
21.2.7.3	Configuring XU to Publish Non-Harvested or Custom Asset Types and Relationships 21-18	
21.3	Using the Oracle Enterprise Repository Exchange Utility .....	21-19
21.3.1	Running the Oracle Registry Repository Exchange Utility .....	21-20
21.3.1.1	Invoking the Oracle Registry Repository Exchange Utility Using Workflows .....	21-21
21.3.1.1.1	Prerequisites.....	21-21
21.3.1.1.2	Synchronizing Using Timer Based Workflows.....	21-21
21.3.1.1.3	Synchronizing When Events are Triggered.....	21-22
21.3.1.1.4	Example Use Cases .....	21-22
21.3.2	How the Exchanged Metadata Is Synchronized .....	21-23
21.3.2.1	Synchronizing the Metadata Published from Oracle Enterprise Repository to Oracle Service Registry 21-23	
21.3.2.1.1	Business Entities .....	21-23

21.3.2.1.2	Service Keys .....	21-24
21.3.2.1.3	Endpoint .....	21-24
21.3.2.1.4	Categorizations .....	21-25
21.3.2.1.5	Registration and Active Status .....	21-25
21.3.2.1.6	Sample Flow of Metadata from Oracle Enterprise Repository to Oracle Service Registry 21-27	
21.3.2.2	Synchronizing the Metadata from Oracle Service Registry to Oracle Enterprise Repository 21-27	
21.3.2.2.1	Business Entities .....	21-28
21.3.2.2.2	Endpoint .....	21-28
21.3.2.2.3	Categorizations .....	21-28
21.3.2.2.4	Sample Flow of Metadata from Oracle Service Registry to Oracle Enterprise Repository 21-28	
21.3.3	Searching for Oracle Service Registry Exchanged Metadata in Oracle Enterprise Repository 21-29	
21.3.4	Checking the Oracle Registry Repository Exchange Utility Log File .....	21-30
21.3.5	Known Issues .....	21-30
21.3.5.1	Resynchronizing Oracle Service Registry Services.....	21-31
21.3.5.2	Publishing Services to Oracle Service Registry .....	21-31
21.3.5.3	Running an Incorrect Version of Java.....	21-31
A.1	Creating and Configuring an Upload Submission Directory.....	A-1
A.1.1	Asset Upload System Settings .....	A-1
A.1.2	Creating and Configuring an Artifact Store .....	A-4
A.1.3	Selecting a New Artifact Store .....	A-4
A.2	Creating an Asset Audit Log.....	A-4
A.3	Oracle Enterprise Repository Default Artifact Store Integration.....	A-4
A.4	Maintaining XSLT Templates for use with the Export-to-PDF Feature.....	A-6
A.4.1	Maintaining XSL:FO Templates for use with Oracle Enterprise Repository .....	A-6
A.5	Support for UTF-8/Unicode Characters .....	A-8
A.6	Data Rules .....	A-11
A.6.1	Flags on Asset Type Elements .....	A-11
A.6.2	Grandfathered Data.....	A-11
A.6.3	Element Storage .....	A-12
A.6.4	Elements Required to Exist on Asset Types.....	A-12
A.6.5	Elements Tied to Functionality .....	A-13
A.6.6	Description of the Asset Type Catalog .....	A-15
A.7	Harvester Properties.....	A-16



---

---

# Preface

*Oracle Fusion Middleware Administrator's Guide for Oracle Enterprise Repository* provides the information needed to complete the four types of administration tasks in Oracle Enterprise Repository: infrastructure administration, asset lifecycle administration, security administration, and advanced asset lifecycle administration.

## Audience

This document is intended for individuals who administering an Oracle Enterprise Repository installation or managing the asset lifecycle process.

## 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 that have purchased support 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.

## Related Documents

For more information, see the following documents in the Oracle Enterprise Repository 12c documentation set:

- *Oracle Fusion Middleware Concepts Guide for Oracle Enterprise Repository*
- *Oracle Fusion Middleware Installation Guide for Oracle Enterprise Repository*
- *Oracle Fusion Middleware Upgrade Guide for Oracle Enterprise Repository*
- *Oracle Fusion Middleware Developer's Guide for Oracle Enterprise Repository*

## Conventions

The following text conventions are used in this document:

<b>Convention</b>	<b>Meaning</b>
<b>boldface</b>	Boldface type indicates graphical user interface elements associated with an action, or terms defined in text or the glossary.
<i>italic</i>	Italic type indicates book titles, emphasis, or placeholder variables for which you supply particular values.
monospace	Monospace type indicates commands within a paragraph, URLs, code in examples, text that appears on the screen, or text that you enter.

---

---

# What's New in This Guide

This preface introduces the new and changed features of Oracle Enterprise Repository and other significant changes that are described in this guide, and provides pointers to additional information. This document is the new edition of the formerly titled *Oracle Fusion Middleware Configuration Guide for Oracle Enterprise Repository*.

For a list of known issues (release notes), see the "Known Issues for Oracle SOA Products and Oracle BPM Products for 12c Release 1 (12.1.3.0.0)" at <http://www.oracle.com/technetwork/middleware/soasuite/documentation/releasenotes121300-2124738.html>.

## New and Changed Features for 12c Release 1 (12.1.3)

For Oracle Enterprise Repository 12c Release 1 (12.1.3), this guide has been updated to include the following new and changed features:

- [Chapter 8, "Configuring and Using Automated Harvesting"](#) describes:
  - The OER Harvesters now support harvesting REST services
  - OER includes the harvester for use with Oracle Service Bus
  - Added [Section 8.3.1.3, "WADL"](#), which discusses harvesting WADL artifacts
- [Chapter 13, "Configuring Oracle Enterprise Repository Express Workflows"](#) has been added to describe the new simplified workflows for this release.

## Other Significant Changes in this Document for 12c Release 1 (12.1.3)

For Oracle Enterprise Repository 12c Release 1 (12.1.3), this guide has been updated in the following ways:

- [Chapter 2, "System Settings Overview"](#) describes additional system settings
- Administration tasks from the *Oracle Fusion Middleware User's Guide for Oracle Enterprise Repository* and the *Oracle Fusion Middleware Integration Guide for Oracle Enterprise Repository* from the previous release have been moved to this guide





# Part I

---

## Administering the Infrastructure

This part describes basic administration tasks for the Oracle Enterprise Repository infrastructure.

This part includes the following chapters:

- [Chapter 1, "Basic Configuration"](#)
- [Chapter 2, "System Settings Overview"](#)
- [Chapter 3, "Customizing Oracle Enterprise Repository"](#)
- [Chapter 4, "Configuring E-Mail Notifications and Distribution Lists"](#)
- [Chapter 5, "Import Export Tool"](#)
- [Chapter 6, "Configuring Reporting with BI Publisher"](#)



---

---

# Basic Configuration

This chapter describes the basics of how to configure Oracle Enterprise Repository by introducing you to the terminology and common tasks of an Oracle Enterprise Repository administrator.

This chapter contains the following sections:

- [Section 1.1, "Overview"](#)
- [Section 1.2, "Departments"](#)
- [Section 1.3, "Roles"](#)
- [Section 1.4, "Access Settings"](#)
- [Section 1.5, "Users"](#)
- [Section 1.6, "Projects"](#)
- [Section 1.7, "Sessions"](#)
- [Section 1.8, "Types"](#)
- [Section 1.9, "Accessing Oracle Enterprise Repository Diagnostics Page"](#)
- [Section 1.10, "Install Java Web Start"](#)

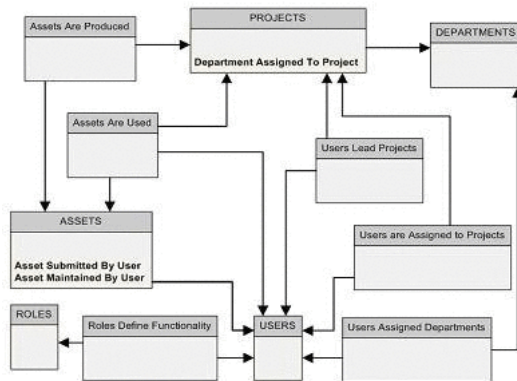
## 1.1 Overview

In Oracle Enterprise Repository, all users are assigned roles. Roles define the Oracle Enterprise Repository functionality that can be exercised by each user. In order for a user to access any files associated with assets in Oracle Enterprise Repository, the user must be assigned to a project.

Projects are the primary means of gathering metrics in Oracle Enterprise Repository. Oracle Enterprise Repository tracks assets produced by projects, as well as assets consumed by projects. Oracle Enterprise Repository users are assigned to projects, and when a user submits a new asset, they are prompted for the producing project. Similarly, when a user wants to reuse an asset, they are prompted for the project on which the asset is reused. Both Users and Projects are assigned to Departments. This is convenient from a reporting standpoint, as organizations can then track the production and consumption of reusable assets to a specific department.

Figure 1–1 describes the relationships between Users, Roles, Projects, Assets, and Departments.

**Figure 1–1 Relationship Diagram**



This figure illustrates the relationship diagram.

\*\*\*\*\*

## 1.2 Departments

You need to set up or create departments before creating users and projects. This section contains the following topics:

- [Section 1.2.1, "Creating a New Department"](#)
- [Section 1.2.2, "Editing Department Information"](#)
- [Section 1.2.3, "Deleting a Department"](#)

### 1.2.1 Creating a New Department

To create a new department, perform the following steps from the Oracle Enterprise Repository Admin screen:

1. In the left panel, click **Departments**.
2. Click **Create New**. The Create New Department dialog is displayed.
3. Enter the appropriate information into the **Name** and **Description** fields.
4. When finished, click **Save**.

### 1.2.2 Editing Department Information

To edit a department information, perform the following steps from the Oracle Enterprise Repository Admin screen:

1. In the left panel, click **Departments**.
2. Click **List All**, or search for a particular department.
3. Select the department to be edited from the list in the right panel. The department's detail page is displayed.
4. Click **Edit**. The Edit Department dialog is displayed.
5. Update the **Name** and **Description** fields as necessary.
6. When finished, click **Save**.

### 1.2.3 Deleting a Department

To delete a department in Oracle Enterprise Repository, perform the following steps from the Oracle Enterprise Repository Admin Screen:

1. In the left panel, click **Departments**.
2. Click **List All**, or search for departments that you want to delete.
3. To delete a department:
  - To delete departments from the Search Results screen, select the check box next to each department that you want to delete, and then click **Delete**. You can delete multiple departments simultaneously using this method.
  - To delete a department from its detail page, open the detail page for the department you want to delete, and then click **Delete**.

## 1.3 Roles

Clicking the Roles link on the Oracle Enterprise Repository Admin page allows administrators to set up roles for an organization's various functions. Oracle Enterprise Repository access settings are established for each role, and each user is assigned a specific role based on his/her function within the organization.

Users assigned the admin role have unlimited access to all Oracle Enterprise Repository functions. By default, in either a new installation or a migration from a previous version, the user account identified as admin is assigned the admin role in order to facilitate initial installation and configuration. It is highly recommended that you limit the assignment of admin privileges to the original Oracle Enterprise Repository Admin user account and no more than one additional account. The various default roles established in Oracle Enterprise Repository offer a variety of privileges to satisfy the functions and responsibilities for any and all Oracle Enterprise Repository users. These roles may be customized to suit the particular needs of your organization.

- Section 1.3.1, "User Role Descriptions"
- Section 1.3.2, "Creating a New Role"
- Section 1.3.3, "Editing a Role"
- Section 1.3.4, "Deleting a Role"

### 1.3.1 User Role Descriptions

The various default roles shipped with Oracle Enterprise Repository are listed below. Users with access to the Admin tab can change role names to suit organizational needs. The permissions granted to these roles can be changed in the Access Settings section.

- User
  - Anyone with an Oracle Enterprise Repository user name and password is considered to be a user. This role can be assigned as the default role for any new users at the time the user is created. All Oracle Enterprise Repository users can:
    - View news about the company's reuse initiatives
    - Locate, evaluate, and use/download assets
    - View projects
    - Generate reports

- Submit assets to the registrar
- **Access Administrator**

The access administrator creates all Oracle Enterprise Repository users, and assigns permissions to them. The access administrator must be familiar with the functions of the Admin Tab. Typically, access administrators can:

  - Create, view, and edit users and permissions
  - Generate reports
- **Advance Submitter**

The advanced submitter role is typically assigned to asset builders and harvesters. Asset builders focus on building the asset content base. They respond to organization-wide asset needs as well as individual project needs. Harvesters study post-implementation projects for asset reuse potential. Typically, advanced submitters can:

  - Locate, evaluate, and use/download assets
  - View projects
  - Generate reports
  - Submit assets to the registrar
  - Edit asset metadata prior to asset registration
- **Registrar**

The registrar is responsible for the acceptance or rejection of any asset. There may be more than one person functioning as a repository registrar, depending on the functions addressed. Typically, registrars can:

  - Locate, evaluate, and use/download assets
  - View projects
  - Generate reports
  - Submit assets to the registrar
  - Edit asset metadata prior to asset registration
  - Accept assets for the registration process
  - Approve tabs
  - Register assets
  - Edit access settings
- **Registrar Administrator**

The registrar administrator is able to establish Types within Oracle Enterprise Repository, using the Type Manager. Typically, registrar administrators can:

  - Locate, evaluate, and use/download assets
  - View projects
  - Generate reports
  - Submit assets to the registrar
  - Edit asset metadata prior to asset registration
  - Accept assets for the registration process

- Approve tabs
- Register assets
- Edit access settings
- Edit Artifact Stores and Types
- **Project Administrator**

Oracle Enterprise Repository tracks asset use at the project level in order to maintain a history for maintenance purposes. Project administrators create projects and assign users to projects using the Projects tab. Project administrators can close projects, indicate which assets were deployed, review, validate, and optionally override the predicted and consumer-reported value of assets used in a project through the project-reported asset value user interface. Typically, project administrators can:

  - Create, edit, and view projects
  - Generate reports
- **System Administrator**

The system administrator configures Oracle Enterprise Repository for use, including, possibly, installation as well as the post-installation configurations. The system administrator typically can:

  - Enable and edit system settings
  - Generate reports

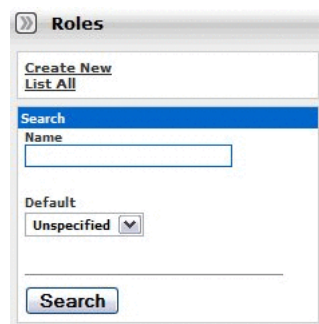
(The system administrator role should not be assigned to general users.)

### 1.3.2 Creating a New Role

This procedure is performed on the Oracle Enterprise Repository Admin screen.

1. In the left panel, click **Roles**. The Roles section is displayed, as shown in Figure 1-2.

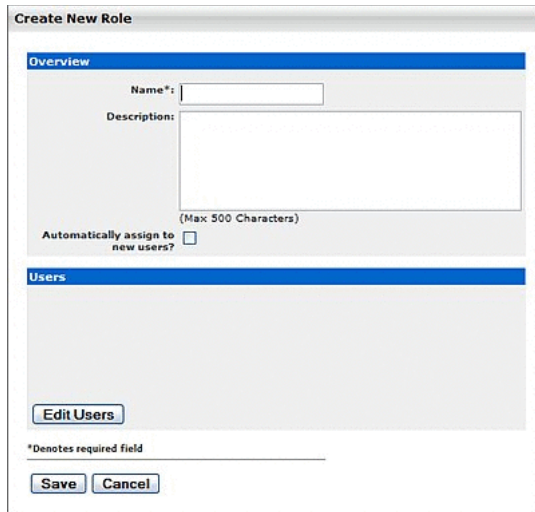
**Figure 1-2 Roles Section**



This figure illustrates the Roles section.

- \*\*\*\*\*
2. Click **Create New**. The Create New Role dialog is displayed, as shown in Figure 1-3.

**Figure 1–3 Create New Role Dialog**

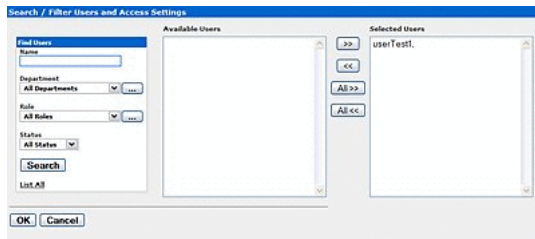


This figure illustrates the Create New Role dialog.

\*\*\*\*\*

3. In Figure 1–3, enter a role name and description in the text fields in the Overview section.
4. If necessary, select the **Automatically assign to new users?** box.
5. Click the **Edit Users** in the Users section. The Add New Users dialog is displayed, as shown in Figure 1–4.

**Figure 1–4 Add New Users Dialog**



This figure illustrates the Add New Users dialog.

\*\*\*\*\*

6. Use **Search** or click **List All** to locate users.
7. Use the arrow buttons to move users between the **Available Users** and **Selected Users** columns.
8. When finished, click **Save**.

To add access settings to the role, see Section 1.4.1, "Granting Permissions to a Role".

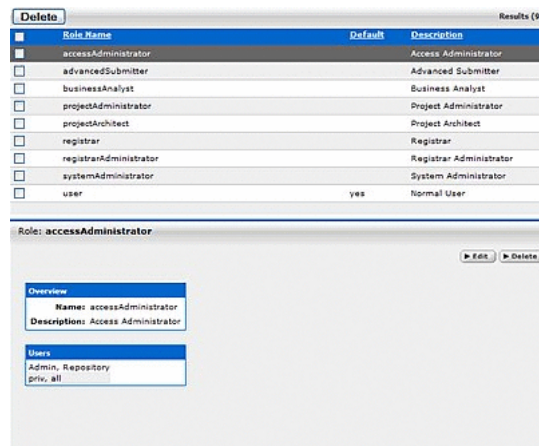
### 1.3.3 Editing a Role

This procedure is performed on the Oracle Enterprise Repository Admin screen.

1. In the left panel, click **Roles**. A list of roles is displayed in the main pane, as shown in Figure 1–5.



**Figure 1-5 List of Roles**

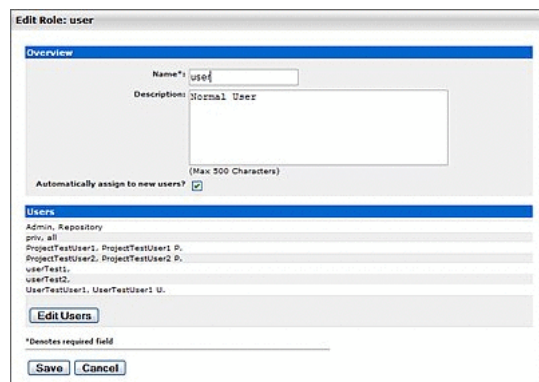


This figure illustrates the list of roles.

\*\*\*\*\*

2. Use Search or browse the list to locate the Role to be edited.
3. Click the role to display its information in the lower pane.
4. Click the **Edit** button in the role detail (bottom frame). The Edit Role dialog is displayed, as shown in Figure 1-6.

**Figure 1-6 Edit Role: User Dialog**



This figure illustrates the Edit Role: user dialog.

\*\*\*\*\*

5. When finished, click **Save**.

### 1.3.4 Deleting a Role

This procedure is performed on the Oracle Enterprise Repository Admin screen.

---

**Note:** Deleting roles requires the Delete permission. See [Section 1.4, "Access Settings"](#) for more information about permissions and access settings.

---

1. In the left panel, click **Roles**. A list of roles is displayed in the main pane.
2. Use **Search** or browse the list to locate the role to be deleted.
3. Option 1:
  - In the list in the upper frame, place a check in the box next to each role to be deleted.
  - Click **Delete**.
4. Option 2:
  - Select the role in the list in the upper pane to display its information in the lower pane.
  - Click the **Delete** button in the role detail (lower frame).
5. Click **OK** in the delete confirmation.

---

---

**Note:** Deleted roles no longer appear in Access Settings.

---

---

## 1.4 Access Settings

Access Settings allow the administrator to assign permissions to each role. Access settings identify the functions and responsibilities of a role, and determine each role's level of access to assets and asset management features in Oracle Enterprise Repository.

System safeguards prevent an administrator from inadvertently modifying permissions in such a way that would prevent his/her subsequent access to Access Settings.

- **Basic Access Settings (BAS)**

Control the access to assets in Oracle Enterprise Repository across the board, based on the roles to which users are assigned. Access to Oracle Enterprise Repository tools, such as the Asset Editor or the Type Manager, is similarly controlled. Basic Access Settings establish access across the system. For example, permission to view assets applies to all assets; as does permission to use/download assets. Many actions in OER leave audit trails, so BAS may be sufficient for you if you have a small number of projects, and internal users.
- **Custom Access Settings (CAS)**

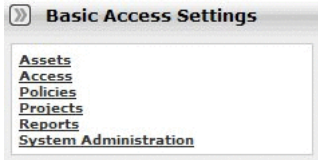
Establish Advanced Role-based Access Control to restrict access to specific assets or groups of assets in Oracle Enterprise Repository to specific user roles.
- Section 1.4.1, "Granting Permissions to a Role"
- Section 1.4.2, "Editing Permissions"
- Section 1.4.3, "Verifying User Permissions"
- Section 1.4.4, "Security Considerations"

### 1.4.1 Granting Permissions to a Role

This procedure is performed on the Oracle Enterprise Repository Admin screen.

1. In the left panel, click **Basic Access Settings**.
2. Select the appropriate grouping from the list in the Basic Access Settings section.

Figure 1-7 Basic Access Settings

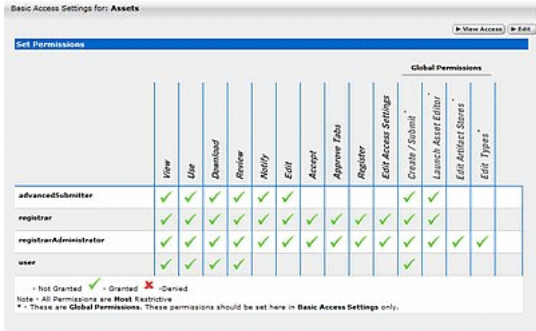


This figure illustrates the Basic Access Settings pane.

\*\*\*\*\*

The selected grouping's access settings are displayed in the main pane.

Figure 1-8 Set Permissions

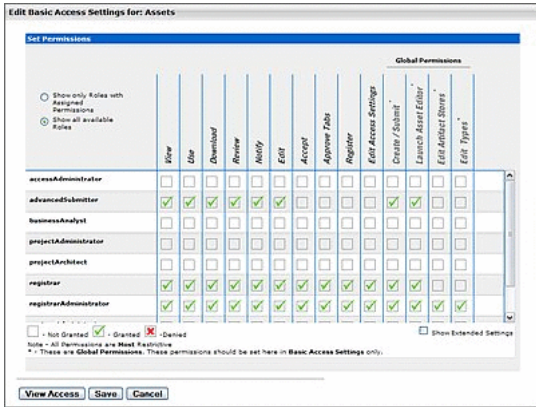


This figure illustrates the Set Permissions page.

\*\*\*\*\*

- 3. Click the **Edit** button (upper right corner of the main pane.) The Edit Basic Access Settings for:... dialog is displayed.

Figure 1-9 Set Permissions



This figure illustrates the Set Permissions dialog.

\*\*\*\*\*

Roles are listed on the left; Permissions are listed along the top.

- 4. Select the appropriate option to determine which roles are displayed.

**Figure 1–10 Set Permissions**



This figure illustrates the Set Permissions pane.

\*\*\*\*\*

5. Select the role to which permissions are to be granted.
6. Use the check boxes to assign the appropriate permission.
  - One click on an empty box adds a check mark (Granted)
  - A second click adds an X (Denied).
  - A third click clears the box (Not Granted).
7. When finished, click **Save**.

## 1.4.2 Editing Permissions

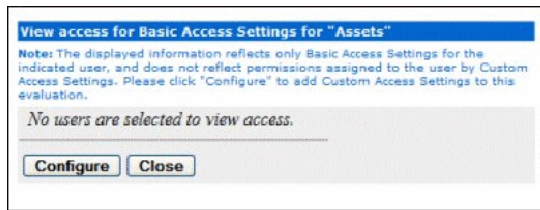
See: Section 1.4.1, "Granting Permissions to a Role"

## 1.4.3 Verifying User Permissions

This procedure is performed on the Oracle Enterprise Repository Admin screen.

1. In the left panel, click **Basic Access Settings**.
2. Select the appropriate grouping from the list in the Basic Access Settings section.  
The selected grouping's access settings are displayed in the main pane.
3. Click the **View Access** button (upper right corner of the main pane). A message dialog is displayed.

**Figure 1–11 View Access for Basic Access Settings for Assets**

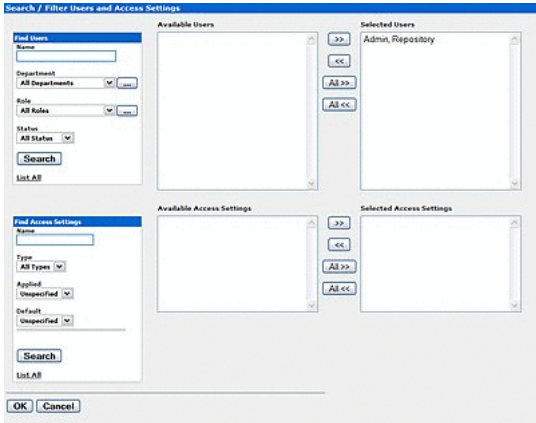


This figure illustrates the View Access for Basic Access Settings for "Assets" dialog.

\*\*\*\*\*

4. Click **Configure**. The Search / Filter Users and Access Settings dialog is displayed.

Figure 1–12 Search / Filter Users and Access Settings



This figure illustrates the Search/Filter Users and Access Settings page.

\*\*\*\*\*

- 5. Use Search or click **List All** in the Find Users section to locate the user(s) whose permissions are to be verified.
- 6. Use the Arrow buttons to move user(s) from the Available Users column to the Selected Users column.
- 7. Click **OK**. The selected user's access settings are displayed.

Figure 1–13 User Access Settings



This figure illustrates the User Access Settings page.

\*\*\*\*\*

- 8. If multiple users are selected, clicking the icon next to a user's name displays information specific to that user.
- 9. When finished, click **Close**.

### 1.4.4 Security Considerations

Role-based Access Control (RBAC) allows Oracle Enterprise Repository to track asset usage and production on a by-user basis and personalizes the presentation of assets, limiting visibility to designated portions of Oracle Enterprise Repository. However, Role-based Access Control is not intended to provide security for asset metadata, nor is it intended to supplement the security of underlying asset repositories.

Oracle Enterprise Repository access settings provide no additional security for otherwise unsecured assets or data. For maximum security, asset metadata should be managed in a manner that prevents access to the metadata via an unsecured repository. Confidential information should be embedded in the asset's payload, or attached as documentation files and hosted in a secure repository.

If you choose to circumvent an authentication and permission challenge from the underlying asset repository, you do so at your own risk.

The following actions also have security implications:

- Configuring SCM systems for access via a single Oracle Enterprise Repository user account
- Allowing system access by unapproved users
- Creating guest accounts
  - Guest accounts may be forbidden under certain Oracle Enterprise Repository licenses.

Any of these actions has the potential to open the contents of Oracle Enterprise Repository to anyone with network access to Oracle Enterprise Repository.

#### 1.4.4.1 Access Setting and the Asset Editor

Depending on the Access Settings in use, users with access to the Asset Editor may be able to see relationships to assets that are otherwise invisible to them in the repository. In this situation, the name of the hidden asset is visible, as is the detail of the relationship between that asset and an asset to which the users have access. No other information on the hidden asset is visible in the Asset Editor. In this situation, it is possible to delete the relationship between the visible and the hidden asset. However, given that the invisible asset is inaccessible to the users in question, the relationship cannot be restored by those users.

#### 1.4.4.2 Permissions

Permissions may be:

- Granted (Permission is explicitly granted)
- Denied (Permission is explicitly denied)
- Not Granted (Permission is not explicitly granted nor denied.)

The table below identifies permissions that may be managed within Basic Access Settings.

#### Access Definitions

Assets: Tools regulated: Oracle Enterprise Repository Asset screen, Asset Editor, Type Manager.

Permission	Description
View	Assets may be viewed in Oracle Enterprise Repository.
Use	Displays/enables the Use - Download button in Oracle Enterprise Repository.
Download	Asset files may be downloaded. Provides visibility to file links after pressing the Use - Download button.
Review	Reviewing assets using Oracle Enterprise Repository.
Edit	Editing assets in the Asset Editor.
Accept	Accepting assets. This button, in the Asset Editor, moves assets from the Submitted -- Pending Review to the Submitted -- Under Review queue.

Permission	Description
Approve Tabs	Approving metadata for assets. This button is available at the bottom of each tab in the Asset Editor display of each asset. Approving a tab logs the name of the approver and the date in the metadata, moves the asset out of the Submitted - Under Review queue for that tab, and changes the tab's text label to green.
Register	Registering assets. This permission provides the user with the ability to register all assets in the Asset Editor.
Edit Access Settings	Applicable only with Custom Access Settings. Provides the ability to assign Custom Access Settings to assets and files.
Create/Submit	Gives a user the ability to submit assets through or create new assets through the Asset Editor.
Launch Asset Editor	Gives a user the ability to open the Asset Editor tool. One must use this tool to edit and register assets.
Edit Artifact Store	Gives a user the ability to create and edit artifact stores (through the Asset Editor).
Edit Types	Gives a user the ability to configure asset type/compliance template metadata. Requires the Launch Asset Editor permission.
Notify	Determines if the user is allowed to send an ad-hoc email to users.
Global Permissions	These are permissions that do not apply to individual assets and cannot be overridden by custom access settings.

Access: Tools regulated: Oracle Enterprise Repository Admin screen: Users, Sessions, Roles, Departments, File Stores, Basic Access Settings, Custom Access Settings

Permission	Description
View	Viewing entries in the sections noted above.
Edit	Editing entries in the sections noted above.
Create	Creating users, roles, departments, file stores, and custom access settings.
Delete	Deleting sessions, roles, and custom access settings.

Projects: Tools regulated: Oracle Enterprise Repository Projects screen

Permission	Description
View	Viewing projects on the Projects page.
Edit	Editing projects on the Projects page.
Create	Creating projects on the Projects page.
Apply Template	Apply a compliance template to a project.

Policies: Tools regulated: Oracle Enterprise Repository Asset screen

Permission	Description
Apply Policy	Allows user to apply a policy to other assets.

Reports: Tools regulated: Oracle Enterprise Repository Reports screen

Permission	Description
View	Viewing reports.

System Administration: Tools regulated: Oracle Enterprise Repository Admin screen: System Settings, Email Templates.

Permission	Description
Edit	Ability to edit general, security, authentication, and file store settings.
Enable	Ability to enable new general, security, authentication, and file store settings.

## 1.5 Users

This section describes the following topics:

- [Section 1.5.1, "Creating a New User"](#)
- [Section 1.5.2, "Viewing User Information"](#)
- [Section 1.5.3, "User Search"](#)
- [Section 1.5.4, "Cloning a User"](#)
- [Section 1.5.5, "Editing User Information"](#)
- [Section 1.5.6, "Deleting a User"](#)

### 1.5.1 Creating a New User

To create a new user, perform the following steps on the Oracle Enterprise Repository Admin screen:

1. In the left panel, click **Users**.
2. Click **Create New** in the Users section. The Create New User dialog is displayed.
3. Enter the appropriate information in each of the text fields in the **Overview** section, as shown in Figure 1–14.

---

**Note:** Ensure that you supply values for each of the required fields, denoted by an asterisk.

---

**Figure 1–14** *Overview Section*

The screenshot shows a form titled "Overview" with the following fields and options:

- Username\*:
- First Name:
- Middle Name:
- Last Name\*:
- Email\*:
- Phone:
- Status: **Active** (dropdown menu)
- Password\*:
- Must change password on next login
- Password never expires

This figure illustrates the Overview section.



\*\*\*\*\*

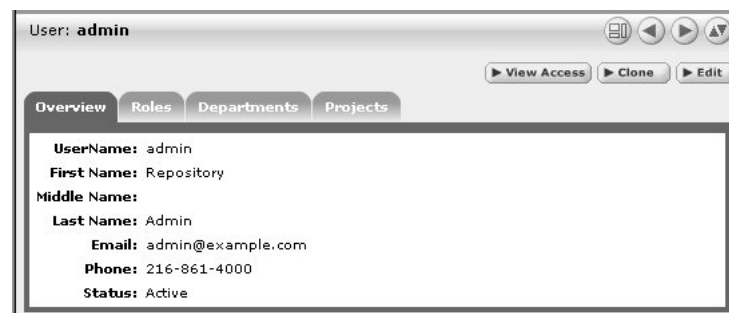
4. Select **Active** in the Status list.
5. In the Roles section, assign roles to the new user by using the arrow buttons to move items from the **Available Roles** column to the **Selected Roles** column. (The **User** role is the default role for all new users.)
6. In the Departments section, assign the new user to departments by using the arrow buttons to move items from the **Available Departments** column to the **Selected Departments** column.
7. When finished, click **Save**.

## 1.5.2 Viewing User Information

To view a user's information, perform the following steps on the Oracle Enterprise Repository Admin screen:

1. In the left panel, click **Users**.
2. Use **Search** or **List All** to locate the user(s) to be viewed.  
The list of users is displayed in the main pane.
3. Select a user from the list. The user's information is displayed in the lower pane.
4. Click **Hide/Display Search Results** button to expand the User detail to fill the main pane.
5. Click the first button in Figure 1–15 to switch to the tabbed view of the User detail.

**Figure 1–15** *User Details Page*



This figure illustrates the User Details page.

\*\*\*\*\*

6. Click the first button in Figure 1–15 to switch back to the standard view of the User detail.
7. Click the second and third button in Figure 1–15 to scroll through the list of users.

## 1.5.3 User Search

This procedure is performed on the Oracle Enterprise Repository Admin screen.

1. In the left panel, click **Users**.
2. Enter appropriate text in the **Name** field.
3. Use the **Department**, **Role**, and **Status** lists as necessary to narrow the search.

4. Click the **Search** button. Search results appear in the list in the upper frame of the main pane.

### 1.5.4 Cloning a User

Cloning a user provides an easy way for administrators to quickly duplicate user accounts.

This procedure is performed on the Oracle Enterprise Repository Admin screen.

1. In the left panel, click **Users**.
2. In the Users section, use **Search** or **List All** to locate the user to be cloned.
3. Click the **Clone** button in the user detail.

The Clone User dialog is displayed. The Overview section is blank, but information in the Roles and Departments sections duplicates that of the cloned user.
4. Enter the appropriate information in each of the text fields in the **Overview** section.
5. If necessary, edit the information in the in the Roles and Departments sections by using the arrow buttons to move items between the **Available** and **Selected** columns.
6. When finished, click **Save**.

### 1.5.5 Editing User Information

This procedure is performed on the Oracle Enterprise Repository Admin screen.

1. In the left panel, click **Users**.
2. Use **Search** or **List All** to locate the user(s) to be edited. The list of users appears in the upper frame of the main pane.
3. Select a user from the list. The user's information is displayed in the lower pane.
4. Click the **Edit** button in the user detail. The Edit User dialog is displayed (same form as the Create New User and Clone User dialogs).
5. Make changes as necessary in the **Overview**, **Roles**, and **Departments** sections.
6. When finished, click **Save**.

### 1.5.6 Deleting a User

To delete a user in Oracle Enterprise Repository, perform the following steps from the Oracle Enterprise Repository Admin Screen:

1. In the left panel, click **Users**.
2. Click **List All**, or search for users that you want to delete.
3. To delete a user:
  - To delete users from the Search Results screen, select the check box next to each user that you want to delete, and then click **Delete**. You can delete multiple users simultaneously using this method.
  - To delete a user from its detail page, open the detail page for the user you want to delete, and then click **Delete**.

## 1.6 Projects

Projects are the primary means of gathering metrics in Oracle Enterprise Repository. Oracle Enterprise Repository tracks assets produced by projects, as well as assets consumed by projects. Oracle Enterprise Repository users are assigned to projects, and when a user submits a new asset, they are prompted for the producing project. Likewise, when a user wants to reuse an asset, they are prompted for the project on which the asset is reused. This allows Oracle Enterprise Repository to generate reports on the reuse savings per project. It also allows Oracle Enterprise Repository to report on the savings generated by asset production teams. Projects are also hierarchical, which allows organizations to, for example, establish a program that may spawn many projects.

Projects are a concept through which you can apply and enforce more advanced governance practices. Oracle Enterprise Repository Compliance Templates (usually taking the form of Architecture Blueprints or Project Profiles) can be applied to projects.

The Oracle Enterprise Repository Projects page provides access to tools for creating and managing projects.

- Section 1.6.1, "Viewing Project Details"
- Section 1.6.2, "Creating a Project"
- Section 1.6.3, "Editing a Project"
- Section 1.6.4, "Reassigning Users/Usage"
- Section 1.6.5, "Locating a Project"
- Section 1.6.6, "Closing a Project"
- Section 1.6.7, "Adding a Relationship to a Project"

### 1.6.1 Viewing Project Details

Project information may be displayed as a series of portlets or tabs:

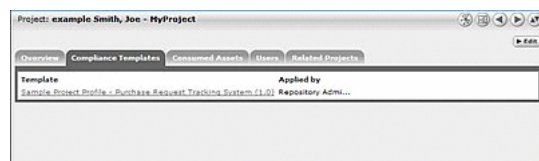
#### Overview

Includes a project Description, and indicates the assigned Department, Start Date, Estimated Hours, and project Status.

#### Compliance Templates

Displays the Compliance Template(s) assigned to the project.

**Figure 1–16** *Compliance Templates Tab*



This figure illustrates the Compliance Templates tab.

\*\*\*\*\*

#### Consumed Assets

Lists any assets used in (or under consideration for use in) the project.

**Figure 1–17 Consumed Assets Tab**



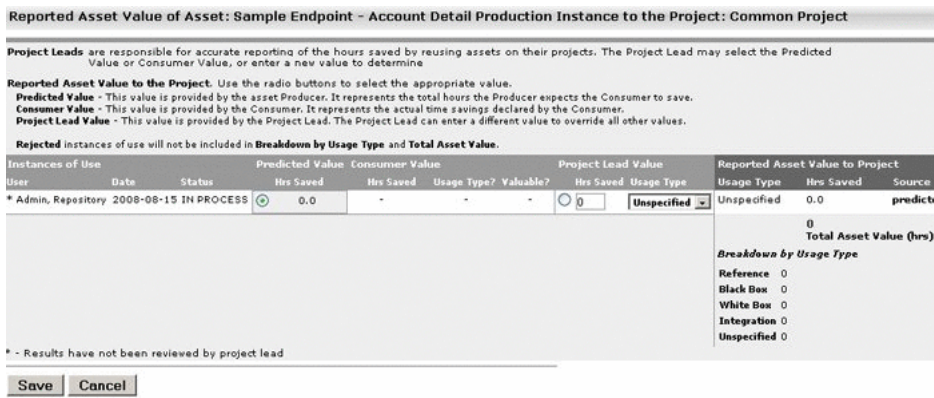
This figure illustrates the Consumed Assets tab.

\*\*\*\*\*

Click the **Zoom** icon next to any listed asset to display its Reported Asset Value.

- Project members can view the Asset Usage Detail.
- Project leaders can view/edit the Reported Asset Value.

**Figure 1–18 Reported Asset Value of Asset**



This figure illustrates the Reported Asset Value of Asset.

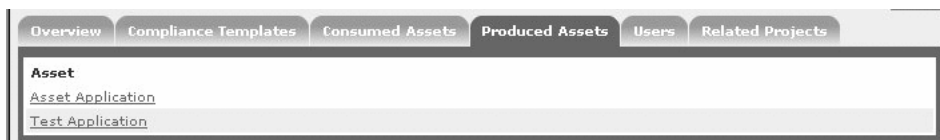
\*\*\*\*\*

Reported Asset Value represents the hours saved by reusing assets on their projects. This value is derived from either the Predicted Value, Consumer Value, or a completely new value determined by the project leader. Only the project leader can set this value.

**Produced Assets**

Lists any assets produced by the project.

**Figure 1–19 Produced Assets Tab**



This figure illustrates the Produced Assets tab.

\*\*\*\*\*

## Users

Lists all users associated with the project, as well as each user's role on the project (leader/members). Project leaders have the ability to assign a Reported Asset Value to assets consumed by the project.

**Figure 1–20 Users Tab**



This figure illustrates the Users tab.

\*\*\*\*\*

## Related Projects

When enabled, lists any related projects, and defines the relationships in parent/child terms.

## 1.6.2 Creating a Project

This procedure is performed in the Projects screen.

1. In the left panel, click **Create New**. The Create New Project dialog is displayed.

**Figure 1–21 Create New Project**

**Create New Project**

**Overview**

Name\*:

Description:   
(Max 4000 Characters)

Estimated Development Hours:

Start Date\*:  ...

Department\*:  ...

Status:

Automatically assign to new users?

**Users** ▶ Edit

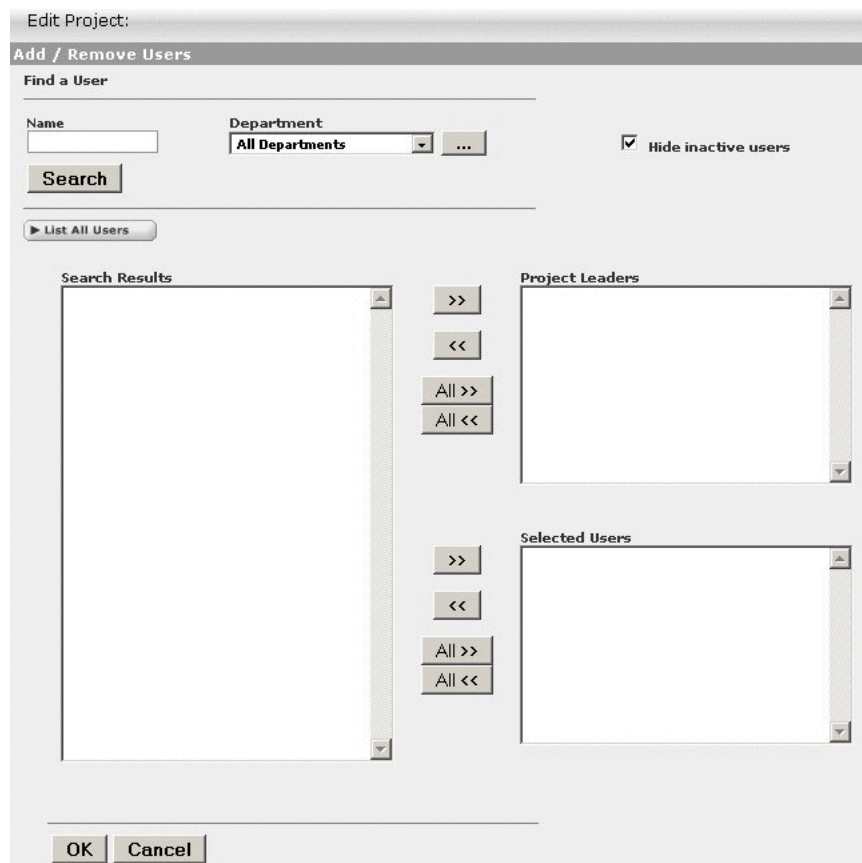
**Related Projects** ▶ Edit

\*Denotes required field

This figure illustrates the Create New Project dialog.

\*\*\*\*\*

2. Enter the appropriate information in the **Name**, **Description**, and **Estimated Development Hours** fields.
3. Click the ... button to open the calendar to select a Start Date.
4. Select a department from the **Department** list.
5. Select the appropriate status in the **Status** list. An Open status indicates that the project is still open for changes. A Closed status would mean that the project is closed and is not open for changes.
6. If necessary, select **Automatically Assign to New Users**.
7. Click **Edit** in the Users section. The Add/Remove Users dialog is displayed, as shown in Figure 1–22.

**Figure 1–22 Add/Remove Users Dialog**

This figure illustrates the Add/Remove Users dialog.

\*\*\*\*\*

8. Use **Search** or **List All Users** to display a list of users in the Search Results section. Use the **Department** list to filter search results.
9. Use the Arrow buttons to move users between the Search Results, Project Leaders, and Selected Users sections.
10. Click **OK**. The assigned Project Leaders and Project Members appear in the Users section of the Create New Project dialog is displayed.

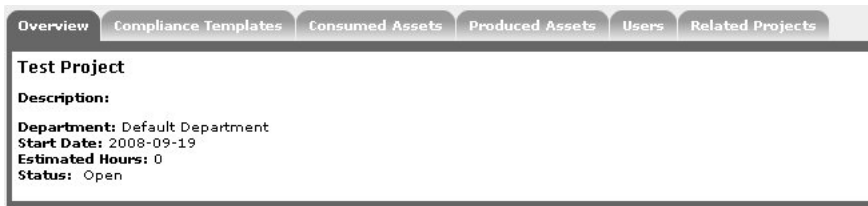
**Figure 1–23 User Dialog**

This figure illustrates the User dialog.

\*\*\*\*\*

11. Click **Save**. The Create New Project dialog closes, and the new project detail appears on the Projects page.

Figure 1–24 Overview Tab



This figure illustrates the Overview tab.

\*\*\*\*\*

### 1.6.3 Editing a Project

This procedure is performed in the Projects page.

1. Use Search or other means to locate the project to be edited.
2. Click **Edit** in the project detail.
3. Edit the project information as necessary. (See Section 1.6.2, "Creating a Project").

### 1.6.4 Reassigning Users/Usage

This procedure is performed in the Projects page.

1. Select the Users tab.

Figure 1–25 Users Tab

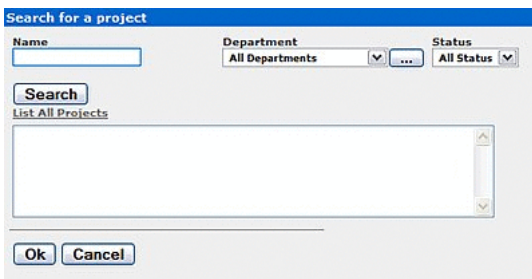


This figure illustrates the Users tab.

\*\*\*\*\*

2. Click **Reassign Users/Usage**. The Reassign dialog is displayed.
3. Use the options to select the appropriate action.
4. Click **Choose a New Project**. The Search for a Project dialog is displayed.

Figure 1–26 Search for a Project



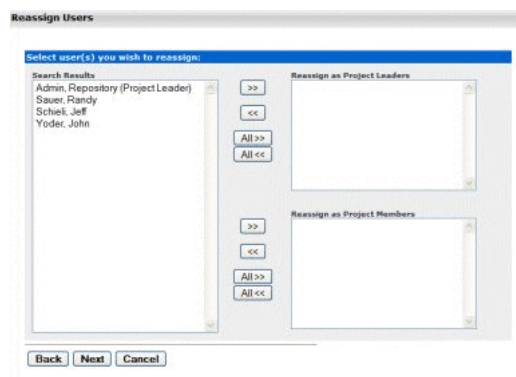
This figure illustrates the Search for a Project dialog.



\*\*\*\*\*

5. Use Search or List All Projects to display a list of projects.
6. Select the project to which the users and/or usages are to be reassigned.
7. Click **OK**.
8. Click **Next**. The Select users you wish to reassign dialog is displayed.

**Figure 1–27 Reassign Users**



This figure illustrates the Reassign Users dialog.

\*\*\*\*\*

9. Use the Arrow buttons to reassign users as Project Leaders and/or Project Members.
10. Click **Next**. A confirmation message appears.
11. Click **Done**.

### 1.6.5 Locating a Project

This procedure is performed on the Projects page.

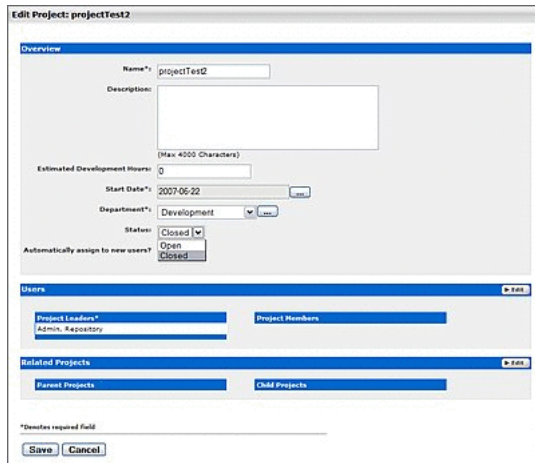
1. Enter a keyword or search term in the Name text box.
2. Use the Department and Status lists as necessary to filter the search.
3. Click **Search**. Search results are listed in the main pane.

### 1.6.6 Closing a Project

This procedure is performed on the Projects page.

1. Select the project to be closed. The project detail is displayed in the bottom frame.
2. Click **Edit**. The Edit Project dialog is displayed.
3. Select **Closed** from the Status list in the Overview section of the Edit Project dialog, as shown in Figure 1–28.

**Figure 1–28 Edit Project Dialog**

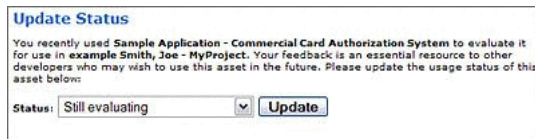


This figure illustrates the Edit Project dialog.

\*\*\*\*\*

4. Click **Save**. If the project has consumed assets, the Update Usage Status dialog is displayed, as shown in Figure 1–29. The consumed assets are listed in this dialog, each with a Status list.

**Figure 1–29 Update Status**



This figure illustrates the Update Status dialog.

\*\*\*\*\*

5. Update the status of each consumed asset as necessary.

---

**Note:** Depending on the specific role and permissions assigned to the project leader, certain assets consumed by the project do not appear in the list. If this is the case, an ATTENTION message appears in the Update Usage Status dialog. Be advised: Only the project members listed below the ATTENTION message can reject the asset. Otherwise, clicking **OK** in the Update Usage Status dialog sets the status of any hidden assets to Deployed.

---

6. When the Update Usage Status pop-closes the view returns to the Edit Project dialog.
7. Click **Save**. The project is closed.

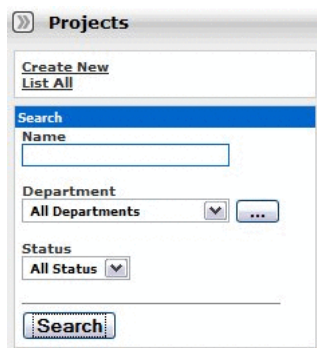
### 1.6.7 Adding a Relationship to a Project

The Project Relationships feature enables you to map the relationships between projects on a parent/child basis.

This procedure is performed on the Projects page.

1. Click **List All** on the Projects page, as shown in Figure 1–30.

**Figure 1–30** *Projects Page*



This figure illustrates the Projects page.

\*\*\*\*\*

A list of projects appears in the top section of the main pane, as shown in Figure 1–31.

**Figure 1–31** *List of Projects*

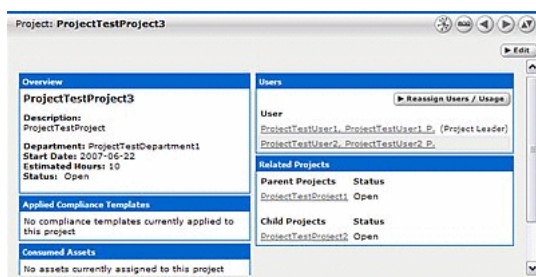
Results (3)		
Project	Department	Status
Example Smith, Joe - MyProject	Development	Open
Registry	Development	Open
Sample Project	Development	Open

This figure illustrates the list of projects.

\*\*\*\*\*

2. Select the project to be edited. The Project Detail pane is displayed in the bottom section of the main pane, as shown in Figure 1–32.

**Figure 1–32** *Project Detail Pane*

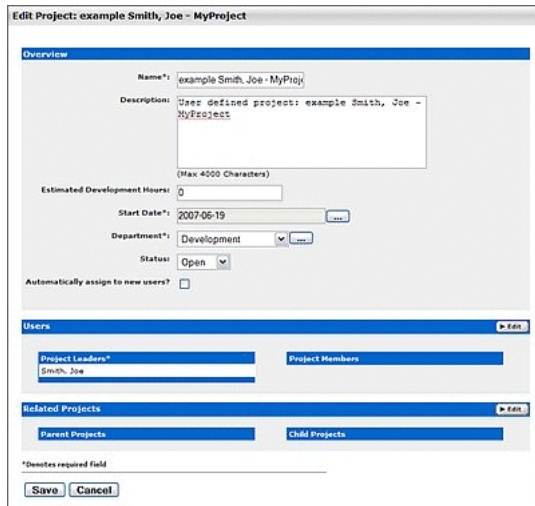


This figure illustrates the project details pane.

\*\*\*\*\*

3. Click **Edit** in the Project Detail pane. The Edit Project dialog is displayed, as shown in Figure 1–33.

**Figure 1–33 Edit Project Dialog**

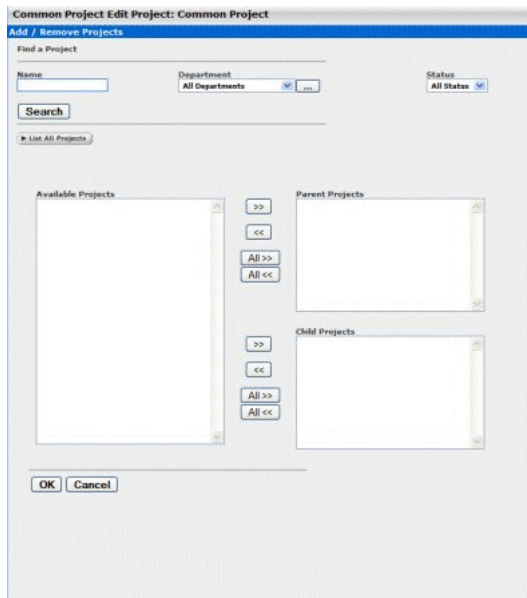


This figure illustrates the Edit Project dialog.

\*\*\*\*\*

4. Click **Edit** in the Related Projects section of the Edit Projects dialog. The Add/Remove Projects dialog is displayed, as shown in Figure 1–34.

**Figure 1–34 Add / Remove Projects**



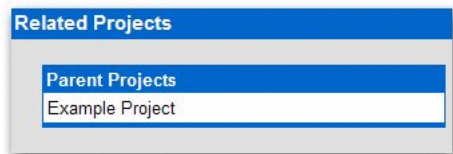
This figure illustrates the Add/Remove projects dialog.

\*\*\*\*\*

5. Click **List All Projects**. A list appears in the Available Projects window.
6. Select the project to be added and use the arrow buttons to move it to either the Parent Projects or the Child Projects category window, as appropriate. The project appears in the selected window.

7. Click **OK** to close the Add/Remove Projects dialog. The new project relationship appears in the appropriate list in the Related Projects section of the Edit Project dialog, as shown in Figure 1–35.

**Figure 1–35 Related Projects Section**

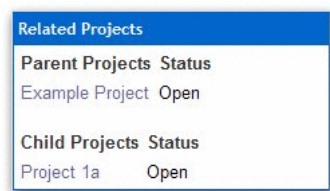


This figure illustrates the Related Projects section.

\*\*\*\*\*

8. Click **OK** to close the Edit Project dialog. The new project relationship appears in the Related Projects section of the Asset Detail, as shown in Figure 1–36.

**Figure 1–36 Related Projects Section**



This figure illustrates the Related Projects section.

\*\*\*\*\*

Project relationship are also displayed in the Oracle Enterprise Repository Navigator.

## 1.6.8 Deleting a Project

To delete a project in Oracle Enterprise Repository, perform the following steps from the Oracle Enterprise Repository Projects page:

1. Click **List All**, or search for the project that you want to delete.
2. To delete a project:
  - To delete projects from the Search Results screen, select the check box next to each project that you want to delete, and then click **Delete**. You can delete multiple projects simultaneously using this method.
  - To delete a project from its detail page, open the detail page for the project you want to delete, and then click **Delete**.

## 1.7 Sessions

Sessions allow the administrator to see which users are logged in to Oracle Enterprise Repository and, if necessary, shut down a user session.

- [Section 1.7.1, "Viewing Sessions"](#)
- [Section 1.7.2, "Deleting a Single Session"](#)

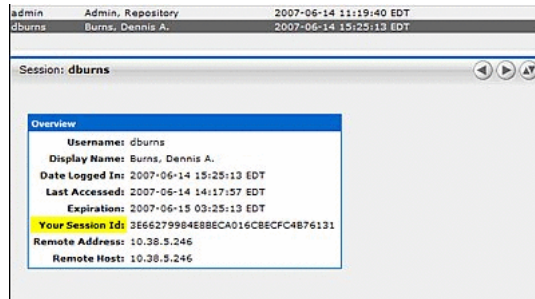
- [Section 1.7.3, "Deleting Multiple Sessions"](#)

## 1.7.1 Viewing Sessions

This procedure is performed on the Oracle Enterprise Repository Admin screen.

1. In the left panel, click **Sessions**. A list of sessions is displayed in the main pane, as shown in Figure 1–37.

**Figure 1–37 Sessions Pane**



This figure illustrates the Sessions pane.

\*\*\*\*\*

## 1.7.2 Deleting a Single Session

1. Select the session to be deleted from the list in the main pane. The session's detail opens in the bottom frame.
2. Click the **Delete** button in the session detail. A confirmation dialog is displayed.
3. Click **OK**. The session is deleted.

## 1.7.3 Deleting Multiple Sessions

1. In the list of sessions in the main pane, select each session to be deleted.

**Tip:** Place a check mark in the check box at the top of the column to select all listed sessions.

2. Click the **Delete** button in the table header. A confirmation dialog is displayed.
3. Click **OK**. The session is deleted.

(Requires the Delete permission. See Section 1.4, "Access Settings".)

## 1.8 Types

A Type defines the makeup of an asset. Every asset needs a type that defines its attributes. You can also generate the XML of a Type. This is specifically used for the Migration process, which requires mapping attributes of one Type to another Type. See Chapter 8, "Using the Type Manager" for more information.

## 1.9 Accessing Oracle Enterprise Repository Diagnostics Page

In Oracle Enterprise Repository, the Diagnostics page is disabled, by default. Navigate to `http://host_name:port/application_name/diag/index.jsp` (replace `host_name` with the appropriate location).

When you open the Diagnostics page in the default mode, the following message is displayed:

*Diag pages are currently disabled. Please contact your Oracle Enterprise Repository Administrator.*

To enable the Diagnostics page, perform the following steps:

1. Navigate to `ORACLE_HOME/user_projects/domains/DOMAIN_NAME/bin/`.
2. Edit `setStartupEnv.sh` or `setStartupEnv.cmd` before starting the Oracle Enterprise Repository server.
3. Add `-DdiagPagesEnabled=true` to the `EXTRA_JAVA_PROPERTIES="{EXTRA_JAVA_PROPERTIES}"` line in this file.
4. Restart the Oracle Enterprise Repository server.

It is recommended to only enable when necessary and disable once the system is running without any issues.

## 1.10 Install Java Web Start

Java Web Start is a browser plug-in that runs files with JNLP extensions. Java Web Start must be available on the client to use Oracle Enterprise Repository's Asset Editor, Type Manager, Navigator Applet, Browse Tree applet and the Import/Export utility.

- [Section 1.10.1, "Download and Install Java Web Start"](#)
- [Section 1.10.2, "Troubleshooting Java Web Start"](#)
- [Section 1.10.3, "Configure Java Web Start on Linux"](#)

### 1.10.1 Download and Install Java Web Start

Java Web Start is included in the latest version of the JDK.

At the time this document was published, Oracle Fusion Middleware products required JDK 7.0 Update 55 or later. However, you should always verify the required JDK version by reviewing the certification information, as described in To download the required JDK, use your browser to navigate to the following URL and download the Java SE JDK:

<http://www.oracle.com/technetwork/java/javase/downloads/index.html>

### 1.10.2 Troubleshooting Java Web Start

When browsers encounter a file with an unknown extension, they often prompt you to Open or Save when opening an application. Choose the **Open** option and browse to the `$JAVA_HOME/jre/bin/javaws` file, and select the appropriate option, to perform this action for the JNLP file type. If a new JRE is required, then follow the instructions in [Section 1.10.1, "Download and Install Java Web Start"](#) to download the latest JRE.

### 1.10.3 Configure Java Web Start on Linux

You must configure Java WebStart before you can use it on a Linux operating system.

To configure Java WebStart:

---

---

**Note:** For Linux, you only have a few options in browsers, and the default browser for the Linux installs is Mozilla Firefox.

---

---

1. Close Mozilla Firefox.
2. Locate the mozilla directory on your computer, for example, `/usr/lib/mozilla-1.4` or `/home/<user>/.mozilla`.
3. Open a command prompt and go to the mozilla directory on your computer, using the following command, for example:  

```
cd home/<user>/.mozilla
```
4. Create the plugins directory, if it does not exist, using the following command:  

```
mkdir plugins
```
5. Go to the **plugins** sub-directory under the Mozilla installation directory, as follows:  

```
cd plugins
```
6. In the current directory, create a symbolic link to the `Java ns7/libjavaplugin_oji.so` file type:  

```
ln -s /oracle/soa/mwhome_gov/jdk7/jre/plugin/i386/ns7/libjavaplugin_oji.so
```
7. Open Mozilla Firefox and test the following link:  
<http://www.java.com/en/download/help/testvm.xml>
8. In Mozilla Firefox, click the **Edit** menu, and then select **Preferences**. The Preferences dialog is displayed.
9. Click **Applications**. The Application page is displayed.
10. In the Search field, enter `JNLP`. The JNLP is displayed.
11. Select the displayed JNLP. The Action list gets activated.
12. Click the Action list to point to the correct javaws installed (such as the one installed with WLS JDK's).

---

---

**Note:** However, on 64 bit Linux, WLS comes with JRockit only. You have to manually download the 64 bit JDK to access the plug-in.

---

---

---

---

**Note:** You can also run a Java Web Start application by saving the application locally, and then using the javaws executable, located in `<JAVA_HOME>/jre/bin`. For example:

```
<JAVA_HOME>/jre/bin/javaws -verbose /path_to_file/impexp.jnlp
```

---

---



## System Settings Overview

This chapter describes Oracle Enterprise Repository's System Settings that administrators use to configure basic operations and specific features.

This chapter contains the following sections:

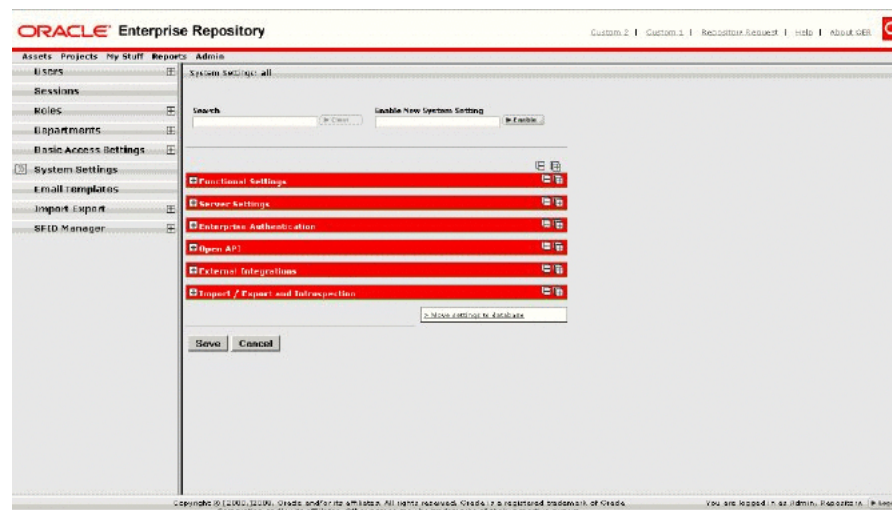
- [Section 2.1, "Access System Settings"](#)
- [Section 2.2, "Search for System Settings"](#)
- [Section 2.3, "Enable System Settings Properties"](#)
- [Section 2.4, "Move Settings to Database"](#)
- [Section 2.5, "System Settings in Oracle Enterprise Repository"](#)
- [Section 2.6, "Known Issues"](#)

### 2.1 Access System Settings

To access the system settings, perform the following steps:

1. [Click System Settings](#) in the sidebar on the Oracle Enterprise Repository Admin screen.

**Figure 2–1** Oracle Enterprise Repository System Settings Page



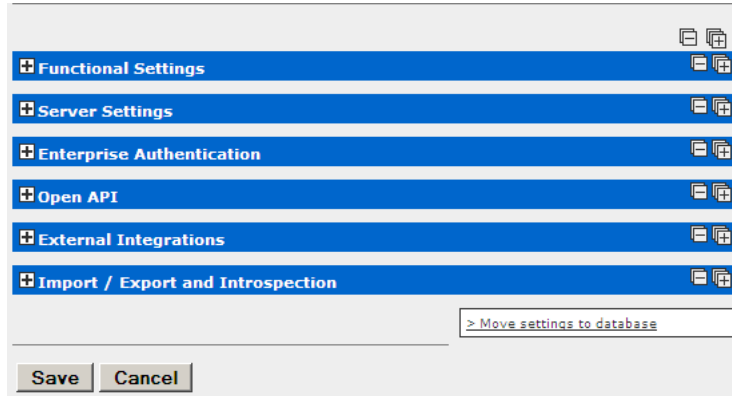
This figure illustrates the Oracle Enterprise Repository system setting page.

\*\*\*\*\*

The System Settings section opens in the main pane.

System Settings are organized into categories. Each category is further divided into feature groups.

**Figure 2–2 Functional Settings Collapsed**



This figure illustrates the Functional Settings section.

\*\*\*\*\*

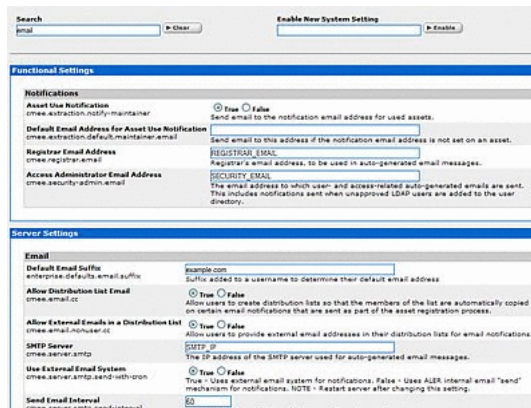
2. Click the plus icon to expand the display of any category or feature group.
3. Click the minus icon to collapse the display.

## 2.2 Search for System Settings

System settings for particular features can be located using the System Settings Search.

1. Enter a search term in the **Search** field. The relevant category sections expand to display the relevant System Settings.

**Figure 2–3 Functional Settings Expanded**



This figure illustrates the Functional Settings section.

\*\*\*\*\*

2. Click **Clear** to clear the search results.

## 2.3 Enable System Settings Properties

In some cases it may be necessary to enable specific properties in order to activate certain optional Oracle Enterprise Repository features. These properties are provided by Oracle Support.

1. Obtain the necessary system setting property.
2. Enter the system setting property in the **Enable New System Setting** field.
3. Click **Enable**. The appropriate System Setting category will open in the lower pane.
4. Edit the system setting properties as necessary.
5. Click **Save** when finished.

## 2.4 Move Settings to Database

---



---

**Note:** Exercising this option moves all system settings stored in property files to the database. This action cannot be reversed.

---



---

Click **Move settings to database**. All system settings are saved to the database. This is for a clustered environment. All servers in the clustered environment will now have one source for system settings.

## 2.5 System Settings in Oracle Enterprise Repository

This section describes all the system setting available in Oracle Enterprise Repository.

This section contains the following topics:

- [Section 2.5.1, "Functional Settings"](#)
- [Section 2.5.2, "Server Settings"](#)
- [Section 2.5.3, "Enterprise Authentication Settings"](#)
- [Section 2.5.4, "Open API Settings"](#)
- [Section 2.5.5, "External Integration Settings"](#)
- [Section 2.5.6, "Import / Export and Introspection Settings"](#)

### 2.5.1 Functional Settings

This section contains the following topics:

- [Section 2.5.1.1, "General User Interface"](#)
- [Section 2.5.1.2, "Search"](#)
- [Section 2.5.1.3, "Print Asset Detail via PDF"](#)
- [Section 2.5.1.4, "Projects and Navigator"](#)
- [Section 2.5.1.5, "Assets in Progress"](#)
- [Section 2.5.1.6, "Policy Management"](#)
- [Section 2.5.1.7, "Notifications"](#)
- [Section 2.5.1.8, "Access Control"](#)

- Section 2.5.1.9, "Software File Identification (SFID)"
- Section 2.5.1.10, "Use - Download"
- Section 2.5.1.11, "Custom Links"
- Section 2.5.1.12, "Reports"
- Section 2.5.1.13, "Miscellaneous"

### 2.5.1.1 General User Interface

#### Properties

- Full Name Format
  - enterprise.defaults.fullname.format
  - Specifies the format for the display of user names. Can contain the strings "{prefix}", "{first}", "{middle}", "{suffix}" and punctuation immediately following any of these name parts in any order. Extra space is trimmed.

---

**Note:** Changes to this setting require server restart.

---

- Display Name Field
  - enterprise.defaults.displayname.field
  - The list selection determines the information displayed for the user. The available options display the user's email address (emailaddress option), the user's full name (fullname option), the user's username (username option), the user's full name and username (fullname (username) option), or the user's full name and user id (fullname (userid) option).

---

**Note:** Changes to this setting require server restart.

---

- Display Name - Full Name and Initial
  - enterprise.defaults.fullname.middleasinitial
  - When a user's middle name is supplied, the True and False options toggle the display of the middle name in full or as an initial.

True results in the format shown in Figure 2–4.

**Figure 2–4 Full Name and Full Middle Name**

User Name	Display Name	Status
jsmith	Smith, John James	Active

This figure illustrates the Full Name and Initial screen. The user's name is displayed as Smith, John James. The user's full middle name is displayed.

\*\*\*\*\*

False results in the format shown in Figure 2–5.

**Figure 2-5 Full Name and Initial**

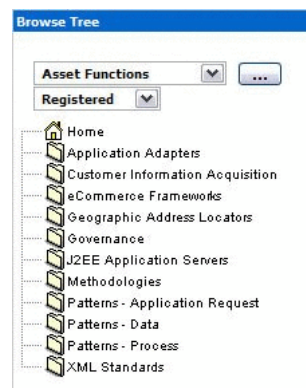
User Name	Display Name	Status
jsmith	Smith, John J.	Active

This figure illustrates the Full Name and Initial screen. The user's name is displayed as Smith, John J. The user's middle name is displayed as the initial J.

\*\*\*\*\*

- **Default Browse Categorization**
  - `cmee.assettab.default-categorization`
  - The list selection determines the initial asset categorization in the Browse Tree on the Assets screen. The available categorizations are Asset Function, Asset Lifecycle Stage, Business Objective, Classification, Domain, Enterprise Architecture Model, Line of Business, Source, Technology, and Tooling.

The results of selecting Asset Function as the Default Browse Categorization is displayed in Figure 2-6.

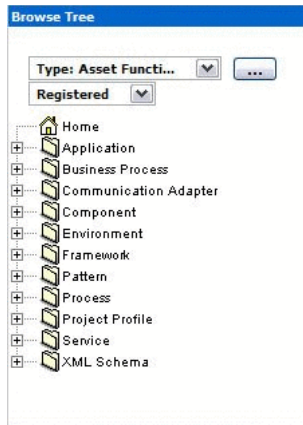
**Figure 2-6 Browse Tree, Asset Function Categorization Selected**

This figure illustrates the Browse tree. Asset Functions is selected in the Categorizations list.

\*\*\*\*\*

- **Show Browse Asset Tree**
  - `cmee.assettab.browse.visible`
  - Toggle to True to make the Browse Asset Tree visible, sortable by asset categorization. Toggle to False to disable the Browse Asset Tree.
- **Default Browse Type**
  - `cmee.assettab.default-categorization-show-assettype`
  - The True and False options toggle the default display of assets by Type in the Browse Tree on the Assets screen (subject to the system setting in Default Browse Categorization).

Figure 2-7 displays the results of toggling this property to True.

**Figure 2–7 Browse Tree, Assets Displayed by Type**

This figure illustrates the Browse tree. The assets are displayed by type.

\*\*\*\*\*

- **Maximum Size for Field in List Pane**
  - `cme.asset.listpane.maxsize`
  - The maximum size for any field in the list pane. Fields which are longer than this size will be truncated and a tooltip will be displayed with the full text when the user mouses over the field.
- **Sticky Registrar Queues**
  - `cme.registrar.stickyqueues`
  - The True and False options toggle the default display of the selected Asset Editor tab when browsing assets.
    - \* True: Each asset opens to the Asset Editor tab in view from the previously selected asset.
    - \* False: Defaults to the Overview tab for each asset viewed in the Asset Editor.
- **Table Zoom**
  - `cme.assettab.multiplegroupelement.force-zoom`
  - The True and False options toggle the ability to open table elements in their own window.
- **Repeat Table Header**
  - `cme.assettab.multiplegroupelement.show-header-interval`
  - The number entered in the text box determines the interval (number of rows) at which the table header repeats in large tables.
    - \* Figure 2–8 displays the result of assigning this property a value of 0.

**Figure 2–8 Cost Categories, No Repeating Headers**

Cost Categories		
Total Projected Fiscal Year Costs: 302610		
Actual Costs (YTD): 250680		
Annual Costs (Detail):		
Cost Category	Projected	Actual
Development	55450	61230
Operations	98600	98600
Maintenance	98560	90850
Support	50000	0
Lifecycle Investment to Date (Total Cost of Ownership): 970500		
Cost Per Transaction (\$): 0.27		
Chargeback Costs: 100% of cumulative transaction costs		

This figure illustrates the Cost Categories. The Cost Category heading does not repeat as the Repeat Table Header property was set to 0.

\*\*\*\*\*

- \* Figure 2–9 displays the result of assigning this property a value of 2.

**Figure 2–9 Cost Categories, Repeating Table Header**

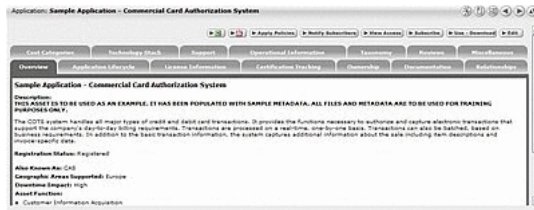
Cost Categories		
Total Projected Fiscal Year Costs: 325000		
Actual Costs (YTD): 334700		
Annual Costs (Detail):		
Cost Category	Projected	Actual
Development	75000	81200
Operations	100000	104900
Cost Category	Projected	Actual
Maintenance	100000	98600
Support	50000	50000
Lifecycle Investment to Date (Total Cost of Ownership): 1250000		
Cost Per Transaction (\$): 0.31		
Chargeback Costs: 100% of cumulative transaction costs		

This figure illustrates the Cost Categories. The Cost Category repeats every two rows as the Repeat Table Header property was set to 2.

\*\*\*\*\*

- Collapse Relationships
  - `cmee.assettab.relationships.collapse`
  - Toggle to True to collapse the list of Related Assets on Asset Detail pages. Expanding the list requires the user to click on the **Zoom** button to the left of Relationship name.
- Show All Detail In Tabs
  - `cmee.framework.show-as-tabs`
  - The True and False options toggle the default display of the asset detail.
  - \* Figure 2–10 displays the results of setting this property to True.

**Figure 2–10 Asset Detail Page, Show All Detail in Tabs set to True**

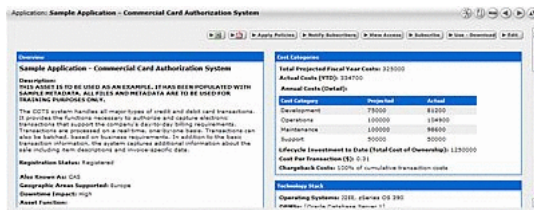


This figure illustrates an asset detail page with the Show All Detail in Tabs property enabled.

\*\*\*\*\*

\* Figure 2–11 displays the results of setting this property to False.

**Figure 2–11 Asset Detail Page, Show All Detail in Tabs set to False**



This figure illustrates an asset detail page with the Show All Detail in Tabs property disabled.

\*\*\*\*\*

- **Long Date Format Specification**
  - `cmee.server.date.format.long.default`
  - Specifies the date format for the display of full date information (yyyy-MM-dd HH:mm:ss z).
- **Short Date Format Specification**
  - `cmee.server.date.format.short.default`
  - Specifies the date format for the display of abbreviated date information (yyyy-MM-dd).
  - \* Similar to Long Date Format Specification.
- **Show System-Supplied Relationships**
  - `cmee.show-system-supplied-relationships`
  - The True and False options toggle the display of system-supplied relationships in the Asset Editor tab and in the asset detail. The False setting will hide system-supplied relationships in the Asset Editor's Relationships tab and in the asset detail; however, they will remain visible in the Asset Editor's Configure Relationships dialog.

System-supplied relationships are used for system-specific relationships within Oracle Enterprise Repository, such as allowing for more complex matches when doing asset updates. Therefore, users cannot create or delete system-supplied relationships, nor can they modify a system-supplied relationship's direction, which is always read-only.



- Hide inactive asset types in the web tool
  - `oracle.gui.assettype.hide.inactive`
  - Toggle to True to hide inactive asset types from search results.

### 2.5.1.2 Search

#### Properties

- Maximum number of results for Asset Search
  - `cmee.search.assets.maxresults`
  - The numerical value entered in the text box determines the maximum number of assets returned by a search. Entering 0 returns an unrestricted number.
- Save Searches
  - `cmee.asset.registry.saved.searches`
  - The True and False options toggle the Save Searches feature. This feature allows search criteria to be saved and reused.

True displays the Saved Searches section in the sidebar on the Assets screen, as illustrated below.

**Figure 2–12** *Saved Searches Enabled*



This figure illustrates the Saved Searches screen when the Save Searches property is set to True.

\*\*\*\*\*

- Search Indexed Fields
  - `cmee.search.general.include.indexedfields`
  - The True and False options toggle the inclusion of indexed fields in general asset searches.
- Search Categorizations
  - `cmee.search.general.include.categorizations`
  - The True and False options toggle the inclusion of categorizations in general asset searches.
- More Search Options Categorization Default Visibility
  - `cmee.search.advanced.categorizations.visible`
  - The True and False options toggle the default display of the Categorizations tree in the expanded form and included in the More Search Options dialog.

Figure 2–13 displays the results of setting this property to False.

**Figure 2–13 More Search Options - Categorizations Tree Not Expanded**

This figure illustrates a search with the More Search Options Categorization Default Visibility property set to False.

\*\*\*\*\*

Figure 2–14 displays the results of setting this property to True.

**Figure 2–14 More Search Options - Categorizations Tree Expanded**

This figure illustrates a search with the More Search Options Categorization Default Visibility property set to True.

\*\*\*\*\*

- Search Specific Field

- `cmee.search.specific.field`
- The True and False options toggle the ability to search specific fields during Standard Searches by identifying the field to be searched in the Keyword text box.

As illustrated below, the string `description:MyAsset` will search for the keyword the `MyAsset` in the Description field of every asset in the Repository.

**Figure 2–15 Search a Specific Field**

The screenshot shows a search interface with the following elements:

- Search** (Title)
- Enter Search String** [?] (Label)
- description: MyAsset** (Text input)
- Type** (Label)
- All Types** (Dropdown menu)
- Asset Function** (Label)
- All Asset Functions** (Dropdown menu)
- More Search Options** (Text)
- Search** (Button)

This figure illustrates the Search pane. The user is searching the Description Field using `MyAsset` as the search term.

\*\*\*\*\*

- **Confirm Empty Search**
  - `cmee.empty.search.confirm`
  - The True and False options toggle the display of a warning message when a user performs a search on a blank string.
    - True displays a warning message.
- **Log Searches**
  - `cmee.log.searches`
  - The True and False options toggle the logging of searches. Only search criteria for keyword searches will be logged.
- **Log Complete Searches**
  - `cmee.log.complete.searches`
  - The True and False options toggle the logging of complete search criteria. All search criteria will be logged.
- **Minimum Number of Characters for User Search**
  - `cmee.users.min.text.search`
  - The numerical value entered in the text box determines the minimum number of characters required to perform a search to locate user information.
    - Attempts to perform a user search with fewer than the minimum number of characters results in a warning message.
- **Assembly Model Visibility**
  - `cmee.assembly.model.visibility`
  - Toggle this property to True to enable the ability to filter out Assembly Models.

### 2.5.1.3 Print Asset Detail via PDF

#### Properties

- Use default XSLT for asset types
  - `cmee.asset.pdf.xslt.default`
  - The True and False options specifies that when exporting an asset to PDF, a default asset type is used. Failing its existence, an error is shown.
- PDF Asset Export Destination
  - `cmee.asset.xml.paths.export-destination`
  - Text entered in the text box identifies the PDF asset export destination.
- PDF Asset Export Destination URL
  - `cmee.asset.xml.paths.export-destination-url`
  - Text entered in the text box identifies the PDF asset export destination URL.
- PDF Asset XSL Directory
  - `cmee.asset.xml.paths.xsl-source`
  - Text entered in the text box identifies the PDF asset XSL directory.
- PDF Asset Generated PDF Destination
  - `cmee.asset.xml.paths.out-destination`
  - Text entered in the text box identifies the PDF asset generated PDF destination.
- PDF Asset Generated PDF Destination URL
  - `cmee.asset.xml.paths.out-destination-url`
  - Text entered in the text box identifies the PDF asset generated PDF destination URL.
- PDF Asset Export WebApp URL
  - `cmee.asset.xml.paths.webapp-url`
  - Text entered in the text box identifies the PDF asset export WebApp URL.

### 2.5.1.4 Projects and Navigator

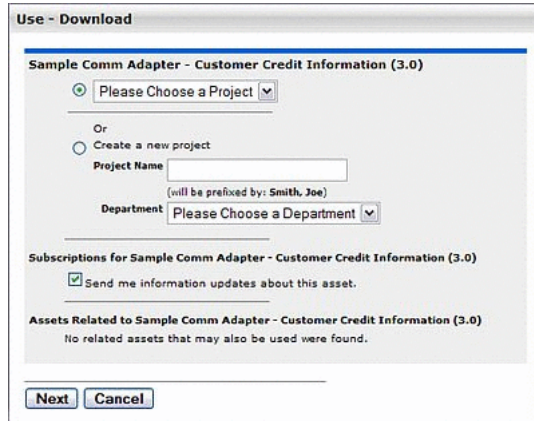
#### Properties

- Enable Projects
  - `cmee.projects.enabled`
  - The True and False options toggle the support of multiple projects.
- Navigator and Project Hierarchy
  - `cmee.projecttool.advanced`
  - The True and False options toggle advanced project options including the Navigator and project hierarchy.
- Navigator Support for Microsoft JDK
  - `cmee.navigator.force-msjdk-support`

- The True and False options toggle the use of an alternative Navigator applet that is compatible with Microsoft's applet plugin for Internet Explorer.
- **User Defined Projects**
  - `cmee.extraction.user-projects.enable`
  - The True and False options toggle the feature that allows users to create a project on the fly during the asset Use/Download process.

Figure 2–16 displays the results of setting this property to True.

**Figure 2–16 Use - Download with User Defined Projects Set to True**

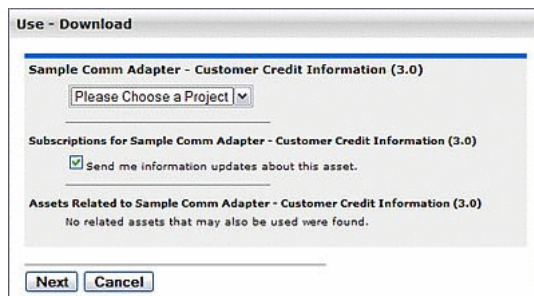


This figure illustrates the Use - Download dialog with the User Defined Projects property set to True. The user has the option to create a new project and to select a department for the newly created project.

\*\*\*\*\*

Figure 2–17 displays the results of setting this property to False.

**Figure 2–17 Use - Download with User Defined Projects Set to False**



This figure illustrates the Use - Download dialog with the User Defined Projects property set to False.

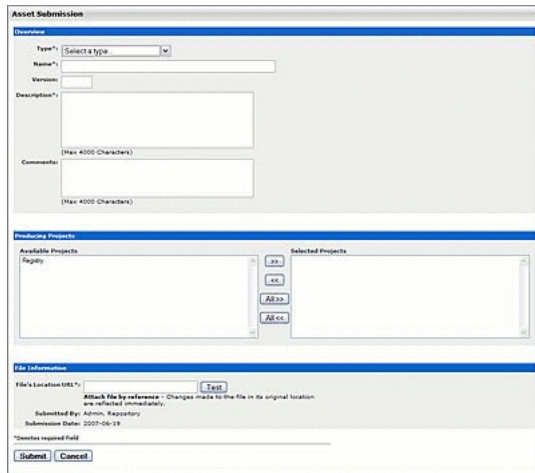
\*\*\*\*\*

- **User Defined Projects - Prefix**
  - `cmee.extraction.user-projects.prefix`
  - Text entered in the property text box becomes the prefix for any user-defined projects. This is useful for visual grouping in reports. (User-defined Projects must be enabled.)

- Producing Projects
  - `cmee.project.producing.enabled`
  - The True and False options toggle the display of the Producing Projects section in the Asset Submission dialog (Standard Submission via the Submit and Asset link on the Oracle Enterprise Repository Assets screen.)

Figure 2–18 displays the results of setting this property to True.

**Figure 2–18** *Asset Submission, Producing Projects Section Visible*

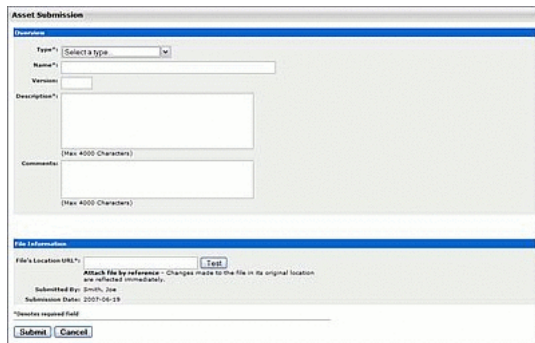


This figure illustrates the Asset Submission screen. The Producing Projects section is visible because the Producing Projects property is set to True.

\*\*\*\*\*

Figure 2–19 displays the results of setting this property to False.

**Figure 2–19** *Asset Submission, Producing Projects Section Hidden*



This figure illustrates the Asset Submission screen. The Producing projects section is hidden because the Producing Projects property is set to False.

\*\*\*\*\*

- Advanced Project Option - Categorization
  - `cmee.projecttool.categorizations`
  - The True and False options toggle the assignment of categorizations to projects.

- Enable Project Reported Asset Value
  - `cmee.projecttool.assetvalue.enabled`
  - The True and False options toggle the metrics of the Reported Asset Value for projects.

### 2.5.1.5 Assets in Progress

#### Properties

- View Assets in Progress
  - `cmee.asset.in-progress.visible`
  - Select True to enable the ability to see assets in progress outside the Asset Editor.

### 2.5.1.6 Policy Management

#### Properties

- Enable Compliance Templates
  - `cmee.compliance-template.enable`
  - Enables the Compliance Template System.
- Enable Asset Policies
  - `cmee.asset-policy.enable`
  - Enables the Policy Management System.
- Collapse Policy Assertions
  - `cmee.asset-policy.collapse.assertions`
  - Collapse the display of assertions by default when showing applied policies in the Asset detail.
- Asset Policy Relationship
  - `cmee.asset-policy.relationship`
  - Defines the relationship type within the system to use for policy assignment - it should be a Two-way, Order Matters relationship.

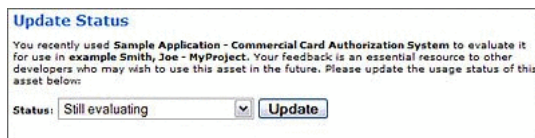
### 2.5.1.7 Notifications

#### Properties

- Asset Use Notification
  - `cmee.extraction.notify-maintainer`
  - The True and False options toggle the feature that automatically sends an email to the notification address listed for an asset when that asset is used.
  - When an email is sent, the following algorithm is used to determine the destination address:
    - \* If the asset has a notification address, use that as the email destination,
    - \* else, if the system setting `cmee.extraction.default.maintainer.email` has a value, use that value as the destination address,

- \* else, if there is a "submitted-by" user that has an email address, use that email address,
  - \* else, if there is a "created-by" user that has an email address, use that email address,
  - \* else, do not send a usage notification email.
- **Default Email Address for Asset Use Notification**
    - `cmee.extraction.default.maintainer.email`
    - This system setting is used to specify the default destination email address for notifications related to usage sent by the system.
  - **System Registrar Email**
    - `cmee.registrar.email`
    - Fill in the text box with the registrar's email address, to be used in auto-generated email messages.
  - **System Registrar Alternate Notification**
    - `cmee.registrar.email.enable`
    - Allows notification emails to be sent to the System Registrar email address when the notification email has not been set on an asset.
  - **Access Administrator Email Address**
    - `cmee.security-admin.email`
    - Fill in the text box with the email address of the individual(s) to whom user- and access-related auto-generated emails are to be sent. This includes notifications sent when unapproved LDAP users are added to the user directory.
  - **Enable Reminder Email**
    - `cmee.extraction.notification.enabled`
    - Enables notification via email to update the status of asset usage and value metrics.
  - **Reminder Delay**
    - `cmee.extraction.initial.delay`
    - Fill in the text box with a numeric value indicating the number of days to wait after a user downloads an asset before sending a reminder to complete the Update Status form. See Figure 2–20.

**Figure 2–20 Update Status**



This figure illustrates the Update Status dialog. Still Evaluating is selected in the Status list.

\*\*\*\*\*

- **Reminder Delay - Subsequent**



- `cmee.extraction.subsequent.delay`
- Fill in the text box with a numeric value indicating the number of days to wait after the initial usage status reminder message before sending subsequent reminders.

### 2.5.1.8 Access Control

#### Properties

- Custom Access Settings
  - `cmee.customaccesssettings.enabled`
  - Enables advanced custom access settings.
- Custom Access Settings on Assets
  - `cmee.customaccesssettings.asset`
  - Enables advanced custom access settings on assets.
- Custom Access Settings on Files
  - `cmee.customaccesssettings.file`
  - Enables custom access settings on files.
- Ability to toggle all permissions
  - `cmee.security.gui.toggleall`
  - Enables the GUI control to allow an administrator to toggle all permissions one role at a time.

### 2.5.1.9 Software File Identification (SFID)

#### Properties

- Enable Automated Usage Detection
  - `cmee.usagedetection.enable`
  - Enables the Automated Usage Detection plug-in to manage SFIDs, and allow automated detection of possible usage.
- Enable Harvesting of Unrecognized SFIDs
  - `cmee.sfid.harvesting.enable`
  - Enables the collection of information related to detected SFIDs that are not yet registered in the system.

### 2.5.1.10 Use - Download

#### Properties

- Asset Usage Custom File
  - `cmee.extraction.export.filename`
  - Enter a filename to save custom asset metadata fields from the asset download pane.
- Asset Usage Custom Field List
  - `cmee.extraction.exported-custom-field`.

- This unique setting allows the creation of Custom Asset Usage Fields.

---

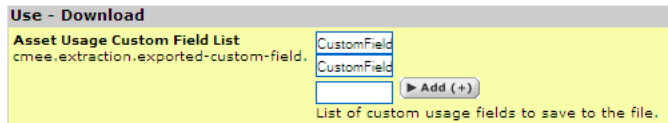
**Note:** This property appears with a period at the end. When you enter this property in the Search field, enter the property along with the period.

---

### Create Custom Usage Fields

1. Enter the property name in the Enable New System Setting box and press Enter.
2. As necessary, click **Add** to open additional text boxes.

**Figure 2–21 Use - Download**

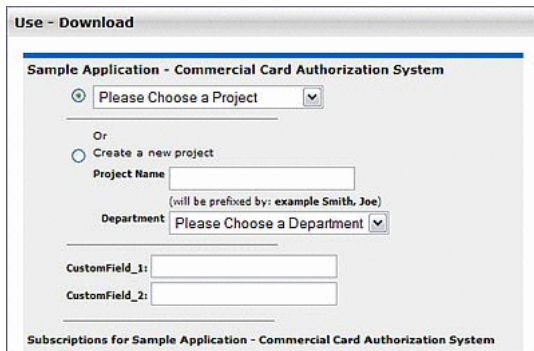


This figure illustrates the Use - Download dialog.

\*\*\*\*\*

3. Enter a name for each additional Custom Usage Field in the corresponding text box.
4. Click Save when finished. The new Custom Usage Fields appear in the Use - Download dialog.

**Figure 2–22 Use - Download with Custom Fields**



This figure illustrates the Use - Download dialog. Two custom fields have been appended to the dialog.

\*\*\*\*\*

### 2.5.1.11 Custom Links

This group of settings controls the configuration and display of custom links on the right side of the Oracle Enterprise Repository menu bar.

#### Properties

- Custom Link 1
  - cmee.tool.show-marketplace
  - The True and False options toggle the display of Custom Link 1.

**Figure 2–23 Custom Link**

This figure illustrates the Custom Link.

\*\*\*\*\*

**Figure 2–24 Custom Link 1 and Custom Link 2**

This figure illustrates the Custom Link 1 and Custom Link 2.

\*\*\*\*\*

(Illustrates both Custom 1 Link and Custom 2 Link set to True.)

- Custom Link 1 Label
  - `cmee.tool.marketplace.label`
  - Text entered in the text box will appear as the label for Custom Link 1 on the Oracle Enterprise Repository menu bar.
- Custom Link 1 URL
  - `cmee.tool.marketplace.url`
  - Enter the URL for Custom Link 1.
- Custom Link 1 Target Frame
  - `cmee.tool.marketplace.target`
  - Specifies the frame in which to open Custom Link 1
- Custom Link 2
  - `cmee.tool.show-marketplace`
  - The True and False options toggle the display of Custom Link 2.
- Custom Link 2 Label
  - `cmee.tool.marketplace.label`
  - Text entered in the text box will appear as the label for Custom Link 2 on the Oracle Enterprise Repository menu bar.
- Custom Link 2 URL
  - `cmee.tool.marketplace.url`
  - Enter the URL for Custom Link 2.
- Custom Link 2 Target Frame
  - `cmee.tool.marketplace.target`
  - Specifies the frame in which to open Custom Link 2.
- Custom Help URL
  - `cmee.tool.help.url`

- Specifies a custom URL for the Help link.

Note: Oracle Enterprise Repository help files are hosted remotely by default. If necessary, Help files can be hosted in your local environment. Change this setting to reflect the location of the Help files.

### 2.5.1.12 Reports

#### Properties

- Enable Reports
  - oracle.reports.enabled
  - Specifies whether reporting features are enabled.
- Report Server URL
  - oracle.reports.server.url
  - Specifies a custom URL for the report server link.
- Report Server Endpoint URL
  - oracle.reports.server.endpoint.url
  - Specifies a custom URL for the Web Service endpoint to authenticate against the report server.
- Report Server Username
  - oracle.reports.server.username
  - Specifies the username credential for the report server.
- Report Server Password
  - oracle.reports.server.password
  - Specifies the password credential for the report server.
- Enable Excel Output for Reports
  - oracle.reports.excel.enable
  - Enables Excel Output for Reports.

### 2.5.1.13 Miscellaneous

#### Properties

- Asset Submission File Requirement
  - cmee.submission.file.required.enable
  - When enabled, a user performing a Standard Asset Submission is required to provide a file location URL or upload files.
- Consumer Survey
  - cmee.extraction.consumer-survey.enable
  - The True and False options toggle the display of the complete asset usage Update Survey, available for any asset listed in the Asset Usage section of the My Stuff page.
    - True displays the full survey, as shown in Figure 2–25.

**Figure 2–25 Update Status, Consumer Survey Property Set to True**

This figure illustrates the Update Status dialog. The Customer Survey is displayed at the bottom of the dialog.

\*\*\*\*\*

False displays only the Status list, as shown in Figure 2–26.

**Figure 2–26 Update Status, Consumer Survey Property Set to False**

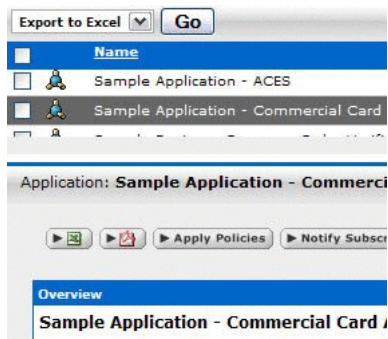
This figure illustrates the Update Status dialog.

\*\*\*\*\*

- **Default Usage Rejection Reason**
  - `cmee.default.usage.rejection.reason`
  - Use the list to select the default asset rejection reason that is recorded when an asset is rejected when updating the status of the asset usage in the My Stuff section. The available options are Does not meet requirements, Immature technology, Incompatibility, Insufficient documentation, Legal / Licensing, Performance, Project timing, Reliability, Research Only, Risk, Scalability, System resource requirements, and Usability.
- **Enable Type Manager**
  - `cmee.typemanager.enabled`
  - The True and False options toggle the ability to display Type Manager. True displays the Type Manager when you select the Manage Types option in the Asset Editor. False does not display the Type Manager.
- **Export Assets into a Microsoft Excel spreadsheet**
  - `cmee.asset.registry.export.excel`
  - The True and False options toggle the ability to export asset detail information to an external Microsoft Excel spreadsheet.

True displays the Export to Excel menu selection in the asset search results frame, as shown in Figure 2–27, and displays the Excel button in the asset detail.

**Figure 2–27 Export to Excel in Search Results**



This figure displays the Export to Excel menu selection in the search results frame.

\*\*\*\*\*

Setting this property to False removes the menu selection and button.

### Affinity Settings

Within the context of Oracle Enterprise Repository, Affinity is the condition that occurs when two or more assets are deployed in the same project. The Affinity element will appear in the asset detail for each asset that meets this condition, and will list other assets that have (deployed) projects in common.

The following Affinity System Settings determine the specific behavior of the Affinity element and its contents.

- Affinity Threshold
  - `cmee.assettab.affinity.threshold`
  - If the number of affinity matches exceeds the numerical value entered for this setting, a "View All Affinity Assets" link appears in the Affinity element in the asset detail.
- Affinity Minimum Project Count
  - `cmee.assettab.affinity.minimum`
  - The numerical value entered in this field determines the minimum number of projects to require inclusion in for assets to appear in Affinity section on Asset Detail page.

If the number of projects that have consumed the asset is less than the supplied numerical value, the Affinity element will indicate that the display criteria have not been met.

## 2.5.2 Server Settings

This section contains the following topics:

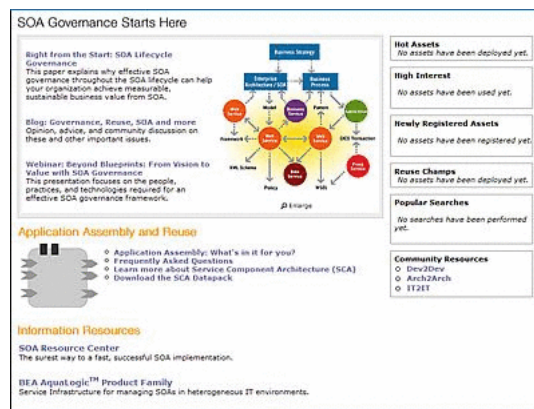
- [Section 2.5.2.1, "General"](#)
- [Section 2.5.2.2, "Server Resources"](#)
- [Section 2.5.2.3, "Email"](#)
- [Section 2.5.2.4, "Upload Area"](#)
- [Section 2.5.2.5, "Audit"](#)
- [Section 2.5.2.6, "Caching"](#)

### 2.5.2.1 General

- **Admin Role**
  - `enterprise.security.roletype.admin`
  - The text entered in this box designates the name applied to the role that has unlimited administrative rights and permissions within Oracle Enterprise Repository.
- **Asset Editor Enable Embedded HTML in Asset Description**
  - `cmee.asseteditor.enable-embedded-html-asset-description`
  - True allows embedded HTML to be entered into an asset's description.
- **Company Name**
  - `cmee.enterprisetab.homepageselection.enabled`
  - The True and False options toggle the ability to assign a specific homepage based on user role or department assignment.

True assigns a specific homepage based on user role or department. False displays the default home page.
- **Enable Homepage Selection**
  - `cmee.server.companyname`
  - Enter the name of your company in the text box.
- **Homepage**
  - `cmee.enterprisetab.homepage`
  - The default URL points to the default Oracle Enterprise Repository homepage, as it appears in the main (right) frame.

**Figure 2–28** Default Oracle Enterprise Repository Homepage



This displays the default Oracle Enterprise Repository Homepage.

\*\*\*\*\*

Change this URL as necessary to load a different page as the Oracle Enterprise Repository homepage.

- **Default Character Encoding**
  - `cmee.server.defaultencoding`

- The default character encoding to use in the system (defaults to the JVM default).
- Examples are UTF-8 and ISO-8859-1. UTF-16 is not supported.
- **Enable Custom Unique Element**
  - `cmee.unique-element.enable`
  - The True and False options toggle the ability to enable an additional plug-in for Types, which defines a unique constraint additional to the Name/Version pair that is used by default.
- **Unique Element Per Type**
  - `cmee.unique-element.per-assettype`
  - The True and False options toggle the ability to make the Unique Element plugin pertain to each Type individually.
- **Asset Editor Disable Auto-refresh Tree**
  - `cmee.asseteditor.disable-autorefresh-tree`
  - The True and False options toggle the ability to disable the auto-refresh of tree whenever user clicks Accept, Submit, or Register.

### 2.5.2.2 Server Resources

#### Properties

- **Database Connection Name**
  - `enterprise.dbaccess.connectionname`
  - The name assigned to the database connection.
- **WebApp JSP Path**
  - `enterprise.server.paths.jsp`
  - The filepath to the location on the application server where JSPs are stored.
- **WebApp Servlet Path**
  - `enterprise.server.paths.servlet`
  - The filepath to the location on the application server where product servlets are stored.
- **Web Resource Path**
  - `cmee.server.paths.resource`
  - The Web server path to the location where Web resources are stored.
- **Web Image Path**
  - `cmee.server.paths.image`
  - The filepath to the location on the Web server where product images are stored. This setting must include the `*images` subdirectory of the Web Resource Path.
- **Web JNLP Path**
  - `cmee.server.paths.jnlp-tool`



- The filepath to the location on the Web server where JNLP tool jars are stored. Due to Java Web Start compatibility issues, this path must not point to an SSL-enabled service.
- **WebApp JSP Path**
  - `cme.server.paths.jsp`
  - The filepath to the location on the application server where JSPs are stored.
- **WebApp Servlet Path**
  - `cme.server.paths.servlet`
  - The filepath to the location on the application server where product servlets are stored.
- **Asset Detail Path**
  - `cme.assettab.asset-detail-page`
  - The application server path used by integrations to load asset details.

### 2.5.2.3 Email

#### Properties

- **Allow Distribution List Email**
  - `cme.email.cc`
  - The True and False options toggle the availability of the email distribution list feature. This allows users to create distribution lists so that the list members are automatically copied on certain email notifications that are sent as part of the asset registration process.
- **Allow External Emails in a Distribution List**
  - `cme.email.nonuser.cc`
  - The True and False options determine the ability for a user to specify external email addresses in distribution lists. True allows external email addresses in distribution lists. False restricts distribution lists to only internal Oracle Enterprise Repository users.
- **SMTP Server**
  - `cme.server.smtp`
  - The hostname or IP address of the SMTP server used for auto-generated email messages.
- **Use External Email System**
  - `cme.server.smtp.send-with-cron`
  - True and False options determine the selection of the email mechanisms for automated Oracle Enterprise Repository messages. True enables the external mechanism. False enables Oracle Enterprise Repository's internal email send mechanism.

Note: Changes to this setting require server restart.
- **Send Email Interval**
  - `cme.server.smtp.send-interval`

- This setting determines the number of minutes between automatic Oracle Enterprise Repository email messages.
- **Send System Email as HTML**
  - `cmee.server.smtp.send.format-html`
  - The True and False options toggle the selection of HTML or plain-text formatting of automatic Oracle Enterprise Repository messages.
    - \* True enables HTML formatting.
    - \* False enables plain text formatting.
- **Send System Email with Reply-To Header**
  - `cmee.server.smtp.send.replyto-required`
  - The True and False options toggle the option to include a Reply-to header in automated Oracle Enterprise Repository messages. The Reply-to header includes the From address.
    - True enables the Reply-to header.
- **Notification Email Delimiter**
  - `cmee.email.delimiter`
  - Use the list to select the appropriate delimiter (comma or semi-colon) for multiple email addresses as determined by the default email client. (For example, Microsoft Outlook uses the semi-colon, while Outlook Express uses the comma).
- **Send Diagnostic Email to Oracle**
  - `cmee.server.smtp.senddiag`
  - The True and False options toggle an automatic monthly diagnostic email to Oracle, for internal informational purposes only.

#### 2.5.2.4 Upload Area

##### Properties

- **Submission Upload Directory**
  - `cmee.server.paths.upload`
  - Defines a locally referenced operating system path to upload files onto the application server directly via an HTTP file upload process. This property enables the following features:
    - \* Enables the Add Attachment Now option in the Asset Submission form. (Required if files are to be attached to assets during the submission process.)
    - \* Enables Uploaded Submission Files option in the Asset Editor.
- **Registrar Submission Upload Path**
  - `cmee.server.paths.upload-registrar`
  - Enables file browsing through the Asset Editor (Optional).
    - \* The path, when hosted by UNC, requires Windows OS or Samba/Shareity.

- \* Enables the appearance of the Browse button in Uploaded Submission Files in the Asset Editor.
- **Submission Upload Artifact Store**
  - `cmee.server.paths.upload-repository`
  - Sets a configured artifact store to allow direct public access to the submitted files. This property is used for one-step acceptance of submitted files in the Asset Editor. When this property is set:
    - \* Asset files that are available for use/download are visible.
    - \* The Accept button is visible in the uploaded submission files section of the Asset Editor.

### 2.5.2.5 Audit

#### Properties

- **Asset Save Export Filename**
  - `cmee.asset.logging.filenameep`
  - Text entered in the text box identifies the filename used to save asset data at every save (for archival and audit purposes).
- **Asset Save Export Fail on Error**
  - `cmee.asset.logging.failonerror`
  - Select the True option to enable the ability to abort the save of the audit if unable to write to the asset export filename for any reason.

### 2.5.2.6 Caching

#### Properties

- **Preload Asset Cache at Start-up**
  - `cmee.assets.preload.at-startup`
  - Select the True option to preload the asset cache on application server start-up. True preloads the asset cache.
- **Preload Asset Cache as Seperate Thread**
  - `cmee.assets.preload.as-seperate-thread`
  - Select the True option to spawn the preloaded asset cache as a separate thread on application server startup.

## 2.5.3 Enterprise Authentication Settings

This section contains the following topics:

- [Section 2.5.3.1, "General"](#)
- [Section 2.5.3.2, "Authentication Token"](#)
- [Section 2.5.3.3, "Password Settings"](#)
- [Section 2.5.3.4, "Plugin Login Settings"](#)
- [Section 2.5.3.5, "LDAP and Active Directory Settings"](#)

- [Section 2.5.3.6, "Single Sign-On \(SSO\) Settings"](#)
- [Section 2.5.3.7, "Container Login Settings"](#)

### 2.5.3.1 General

#### Properties

- Unapproved User Login
  - enterprise.security.unapproveduser.allowlogin
  - The True and False options toggle the ability for unapproved users to log in to the system. This is useful when using LDAP or container-managed authentication.
- Display Login Pop-up
  - enterprise.security.showloginpopup
  - The True and False options toggle the default display of the Oracle Enterprise Repository login dialog. The dialog identifies the users and displays login information.
- Maximum Failed Login Attempts
  - enterprise.security.loginattempt.max
  - The numerical entry in the text box determines the number of failed login attempts before user lock-out. A value of zero or less disables lock-out. Figure 2–29 displays the message that notifies a user that they have been locked out.

**Figure 2–29 Oracle Enterprise Repository Login Page--User Locked Out**



This figure illustrates the Oracle Enterprise Repository Login page. The User is Locked Out message indicated the user has exceeded the maximum amount of failed login attempts.

\*\*\*\*\*

- Concurrent Logins Per User
  - enterprise.security.concurrentuser.max
  - The numerical value entered in the text box determines the maximum number of concurrent logins each user is permitted. A value of less than or equal to zero permits infinite concurrent user logins.
- Grace Logins
  - enterprise.security.password.expiration.gracelogins
  - The numerical value entered in the text box determines the number of times a user is allowed to log in after his/her password has expired.
- Enable LDAP Login

- enterprise.authentication.ldap.enabled
- Enables the LDAP Login Module for authentication.
- **Enable Container Login**
  - enterprise.authentication.container.enabled
  - Enables the Container Login Module for authentication.
- **Sync Container Roles**
  - enterprise.authentication.container.synchroles.enabled
  - Select the True option to enable the syncing of container roles.
- **Enable Advanced Container Login**
  - enterprise.authentication.advancedcontainer.enabled
  - Select the True option to enable the Advanced Container Login Module for authentication.
- **Default Email Suffix**
  - enterprise.defaults.email.suffix
  - Suffix added to a username to determine their default email address.
- **New Unapproved User Notification**
  - cmee.new.unapproved.users.notify
  - The True and False options toggle the feature that automatically sends an email to the security administrator when a new user is created via external authentication, such as LDAP or a Java EE container.

### 2.5.3.2 Authentication Token

#### Properties

- **Authentication Token Expiration**
  - enterprise.defaults.authtokens.expiration.value
  - The numerical value entered in the text box determines the period of time (in units determined by the Authentication Token Expiration Units menu selection) until an authentication token expires and is eligible to be purged.
- **Authentication Token Expiration Units**
  - enterprise.defaults.authtokens.expiration.period
  - Use the list to select the units of time to be used in conjunction with the Authentication Token Expiration setting. Minutes, Hours, and Days are the available options.
- **Authentication Token Purge Expired Frequency**
  - enterprise.defaults.authtokens.purgefrequency.value
  - The numerical value entered in the text box determines the period of time (in units determined by the Authentication Token Purge Expired Frequency Units menu selection) between purges of expired authentication tokens.
- **Authentication Token Purge Expired Frequency Units**
  - enterprise.defaults.authtokens.purgefrequency.period

- Use the list to select the units of time to be used in conjunction with the Authentication Token Purge Expired Frequency setting.

### 2.5.3.3 Password Settings

#### Properties

- Password - Minimum Length
  - `enterprise.security.password.composition.minimumlength`
  - The numerical value entered in the text box determines the minimum number of characters required for a valid password.
  - Note: This setting does not apply to passwords created by individuals in the Admin role.
- Password - Alpha Character Required
  - `enterprise.security.password.composition.require.alpha`
  - Select the True option to set the requirement that passwords contain at least one alphabetic character.
  - Note: This setting does not apply to passwords created by individuals in the Admin role.
- Password - Numeric Character Required
  - `enterprise.security.password.composition.require.numeric`
  - Select the True option to set the requirement that passwords contain at least one numeric character.
  - Note: This setting does not apply to passwords created by individuals in the Admin role.
- Password - Special Character Required
  - `enterprise.security.password.composition.require.special`
  - Select the True option to set the requirement that passwords contain at least one special character. The list of special characters used for the password are as follows:  
``~!@#$%^&*()-_+=[{ ]\ \ | ; : ' \ " , < . > / ?`
  - Note: This setting does not apply to passwords created by individuals in the Admin role.
- Password Expiration
  - `enterprise.security.password.expiration.days`
  - The numerical value entered in the text box determines the number of days until password expiration.
  - A value of zero or less disables password expiration.
- Password History Tracking
  - `enterprise.security.password.history.count`
  - The numerical value entered in the text box determines the number of prior passwords remembered by the system. The password history is used to prevent the reuse of passwords.

- Password History - Minimum Tracking Age
  - `enterprise.security.password.history.minagehours`
  - The numerical value entered in the text box determines the minimum number of hours a password must be in effect before it is added to the password history. This feature is useful for preventing circumvention of the password history by repeatedly changing passwords.

### 2.5.3.4 Plugin Login Settings

#### Properties

- Plug-in Login Module Class Name
  - `enterprise.loginmodules.pluggableloginmodule.classname`
  - Text entered in the text box identifies the class name for the plug-in login module.  
Click **+Add** as necessary to add class names for additional plug-in login modules.
- Plug-in Login Module Display Name
  - `enterprise.loginmodules.pluggableloginmodule.displayname`
  - Text entered in the text box appears as the display name of the plug-in login module.  
Click **+Add** as necessary to add display names for additional plug-in modules.
- Plug-in Login Module
  - `enterprise.loginmodules.pluggableloginmodule.enabled`
  - Enter `true` in the text box to enable the plug-in login module.  
Click **+Add** as necessary to add and enable/disable additional plug-in modules.
- Plug-in Login Module - Internal Checking
  - `enterprise.loginmodules.pluggableloginmodule.internalchecking`
  - Enter `true` in the text box to enable internal checking of password expiration and grace logins for the configured plug-in login module.  
Enter `false` in the text box when using LDAP-based, SSO-based, Container Managed or other external login modules to defer these types of checks to the external authentication system.  
Click **+Add** as necessary to add and enable/disable internal checking for additional plug-in modules.

### 2.5.3.5 LDAP and Active Directory Settings

#### Properties

- LDAP Server Host Name
  - `ldap.host`
  - Enter the host name for the LDAP server. This is a mandatory field.
- LDAP Server Port Number

- ldap.port
  - Enter the LDAP server port number. The default is 389. This is an optional field.
- LDAP Mask
  - ldap.mask
  - Enter the LDAP mask, either the authentication bind or search criteria. This is a mandatory field.
- Creation of Unapproved User Accounts
  - ldap.allow-user-creation
  - Select True to determine if unapproved user accounts can be created. The default value is true. This is an optional field.
- Assign default roles to users
  - ldap.assign-default-roles
  - Select true to determine, if users shall be assigned default roles.
- Auto create missing roles
  - ldap.auto-create-missing-roles
  - Select true to determine, if auto create missing roles is enabled.
- Auto create missing departments
  - ldap.auto-create-missing-depts
  - Select true to determine if auto create missing departments is enabled.
- LDAP Version
  - ldap.version
  - The LDAP version used, which can be either 2 or 3. The default value is 2. This is an optional field.
- Administrator Account Distinguished Name
  - ldap.bindDN
  - The distinguished name of the administrator account. This is an optional field.
- Administrator Account Password
  - ldap.bindPassword
  - The password for the administrator account. This is an optional field.
- Use SSL Connection
  - ldap.ssl.enabled
  - Enables an SSL Connection for LDAP. The default is false.
- Follow referrals
  - ldap.follow-referrals
  - If true, follow referrals.
- Retrieve data using the admin account
  - ldap.retrieve-data-as-admin



- If true, then bindDN must be set. The default is false.
- Search Start Location
  - ldap.baseDN
  - Start location for a search, works with ldap.mask and ldap.scope to define the search. This is an optional field.
- Search Scope
  - ldap.scope
  - Determines the scope of the search. This is an optional field.
- Attribute Name that Identifies a Found Entry
  - ldap.uniqueIDAttrib
  - Attribute name that identifies a found entry beyond the use of the distinguished name, if not specified, then defaults to uid. This is a mandatory field.
- Found Entry Email Attribute Name
  - ldap.emailAttrib
  - Email attribute name of a found entry. This is an optional field.
- Found Entry First Name Attribute Name
  - ldap.givennameAttrib
  - First name attribute name of a found entry. This is an optional field.
- Found Entry Middle Name Attribute Name
  - ldap.middlenameAttrib
  - Middle name attribute name of a found entry. This is an optional field.
- Found Entry Last Name Attribute Name
  - ldap.surnameAttrib
  - Last name attribute name of a found entry. This is an optional field.
- Found Entry Telephone Number Attribute Name
  - ldap.telephoneAttrib
  - Telephone number attribute name of a found entry. This is an optional field.
- Use LDAP Departments
  - ldap.enable-synch-depts
  - If true, get user departments from LDAP upon login.
- Department Attribute
  - ldap.deptAttrib
  - Name of attribute of the departments of the entry found.
- Use LDAP Roles
  - ldap.enable-synch-roles
  - If true, get user roles from LDAP upon login.
- Role Attribute

- ldap.rbac.roleAttrib
- Name of attribute of the roles of the entry found.
- Second Level Lookup Attribute
  - ldap.redirectDnAttrib
  - Attribute which identifies a second level lookup to retrieve user info; must be a dn(distinguished name). This is an optional field.
- Bridge Class Name
  - ldap.bridge.classname
  - This is a required field. The text for this property represents the class name of the bridge to be used.

### 2.5.3.6 Single Sign-On (SSO) Settings

#### Properties

- Username Header Name
  - enterprise.container.auth.username
  - The value entered in the text field represents the name of the header which contains the user's username.
- Firstname Header Name
  - enterprise.container.auth.firstname
  - The value entered in the text field represents the name of the header which contains the user's firstname.
- Middlename Header Name
  - enterprise.container.auth.middlename
  - The value entered in the text field represents the name of the header which contains the user's middlename.
- Lastname Header Name
  - enterprise.container.auth.lastname
  - The value entered in the text field represents the name of the header which contains the user's lastname.
- Status Header Name
  - enterprise.container.auth.status
  - The value entered in the text field represents the name of the header which contains the user's status.
- Email Header Name
  - enterprise.container.auth.email
  - The value entered in the text field represents the name of the header which contains the user's email.
- Phone Header Name
  - enterprise.container.auth.phone

- The value entered in the text field represents the name of the header which contains the user's phonenummer.
- Roles Header Name
  - enterprise.container.auth.roles
  - The value entered in the text field represents the name of the header which contains the user's roles.
- Department Header Name
  - enterprise.container.auth.depts
  - The value entered in the text field represents the name of the header which contains the user's departments.
- Use Container passed Departments
  - enterprise.container.auth.enable-synch-depts
  - Select the True option to get the user departments from the container.
- Departments passed within single header
  - enterprise.container.auth.depts-single-header
  - Select the True option to pass the departments within a single header.
- Department Delimiter
  - enterprise.container.auth.depts-delimiter
  - The value entered in the text field represents the delimiter between department names.
- Use Container passed Roles
  - enterprise.container.auth.enable-synch-roles
  - Select the True option to get user the roles from the container.
- Roles passed within single header
  - enterprise.container.auth.roles-single-header
  - Select the True option to pass the roles within a single header.
- Role Delimiter
  - enterprise.container.auth.roles-delimiter
  - The value entered in the text field represents the delimiter between role names.
- Assign default roles to users
  - enterprise.container.auth.assign-default-roles
  - Select the True option to determine if users shall be assigned with default roles.
- Auto create missing roles
  - enterprise.container.auth.auto-create-missing-roles
  - Select the True option to determine if the system will auto create missing roles.
- Auto create missing departments
  - enterprise.container.auth.auto-create-missing-depts

- Select the True option to determine if the system will auto create the missing departments.
- Pass All Cookies - JWS
  - `cmee.jws.pass-all-cookies`
  - Select the True option to pass all of the current cookies from the user's browser to Java Web Start for the Asset Editor to use.
- Suppress Authorization Header - JWS
  - `cmee.jws.suppress-authorization-header`
  - Select the True option to prevent authorization header from being propagated by JWS launched utilities.

### 2.5.3.7 Container Login Settings

#### Properties

- Fallback Authentication
  - `enterprise.loginmodules.fallbackauthentication`
  - Enables fallback authentication.
- Container Login Module Class Name
  - `enterprise.loginmodules.containerloginmodule.classname`
  - Text entered in the text box identifies the class name of the container login module.
- Container Login Module Display Name
  - `enterprise.loginmodules.containerloginmodule.displayname`
  - Text entered in the text box appears as the display name of the container login module.
- Container Login Module
  - `enterprise.loginmodules.containerloginmodule.enabled`
  - Select the True option to enable the container login module.
- Container Login Module - Internal Checking
  - `enterprise.loginmodules.containerloginmodule.internalchecking`
  - Select the True option to enable internal checking for the container login module.

## 2.5.4 Open API Settings

This section contains the following open API settings:

### 2.5.4.1 Open API

#### Properties

- Open API
  - `cmee.extframework.enabled`
  - Enables the Open API.

- Open API - Max Assets
  - `cmee.extframework.assets.max`
  - Maximum number of assets returned by queries through the Open API.
- Open API - Max Projects
  - `cmee.extframework.projects.max`
  - Maximum number of projects returned by queries through the Open API.
- Open API - Max Users
  - `cmee.extframework.users.max`
  - Maximum number of users returned by queries through the Open API.
- Open API - Location for temporary export file
  - `cmee.extframework.export.tempfile`
  - Path of temporary export file.
- REXAPI SSL TrustStore
  - `rexapi.ssl.trustStore`
  - Path to the trusted keystore used by the REX API to establish an SSL connection.
- REXAPI SSL TrustStore Password
  - `rexapi.ssl.trustStorePassword`
  - The password to the trusted keystore used by the REX API to establish an SSL connection.
- REXAPI SSL KeyStore
  - `rexapi.ssl.keyStore`
  - Path to the keystore used by the REX API to establish an SSL connection.
- REXAPI SSL KeyStore Password
  - `rexapi.ssl.keyStorePassword`
  - The password to the keystore used by the REX API to establish an SSL connection.

---

---

**Note:** For more information about REX Open API, see the "Repository Extensibility Framework" chapter in *Oracle Fusion Middleware Developer's Guide for Oracle Enterprise Repository*.

---

---

## 2.5.5 External Integration Settings

This section contains the following topics:

- Section 2.5.5.1, "UDDI"
- Section 2.5.5.2, "File Stores"
- Section 2.5.5.3, "Rational ClearCase"
- Section 2.5.5.4, "Eventing"

### 2.5.5.1 UDDI

#### Properties

- UDDI
  - `cmee.uddi.enabled`
  - Enables the UDDI Web Service Plugin.
- UDDI - Default Business
  - `cmee.uddi.default.business`
  - Use the list to select the default business entity to be assigned to any UDDI services to which no business has been assigned.
- Business Service Relationship Name
  - `cmee.uddi.business.service.relationship`
  - Use the list to select the relationship that will link UDDI Business Services to UDDI Business Entities.
- UDDI
  - `cmee.uddi.server.enabled`
  - Select the True option to allow Oracle Enterprise Repository to act as a UDDI registry for selected applications.
- Import - Default UDDI Service Type
  - `cmee.import.uddi.service.assettype`
  - Use the list to specify the default asset type to be assigned to imported UDDI services.

### 2.5.5.2 File Stores

#### Properties

- Advanced Access File Stores
  - `registry.advanced.filestores.enabled`
  - Select the True option to set advanced file store options.

### 2.5.5.3 Rational ClearCase

#### Properties

- Configuration Specification
  - `cmee.config-spec.enabled`
  - Select the True option to enable attachment of configuration specification files to assets.

### 2.5.5.4 Eventing

#### Properties

- Enable Event Manager
  - `cmee.eventframework.enabled`

- Select the True option to enable the Advanced Registration Flow subsystem.
- Event Manager Notifier Thread Sleep (seconds)
  - `cmee.eventframework.notifier.sleep`
  - Specifies in seconds how long the notifier should wait, if an endpoint is unavailable, before retrying to send an event.
- Event Manager Store Thread Sleep (seconds)
  - `cmee.eventframework.store.sleep`
  - Specifies in seconds how long the Event Manager's Store Manager thread should sleep before polling for the next available event stored in memory.
- Event Manager Delivery Thread Sleep (seconds)
  - `cmee.eventframework.delivery.sleep`
  - Specifies in seconds how long the Event Manager's Delivery Manager thread should sleep before selecting the next available event from the JMS server.

## 2.5.6 Import / Export and Introspection Settings

This section contains the following topics:

- [Section 2.5.6.1, "Import / Export"](#)
- [Section 2.5.6.2, "WSDL and XSD"](#)

### 2.5.6.1 Import / Export

#### Properties

- [Import/Export Client](#)
  - `cmee.importexport.enabled`
  - Select the True option to enable the Import / Export client.
- Import/Export job monitor max idle (msecs)
  - `cmee.extframework.impexp.monitor.maxidle`
  - The value entered in the text field represents the maximum time (in msecs) that an import/export job can be idle before it is automatically killed by the job monitor.
- Import/Export job monitor max idle for Rex transactions(msecs)
  - `cmee.extframework.impexp.monitor.rex.maxidle`
  - The value entered in the text field represents the maximum time (in msecs) that a rex transaction can be idle before it is automatically killed by the job monitor.
- Import/Export job monitor max runtime (msecs)
  - `cmee.extframework.impexp.monitor.maxruntime`
  - The value entered in the text field represents the maximum time (in msecs) that an import/export job can run before it is automatically killed by the job monitor.
- Import/Export job monitor period (msecs)
  - `cmee.extframework.impexp.monitor.period`

- The value entered in the text field represents the time (in msec) between scans for the job monitor.
- **Import/Export job monitor auto cleanup (msec)**
  - `cmee.extframework.impexp.monitor.autocleanup`
  - The value entered in the text field represents the time (in msec) that an import/export job's results remain in the system before they are automatically cleaned up by the job monitor.
- **Import/Export job proxy period (msec)**
  - `cmee.extframework.impexp.proxy.period`
  - The value entered in the text field represents the time (in msec) between checks by job proxies for job status on other cluster nodes.
- **Import/Export job proxy timeout (msec)**
  - `cmee.extframework.impexp.proxy.timeout`
  - The value entered in the text field represents the time (in msec) before a job proxy stops monitoring status of remote jobs on other clustered nodes.
- **Import/Export audit temp file location**
  - `cmee.extframework.impexp.audit.dir`
  - The value entered in the text field represents the directory in which temp files will be stored as they are created. These files are cleaned up after being served.

### 2.5.6.2 WSDL and XSD

#### Properties

- **Enable WSDL Introspection Element**
  - `cmee.wsdl-introspection.enabled`
  - Select the True option to enable an asset detail plugin element that displays WSDL Summary information.

## 2.6 Known Issues

This section describes the issues found in Oracle Enterprise Repository system setting properties:

### Duplicate System Setting Properties

In Oracle Enterprise Repository System Settings page, the following property names appear twice:

In Server Settings, the Server Resources settings section has the following duplicate properties:

#### WebApp JSP Path

```
enterprise.server.paths.jsp = {%cmee.server.paths.jsp%}
```

#### WebApp Servlet Path

```
enterprise.server.paths.servlet = {%cmee.server.paths.servlet%}
```

These two settings should remain static and reference their counterpart properties: `cmee.server.paths.jsp` and `cmee.server.paths.servlet` respectively.



---

---

# Customizing Oracle Enterprise Repository

This chapter provides information about customizing the Oracle Enterprise Repository Homepage and other interfaces for your organization.

This chapter contains the following sections:

- [Section 3.1, "Overview"](#)
- [Section 3.2, "Rebranding Oracle Enterprise Repository"](#)
- [Section 3.3, "Role/Department-based Homepage Display"](#)
- [Section 3.4, "Install Oracle Enterprise Repository Solution Packs"](#)

## 3.1 Overview

Organizations can customize the look and feel of Oracle Enterprise Repository. The Oracle Enterprise Repository color palette can be modified. In addition, the Oracle Enterprise Repository home page can be customized by role or by department. This allows organizations to promote governance and specific assets to a subset of Oracle Enterprise Repository users.

You can access the Oracle Enterprise Repository homepage and customize it with the your company logo and company specific information. This section contains the following topics:

- [Section 3.1.1, "Accessing the Oracle Enterprise Repository Home Page"](#)
- [Section 3.1.2, "Customizing the Oracle Enterprise Repository Home Page"](#)

### 3.1.1 Accessing the Oracle Enterprise Repository Home Page

To access the Oracle Enterprise Repository home page, perform the following steps:

1. Log in to `http://<host>:<port>/<oer_instance_name>` where you have installed Oracle Enterprise Repository. Alternatively, you can also log in to `http://<host>:<port>/<oer-app-name>`.
2. On the login screen, enter the appropriate information in the **Username** and **Password** boxes.
3. Click **Login**. The Login Status dialog displays a welcome message that includes the date and time of your login.
4. In the Login Stats dialog, click **Close**.

The Oracle Enterprise Repository home page is displayed.

---



---

**Note:** If your OER installation is behind a firewall, you may need to set a proxy in your browser to view the OER Homepage.

---



---

### 3.1.2 Customizing the Oracle Enterprise Repository Home Page

To customize the Oracle Enterprise Repository home page with the your company logo and company specific information, perform the following steps:

1. Log into the Oracle Enterprise Repository home page.
2. Set the `cmee.enterprisetab.homepage` setting to any Web site of your choice. By default, this system setting is set to `http://localhost:8080/oer/custom/home.jsp`.
3. Alternatively, you can also create a Web page and add it to the Custom directory in the Oracle Enterprise Repository installation. The Custom directory path depends on the application server that Oracle Enterprise Repository is installed on.

## 3.2 Rebranding Oracle Enterprise Repository

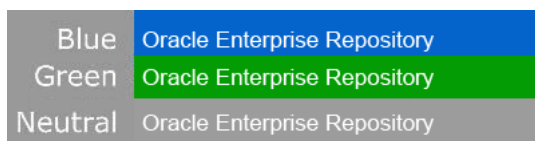
This section describes how to rebrand Oracle Enterprise Repository and how to customize skins in Oracle Enterprise Repository. It contains the following topics:

- Section 3.2.1, "Start with CSS"
- Section 3.2.2, "Customizing Skins"

### 3.2.1 Start with CSS

Oracle Enterprise Repository uses CSS (cascading style sheets) that can be edited to control colors and other visual aspects of the application. Oracle Enterprise Repository is built on a mostly-neutral color palette and is branded around a single color. You can edit the color and type faces from the CSS, as shown in Figure 3–1.

**Figure 3–1 Cascading Style Sheets**



This figure illustrates the Cascading Style Sheets pane.

\*\*\*\*\*

Oracle Enterprise Repository ships with the following skins:

- Blue -- #0066CC
- Green -- #669933
- Neutral -- #999999

Oracle Enterprise Repository ships with the Oracle skin as the default.

The skins also include these colors:

- White -- #FFFFFF
- Grey -- #EEEEEE

## 3.2.2 Customizing Skins

This section describes how you can customize the skin in Oracle Enterprise Repository. It contains the following topics:

- Section 3.2.2.1, "Choosing a Skin"
- Section 3.2.2.2, "Creating a Skin"
- Section 3.2.2.3, "Images"
- Section 3.2.2.4, "Activating Your Custom Skin"
- Section 3.2.2.5, "Installing a Oracle-supplied Custom Skin in Oracle Enterprise Repository"
- Section 3.2.2.6, "Using the Directory Structure"

### 3.2.2.1 Choosing a Skin

Oracle Enterprise Repository ships with the Oracle skin as the default, but you can change the active skin. The file that controls the skin is the `WEB-INF/classes/skin.properties` file. In the `skin.properties` file, you can change the `skin.name.org` property to either "green", "blue", or "neutral" for the skin you want to use and restart the server, then new color palette for the application is displayed.

### 3.2.2.2 Creating a Skin

If you want to create your own custom skin for Oracle Enterprise Repository, you must create a new folder for the skin and add it to the skins folder. A good way to get started is to copy an existing skin folder and rename it. Oracle has designed the interface of Oracle Enterprise Repository around a single color. A recommended best-practice is for a new skin's color palette to be in a similar, mid-tonal range as the default palettes shipped with Oracle Enterprise Repository.

To update the CSS with your new color, do a search and replace on the hex number of the color you want to replace. For instance, if you copied the Blue skin, replace 0066CC with the new color. Do this for the `settings.css` and `style.css` files. The following CSS files are located in the `<OER_home>/applications/oer/oer-web/skins/blue` directory:

- `settings.css`
- `style.css`
- `tabs.css` (ignore this file)
- `tree.css` (ignore this file)

### 3.2.2.3 Images

You can complete rebranding by updating the default images. The images are stored in two different places, but the paths look the same. As a best-practice, your new images should be the same size (width and height) as the images they replace. The following images are located in the `<OER_home>/applications/oer/oer/oer-web/skins/blue/images/menu` directory:

- `smalllogo.gif` (271 X 21)
- `wave.jpg` (306 X 31)
- `oracleicon.gif` (68 X 51)
- `logo.gif` (414 X 28)

- menuLogo.jpg (423 X 54)
- login\_splash.jpg (338 X 508)

This location on "skins" folder has all the color skins including default oracle. The folder also has all the .css and other folders.

#### 3.2.2.4 Activating Your Custom Skin

After completing the previous steps, you should have a custom skin that looks much like one of default skins in the skins folder. The final step is to create a new skin.properties file for your new skin. This is accomplished by copying and editing an existing skin.properties file. There are two items in the properties file that need to be edited: the name and the color values.

- In the line `skin.name.org=blue`, replace blue with the name of the new skin.
- Replace any hexadecimal color values with the values for the colors in the new skin's palette.

Once these changes are made, the new skin is activated by restarting the server.

#### 3.2.2.5 Installing a Oracle-supplied Custom Skin in Oracle Enterprise Repository

To install a Oracle-supplied Custom Skin:

1. Obtain the custom skin ZIP file from OTN.
2. Extract the contents of the ZIP file into a temporary directory. Be sure to retain the directory paths for each of the files within the ZIP file.
3. Make a backup copy of the existing Oracle Enterprise Repository `<FMW_home>/user_projects/domains/oer/config-oer/skin.properties` file.
4. Copy the new skin.properties file from the custom skin ZIP archive to the `<FMW_home>/user_projects/domains/oer/config-oer/` directory.
5. Move the extracted custom skin directory containing the images and CSS files into the `<OER_home>/applications/oer/oer/oer-web/skins/` directory. The skin.properties file points to this directory to find the skin resources.
6. Restart the application server to apply the changes.

#### 3.2.2.6 Using the Directory Structure

This section describes how to use the directory structure for creating skins. It contains the following topics:

- "Creating a New Skin"
- "Using the Custom Tag Library"
- "Using the Skintags Library Tags"
- "CSS"
- "Image"

#### Creating a New Skin

To create a new skin:

1. Create a `your_skins_folder` under `/skins`.
2. Change CSS and images as needed.

3. Edit the `skins.properties` file so the entry `skins.org.name` reflects your `_skins_` folder.
4. Restart the application server.

You must follow the naming conventions as shown in the example below:

```
/skins/your_skins/folder/style.css
/skins/your_skins/folder/images/logo.gif
/skins/your_skins/folder/buttons/all_the_buttons
/skins/your_skins/folder/icons/the_other_icons
```

### Using the Custom Tag Library

A skintags custom tag library is used to point CSS and image references to the proper `/skins` folder. Currently, there are two skin tags: `css` and `image`. Existing JSPs have been updated. Any new JSPs must use the skintags library.

### Using the Skintags Library Tags

To use any of the skin tags, add the following to the JSP:

```
<%@ taglib uri="/skintags" prefix="skin" %>
```

(The prefix can be anything you like because it is a reference within the JSP only).

### CSS

Tags for cascading style sheets resemble the following example:

```
<skin:css element='the_relative_path_to_the_style_sheet' />
```

To use it within a stylesheet:

```
<LINK HREF="" REL="stylesheet" TYPE="text/css">
```

### Image

An image tag resembles the following example:

```
<skin:image element='the_relative_path_to_the_image' />
```

The example below illustrates the use of an image tag within an HTML `IMG` tag:

```
<IMG SRC="" WIDTH="182" HEIGHT="32" BORDER="0" ALT="Geneva">
```

(where the `relative_path` starts from the skin folder (as defined in the `skin.properties` file.)

## 3.3 Role/Department-based Homepage Display

This feature displays a specific homepage based on user role or department assignment. This section contains the following topic:

- Section 3.3.1, "Installation"

### 3.3.1 Installation

You can customize the Oracle Enterprise Repository homepage by role or by department. Perform the following steps:

1. Create a file named `homepages.xml`. See Example 3–1.
2. Add the `homepages.xml` file to the application classpath. The Oracle Enterprise Repository application's `WEB-INF/classes` directory is recommended.

3. Set the Homepage property `cmee.enterprisetab.homepage` to `http://SERVER/APP/custom/autoselect.jsp`, where *SERVER* is the name of the server and *APP* is the context name of the Oracle Enterprise Repository Web application.

Homepage selection is defined by the contents of the `homepages.xml` file.

#### **Example 3–1 Example homepages.xml File**

The following provides an example of the contents of a `homepages.xml` file:

```
1 <HomepageSelector>
2 <select criteria="roles">
3 <option value="Admin">http://www.oracle.com</option>
4 <option value="Architect">architect.jsp</option>
5 <option value="Developer">developer.jsp</option>
6 <option value="User">generaluser.jsp</option>
7 <defaultLocation>homepageDefault.jsp</defaultLocation>
8 </select>
9 </HomepageSelector>
```

Note the following line numbers:

- Lines 2:
  - Sets the value for criteria. Valid values are
    - \* roles
    - \* departments
- Lines 3-6:
  - The option value is indicated by the name of the role or department used for selection.
  - The node value is a relative path or absolute path to the page to which the user is redirected. The path for a home page hosted on the Oracle Enterprise Repository application server is relative to the `custom/autoselect.jsp` file. In the above example, the `architect.jsp` file is also located in the `custom` directory.
  - Order is important. Users with multiple roles or memberships in multiple departments are directed to the first matching selection.
- Line 7
  - Identifies the default homepage for users who do not meet any of the selection criteria.

---

---

**Note:** After configuring `homepages.xml`, if you encounter a "404 Not found" error when redirecting to a custom jsp, then delete all the cached data in `<domain directory>/servers/<server name>/tmp` folder, and then redeploy.

---

---

## 3.4 Install Oracle Enterprise Repository Solution Packs

This section describes the install instructions for the base data pack and harvester solution pack.

### Base Data Pack

Base data pack is installed as part of the Oracle Enterprise Repository database installation. The Base data is not available by any other method. It is not delivered during any upgrades from one version of Oracle Enterprise Repository to the next.

The asset types delivered as part of Base Data are largely considered templates. The creation of a new asset type originates by cloning something that already exists. It is recommended that two best practices are applied to these asset types.

1. The base data asset types must not be deleted until at least one asset type is created, which is specific to the new Oracle Enterprise Repository application.
2. If enhancements are desired to a base asset type, it is recommended to change that base asset type. Oracle delivered types are used in the various asset models and are necessary to ensure migration to future Oracle models.
3. If enhancements are desired to a solution pack delivered asset type, it is essential that the elements delivered with the Solution Pack are not removed from the type. The recommendation is that
  - For the Editor: a new tab with a name such as Deprecation be created and the elements moved to that new tab
  - For the Viewer: the elements are moved to the section called Hidden Elements.

### Harvester Solution Pack

The Harvester is available in **12.1.3.0.0-OER-Harvester-Solution-Pack.zip**, which is bundled with the Oracle Enterprise Repository installation.

The installation of the Harvester Solution Pack requires use of the Import /Export tool. This tool is available from the Admin tab of Oracle Enterprise Repository in the Import Export option. The asset types delivered as part of the Harvester Solution Pack are required by the Harvester tool.

For information about the Import/Export tool, see Chapter 5, "Import Export Tool".

### Contract Management Solution Pack

The Contract Management Solution Pack is bundled with the Oracle Enterprise Repository installation.

The installation of the Contract Management Solution Pack requires use of the Import /Export tool. This tool is available from the Admin tab of Oracle Enterprise Repository in the Import Export option. For information about the Import/Export tool, see Chapter 5, "Import Export Tool".





---

---

## Configuring E-Mail Notifications and Distribution Lists

This chapter describes how to configure e-mail notifications and templates and manage distribution lists.

This chapter includes the following sections:

- [Section 4.1, "Overview"](#)
- [Section 4.2, "Setting Up E-Mail Notification"](#)
- [Section 4.3, "Managing E-mail Distribution Lists"](#)
- [Section 4.4, "Working with E-Mail Templates"](#)

### 4.1 Overview

Oracle Enterprise Repository supports automatic e-mail to a variety of participants in the asset registration process. E-mails are sent based on repository events, whether the events are triggered by interaction with OER via the user interface, or by OER automated workflows. Automatic e-mails can be enabled or disabled and templates for these e-mails can be configured for a particular OER instance. Recipients of both automated notifications and discretionary subscriber e-mail have the ability to designate an e-mail distribution list for e-mails that come to them from OER.

#### 4.1.1 Using E-Mail Notifications and E-Mail Distribution Lists

Users can create e-mail distribution lists so that the list members are automatically copied on certain e-mail notifications that are generated as part of the asset registration process. Distribution lists can help automate the asset registration workflow because members of the distribution list are notified about asset registration events that the list owner is involved in as an asset reviewer or approver. For example, if a list owner is expecting to be absent, they can "delegate" asset review and approval responsibility by adding other authorized users to their e-mail distribution list.

### 4.2 Setting Up E-Mail Notification

To receive e-mail notification when a new version of an asset is created, you need to enter the `cmee.registrar.email` system setting with the e-mail address. To update the `cmee.registrar.email` system setting with an e-mail address, navigate to the Admin page, System Settings. Enable the system setting, if you already have not, by entering `cmee.registrar.email` in the Enable New System Setting field, and then click **Enable**. If the setting already exists, then enter `cmee.registrar.email` in the Search field. In the Notifications section, enter the e-mail address in the System Registrar E-mail field,

as shown in [Figure 4-1](#). Click **Save** to save the settings.

**Figure 4-1 Functional Settings**

### 4.3 Managing E-mail Distribution Lists

Distribution lists are created on a user's MyStuff -> Configuration Options -> Manage E-mail Notifications page. Existing Oracle Enterprise Repository users can be selected for a distribution list, as well as external e-mail addresses, but only if that option is permitted by the Oracle Enterprise Repository administrator.

These procedures are performed in the E-mail Notifications section on the My Stuff page, as shown in [Figure 4-2](#).

**Figure 4-2 E-mail Notifications**

#### 4.3.1 Creating a Distribution List

Perform the following steps to create an e-mail distribution list.

1. Click **Create List** to open the Update Distribution List dialog, as shown in [Figure 4-3](#).

**Figure 4–3 Update Distribution List**

2. Click **Create List** to open the Update Distribution List dialog.
  - a. Enter appropriate text in the Name text box.
  - b. Use the Department list to narrow the search.
  - c. Click **Search**.

Search results appear in the list in the Search Results column.

3. Add users to the distribution list by using the arrow buttons to move users from the Search Results column to the Selected Users column. To add or remove all users from either column, use the All >> and All << buttons.

---

**Note:** Distribution list can contain a maximum of 20 members.

---

4. To add external users to the list, type the e-mail addresses in the Additional E-mail Addresses box, using commas as delimiters.  
For example: jack@example.com,jill@example.com,jake@example.net
5. Click **OK** to save the distribution list.

**Figure 4–4 E-mail Notifications**

### 4.3.2 Updating a Distribution List

Follow these steps to update an e-mail distribution list.

1. Click **Update List** to open the Update Distribution List dialog.
2. Add new members or remove members following the instructions in [Section 4.3.1, "Creating a Distribution List"](#).
3. Click **OK** to save the distribution list.

### 4.3.3 When List Members Are Copied on E-mails

The members of a user's distribution list are copied on an e-mail in the following notification e-mails:

- Asset Accepted
- Asset Assign
- Asset Registered
- Asset Submission/Rejection
- Asset Unaccepted
- Asset Unassigned
- Asset Unregistered
- Asset Unsubmitted

## 4.4 Working with E-Mail Templates

Certain specific events within Oracle Enterprise Repository trigger automated e-mail responses. For instance, when a new asset is submitted, an e-mail notification is sent to the registrar. The various templates are described below.

- **All Asset Tabs Approved:** Notifies the registrar and the users assigned to the asset that all tabs have been approved.

An asset's tabs can be approved in the Asset Editor. Each tab, with the exception of the Administration tab, has an Approve button. This button will not be available until the asset is has been accepted, which can be done under the Administration tab.

- **Asset Accepted:** Notifies the submitter that the asset has been accepted.

An asset can be accepted in the Asset Editor under the Administration tab. The Accept option is not available until the asset has been submitted.

- **Asset Active:** Notifies the submitter and subscribers that the asset has been marked active.

An asset can be marked as Active in the Asset Editor under the Administration tab under the Status section.

- **Asset Assign:** Notifies users that the asset has been assigned to them for processing.

An asset can be assigned to a user in the Asset Editor under the Administration tab within the Assign Users section.

- **Asset Deletion:** Notifies the submitter and subscribers that the asset has been deleted.

An asset can be deleted in the Asset Editor under the Administration tab within the Status section.

- **Asset Editor Submission:** Notifies the registrar of an asset submission via the Asset Editor.  
Assets can be submitted within the Asset Editor by either selecting File->New or File->Copy/Migrate.
- **Asset Inactive:** Notifies the submitter and subscribers that the asset has been marked inactive.  
An asset can be marked inactive in the Asset Editor under the Administration tab within the Status section.
- **Asset Registered:** Notifies the submitter that an asset has been registered.  
An asset can be registered in the Asset Editor under the Administration tab. Assets must be submitted and accepted before they can be registered.
- **Asset Retired:** Notifies the submitter and subscribers that the asset has been retired.  
An asset can be retired in the Asset Editor under the Administration tab within the Status section.
- **Asset Submission Rejection:** Notifies the submitter that the asset has been rejected.  
An asset can be rejected by the Registrar in the Asset Editor.
- **Asset Tab Approved:** Notifies the Registrar that a specific tab has been approved.  
An asset's tabs can be approved in the Asset Editor. Each tab, with the exception of the Administration tab, has an Approve button. This button will not be available until the asset is has been accepted, which can be done under the Administration tab.
- **Asset Tab Unapproved:** Notifies the Registrar that a specific tab has been unapproved. For unregistered assets, the users assigned to the asset are also notified.  
An asset's tabs can be unapproved in the Asset Editor. Each tab, with the exception of the Administration tab, has an Approve or Unapprove button, depending on the state. This button will not be available until the asset has been accepted, which can be done under the Administration tab.
- **Asset Unaccepted:** Notifies the submitter that the asset has been unaccepted.  
An asset can be unaccepted in the Asset Editor under the Administration tab. An asset must have been submitted before the accept/unaccept option is made available.
- **Asset Unassign:** Notifies users that were previously assigned to an asset that they have been unassigned.  
A user can be unassigned to an asset in the Asset Editor under the Administration tab within the Assign Users section.
- **Asset Unregistered:** Notifies the submitter that an asset has been unregistered.  
An asset can be unregistered in the Asset Editor under the Administration tab. Assets must be submitted and accepted before they can be registered/unregistered.
- **Asset Unsubmitted:** Notifies the submitter that the asset has been sent to the Unsubmitted queue.

An asset can be unsubmitted in the Asset Editor under the Administration tab. An asset must be unaccepted before it can be unsubmitted.

- **Repository Submission:** Notifies the registrar of an asset submission.

An asset can be submitted from the Web Tool under the Assets tab by selecting Submit an Asset.

- **Usage Query 1:** Queries the asset consumer regarding the intended user off an asset.

This is a notification to remind the user to submit a survey regarding this asset. The survey can be found in the Web Tool under the My Stuff tab within the Consumed Assets section. In the detail of the given asset is a button called Update Survey Status. This button will only be available if the Consumer Survey system setting is enabled. The setting is `cmee.extraction.consumer-survey.enable`.

- **Usage Query 2:** Depending on the response to Usage Query 1, asset consumers may receive a second query regarding the intended use of the asset.

This is a notification to remind the user to submit the survey regarding this asset. The survey can be found in the Web Tool under the My Stuff tab within the Consumed Assets section. In the detail of the given asset is a button called Update Survey Status. This button will only be available if the Consumer Survey system setting is enabled. The setting is `cmee.extraction.consumer-survey.enable`.

#### 4.4.1 Viewing E-Mail Templates

You can configure e-mail notifications in the E-mail Templates section of the Admin tab. Each E-mail Template has a list of valid substitutions, which are available for use, in that given template. The examples of substitutions are `<%asset.name%>`, which populates the template with the name of an asset. The list of valid substitutions varies for each of the 30 templates available in Oracle Enterprise Repository. To view the e-mail notifications in Oracle Enterprise Repository, perform the following steps:

1. Log in to Oracle Enterprise Repository using your login credentials. The Oracle Enterprise Repository home page is displayed.
2. Click **Admin**. The Admin page is displayed.
3. Click **E-mail Templates**. The list of E-mail Templates along with its status and description is displayed, as shown in [Figure 4-5](#).



**Figure 4-5 List of E-mail Templates**

Name	Status	Description
All Asset Tabs Approved	Active	Notifies the registrar and the users assigned to the asset that all tabs have been approved.
Asset Accepted	Active	Notifies the submitter that the asset has been accepted.
Asset Active	Active	Notifies the submitter and subscribers that the asset has been marked active.
Asset Assign	Active	Notifies users that the asset has been assigned to them for processing.
Asset Deletion	Active	Notifies the submitter and subscribers that the asset has been deleted.
Asset Editor Submission	Active	Notifies the registrar and the users assigned to the asset that it has been expired.
Asset has been expired	Active	Notifies the submitter and subscribers that the asset has been marked inactive.
Asset Inactive	Active	Notifies the submitter that an asset has been registered.
Asset Registered	Active	Notifies the registrar and the users assigned to the asset that the asset registration status status ' <code>&lt;%asset.reg.status%&gt;</code> ' has remained unchanged for more than <code>&lt;%action.pending.days%&gt;</code> days.
Asset Regressed.	Active	Notifies the submitter and subscribers that the asset has been retired.
Asset Retired	Active	Notifies the submitter that the asset been rejected.
Asset Submission Rejection	Active	Notifies the Registrar that a specific tab has been approved.
Asset Tab Approved	Active	Notifies the Registrar that a specific tab has been unapproved. For unregistered assets, the users assigned to the asset are also notified.
Asset Tab Unapproved	Active	Notifies the submitter that the asset has been unaccepted.
Asset Unaccepted	Active	Notifies users that were previously assigned to an asset that they have been unassigned.
Asset Unassign	Active	Notifies the submitter that an asset has been unregistered.
Asset Unregistered	Active	Notifies the submitter that the asset has been sent to the Unsubmitted queue.
Asset Unsubmitted	Active	Notifies the contact specified in the asset Notification Email field that the asset has been used.
Asset Used	Active	Notifies project leaders when a compliance template has been applied to a project.
Compliance Template Applied	Active	Notifies the registrar and the users assigned to the asset that the metadata has changed.
Metadata of asset has changed.	Active	Notifies subscribers when a new version of an asset is being developed.
New Asset Version Under Development	Active	Notifies subscribers that a new version of the asset is registered.
New Version Registered	Active	Notifies the registrar and the users assigned to the asset that it is due for expiration.
Prior to expiration	Active	Notifies the registrar and the users assigned to the asset that the registration status status ' <code>&lt;%asset.reg.status%&gt;</code> ' has remained unchanged for more than <code>&lt;%action.pending.days%&gt;</code> days.
Registration status unchanged.	Active	Notifies the registrar of an asset submission.
Repository Submission	Active	Notifies the registrar and the users assigned to the expired asset that the status has changed.
Status of expired asset has changed.	Active	Queries the asset consumer regarding the intended use of the asset.
Usage Query 1	Active	Depending on the response to Usage Query 1, asset consumers may receive a second query regarding the intended use of the asset.
Usage Query 2	Active	Notifies the user to whom an asset usage record has been reassigned.
Usage Reassigned		

- Click any template to display the details of that E-mail Template in the pane below. For example, click Asset Used. Figure 4-6 describes the "Asset Used" e-mail template, which is sent to notify the contact specified in the asset Notification E-mail field that the asset has been used.

**Figure 4-6 E-mail Template - Asset Used**

Email Template: **Asset Used**

**Name:** Asset Used

**Description:** Notifies the contact specified in the asset Notification Email field that the asset has been used.

**Status:** Active

**Subject:** `<%user.name%> has used <%asset.name%>`

**Mail From:** `<%user.email%>`

**Message:** `<%user.name%> has used asset <%asset.name%> in a project.`

(Max 4000 Characters)

**Valid Substitutions:**

- `<%asset.name%>` - The display name of the used asset
- `<%producingprojects.names%>` - The name of the projects that produced the asset
- `<%project.name%>` - The name of the project in which the asset was used
- `<%registrar.email%>` - The registrar's email address
- `<%user.email%>` - The email address of the user who submitted the asset
- `<%user.name%>` - The name of the user who submitted the asset

Save Cancel

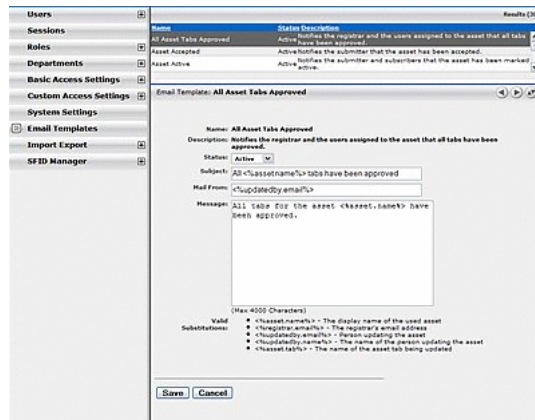
- Click Save.
- Similarly, you can view the details of the other E-mail Templates.

#### 4.4.2 Editing E-Mail Templates

The procedure begins on the Oracle Enterprise Repository Admin screen.

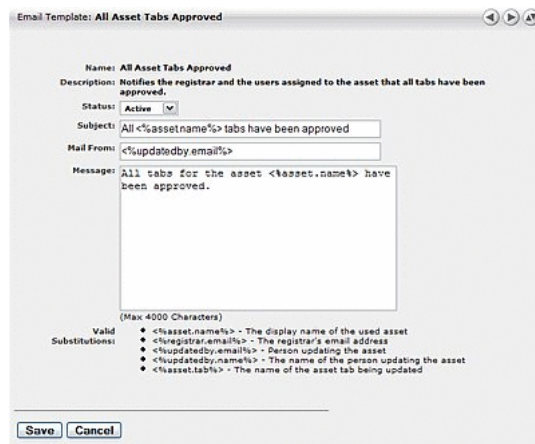
1. Click **E-mail Templates** in the Admin sidebar. The list of e-mail templates appears in the upper frame of the main pane.

**Figure 4–7 List of E-mail Templates**



2. Select the e-mail template to be edited from the list. The E-mail Template edit screen opens in the bottom frame.

**Figure 4–8 E-mail Template: All Asset Tabs Approved**



3. Use the Status list to activate/deactivate the e-mail message.
4. Edit the template information as necessary.

### Notes on editing

- The Status field can be set to Active or Inactive. When inactive, no e-mail notification is sent for the specified event. Active is the default status.
- Substitutions are dynamic variables that can be placed within the subject, message, and Mail From e-mail address. When the notification is sent, these variables is replaced with the appropriate values. The following example demonstrates the syntax for entering a substitution for the name of the asset:

*You have submitted the asset called <%asset.name%>*

A list of acceptable substitutions is displayed with each e-mail notification on the E-mail Templates screen.



---

---

## Import Export Tool

This chapter describes the features of the Import Export tool and how to use it to import items into and export items from the Oracle Enterprise Repository.

This chapter includes the following topics:

- [Overview](#)
- [Using the Import/Export Tool](#)
- [Starting the Import/Export Tool](#)
- [Export Items from Oracle Enterprise Repository](#)
- [Importing Items into Oracle Enterprise Repository](#)

### 5.1 Overview

The Oracle Enterprise Repository Import/Export Tool is designed to allow the interchange of assets and related metadata between instances of Oracle Enterprise Repository.

---

---

**Note:** When using the Import/Export tool, adhere to the following best practices:

- Exporting from Oracle API Catalog and importing to Oracle API Catalog is supported
- Exporting from Oracle Enterprise Repository and importing to Oracle Enterprise Repository is supported
- Exporting from Oracle API Catalog and importing to Oracle Enterprise Repository is supported
- Exporting from Oracle Enterprise Repository and importing to Oracle API Catalog is supported only with use of the Export to API Catalog utility. You can use the Import/Export tool to import zip files created by this utility. Using the Import/Export tool in other ways to import assets from OER to OAC is NOT SUPPORTED and NOT RECOMMENDED.

See "Installing the Export to API Catalog Feature Patch" in *Oracle Fusion Middleware Installation Guide for Oracle Enterprise Repository* for information about installing the utility and "Exporting API Assets to API Catalog" in *Oracle Fusion Middleware Developer's Guide for Oracle Enterprise Repository* for information about using the utility to export API assets from OER to OAC.

---

---

Archives must be completely self-contained. That is, all dependent data required to fully describe all exported items are also exported. The dependencies are as follows:

- Assets, Compliance Templates, and Policies rely on Types
- Asset Types, Compliance Template Types, and Policy Types rely on Categorization Types and Relationship Types

For example, when exporting a single Asset, the resultant archive contains that Asset, its associated Asset Type, and any Categorization Types and Relationship Types on which that Asset Type depends. When importing, all data contained within the archive is imported into the destination instance of Oracle Enterprise Repository. Conflicts resulting from the existence of items with the same name are resolved in the following manner:

- Assets, Compliance Templates, Policies, and their respective Types must have unique names within the system. When an imported item has the same name as an item already in the destination Oracle Enterprise Repository instance, "v2" is appended to the name of the new (imported) item. Each iteration of this occurrence results in a new version of the item: "v3," "v4," and so on. If the name of an imported item is already appended with a "Vn" designation, "v2" is further appended to that name.
- Categorization Types merge data in the event of a destination Categorization Type of the same name. Categorizations in the source archive that are not present on the destination system are added to the existing (destination) Categorization Type.
- Source and destination Relationship Types sharing the same name and type are ignored; existing descriptive text is retained on the destination system. When source and destination Relationship Types share the same name, but are of different types, the name of the source Relationship type is appended with "v2," as described above, upon import to the destination system.

---

---

**Note:** If your instance of Oracle Enterprise Repository is or will be configured to be secured by Siteminder, then you must configure the policy server to ignore (or unprotect) the following URL to allow the Import/Export tool to function properly

*<http://appserver.example.com/oe/services/>*

---

---

## 5.2 Using the Import/Export Tool

This section describes how you can use the Import/Export tool to import or export objects into or from Oracle Enterprise Repository. This section contains the following topics:

- [Section 5.2.1, "Overview"](#)
- [Section 5.2.2, "The parameter.properties File"](#)
- [Section 5.2.3, "The Default Import Behavior"](#)

---

---

**Note:** The Import/Export tool requires Java Web Start. See [Section 1.10, "Install Java Web Start"](#) for more information about installing and configuring Java Web Start.

---

---

## 5.2.1 Overview

The Import/Export tool is available from within Oracle Enterprise Repository in the Admin tab, Import Export page. The Oracle Enterprise Repository user is required to have the Admin role specified by the system setting "Admin Role" (`enterprise.security.roletype.admin`).

The Import/Export tool is a Java WebStart program that downloads and launches when invoked.

The Export operation enables you to search for objects in Oracle Enterprise Repository to export. You can search for Asset Types, Assets, Categorization Types, and Relationship Types. The result of the Export operation is a .zip file with the searched for objects and all dependencies of that object. By default, a sample `parameter.properties` file is contained within the resulting .zip file.

The Import operation takes a .zip file that was exported by the Import/Export tool and loads the contained items into Oracle Enterprise Repository. The behavior for the Import is determined by `META-INF/parameter.properties` or a series of default actions.

---

**Note:** You can also import assets exported from the search results screen, as described in "Exporting Asset Search Results to ZIP" in *Oracle Fusion Middleware Developer's Guide for Oracle Enterprise Repository*.

---

Oracle Enterprise Repository Administrators who must move data between different instances of Oracle Enterprise Repository must use the Import/Export tool or the Oracle Enterprise Repository OpenAPI. When the Import/Export tool is utilized, a `parameter.properties` file must be included in the .zip bundle before performing the import operation. The Export operation creates a placeholder file located in the `/META-INF` directory.

## 5.2.2 The `parameter.properties` File

Each import bundle contains a `/META-INF` directory. This directory contains a file that controls how the Import/Export tool behaves when the bundle is imported. If the `parameter.properties` file does not exist, it must be created before any Import operation.

### 5.2.2.1 Supported Parameters in the `parameter.properties` File

The supported parameters in the properties file are listed with the legal values for the parameter:

#### PreserveUUIDs

- **Used by:** Asset
- **Description:** This parameter retains the UUIDs of assets in the bundle rather than randomly generating new UUIDs
- **Values:**
  - **true** - UUIDs on newly created assets are UUID specified in the import bundle
  - **false** - UUIDs on newly created assets are randomly generated by Oracle Enterprise Repository

**Standard.saver.assettype.props.duplicate.matching**

- **Used by:** Asset Type
- **Description:** This parameter specifies how the import tool identifies that an asset type in the bundle is a duplicate of something that already exists in Oracle Enterprise Repository.
- **Values:**
  - **UUID** - A duplicate is determined if the UUID of the asset type specified in the import bundle matches the UUID of an asset type in Oracle Enterprise Repository.
  - **Name** - A duplicate is determined if the Name of the asset type specified in the import bundle matches the Name of an asset type in Oracle Enterprise Repository.

**Standard.saver.assettype.props.duplicate.handling**

- **Used by:** Asset Type
- **Description:** This parameter specifies how the import tool handles asset types in the bundle that are identified to be Duplicates of asset types in Oracle Enterprise Repository.
- **Values:**
  - **Create New Version** - Create a new asset type with the UUID and archetype from the import bundle, but a name which is versioned with a suffix of a "V" and a number (for example, Service V2). The asset type has a new UUID.
  - **Ignore** - Ignore the duplicate asset type from the bundle. Do not import the asset type and do not modify the asset type in Oracle Enterprise Repository.
  - **Merge** - Merge the asset type in Oracle Enterprise Repository with the duplicate asset type from the bundle. The Merge operation does not modify any of the elements of the asset type in Oracle Enterprise Repository. The operation is additive, appending elements in the asset type from the bundle to the asset type in Oracle Enterprise Repository. Matching of the elements is by the XML Mapping of the element. The rules are:
    - \* If the element's XML Mapping exists, then ignore the element.
    - \* If the element's XML Mapping does not exist, then add it. The element is added by first locating the element's tab, and then creating the tab if it does not exist. The element is then added to the end of the tab.
  - **Overwrite** - Replace the asset type in Oracle Enterprise Repository with the duplicate asset type from the bundle. The assets of the asset type in Oracle Enterprise Repository is not modified, although the display of these assets is affected.

**Standard.saver.relationshiptype.props.duplicate.handling**

- **Used by:** Relationship Type
- **Description:** This parameter specifies how the import tool handles relationship types in the bundle that are identified to be Duplicates of relationship types in Oracle Enterprise Repository. Duplicate relationship types are always matched by the Name and Direction characteristics of the relationship type.
- **Values:**

- **Ignore** - Ignore the duplicate relationship type from the bundle. Create any asset relationships that use this relationship type and do not already exist in Oracle Enterprise Repository.
- **Overwrite** - Replace the relationship type in Oracle Enterprise Repository with the duplicate relationship type from the bundle. This overwrites all information for the relationship type. This also creates any asset relationships that use this relationship type and do not already exist in Oracle Enterprise Repository.

#### **Standard.saver.metadata.props.duplicate.handling**

- **Used by:** CMF Metadata
- **Description:** This parameter specifies how the import tool handles blocks of CMF Metadata in the bundle that are identified to be Duplicates of CMF Metadata in Oracle Enterprise Repository. Duplicate CMF Metadata is always matched by the UUID of the metadata.
- **Values:**
  - **Ignore** - Ignore the duplicate CMF Metadata from the bundle.
  - **Overwrite** - Replace the CMF Metadata in Oracle Enterprise Repository with the duplicate CMF Metadata from the bundle. This overwrites all information for the CMF Metadata.

#### **Standard.saver.metadata.props.duplicate.handling**

- **Used by:** CMF Metadata Entry Types
- **Description:** This parameter specifies how the import tool handles CMF Metadata Entry Types in the bundle that are identified to be Duplicates of CMF Metadata Entry Types in Oracle Enterprise Repository. Duplicate CMF Metadata Entry Types are always matched by the Name of the metadata entry type.
- **Values:**
  - **Ignore** - Ignore the duplicate CMF Metadata Type from the bundle.
  - **Overwrite** - Replace the CMF Metadata Entry Type in Oracle Enterprise Repository with the duplicate CMF Metadata Entry Type from the bundle. This overwrites all information for the CMF Metadata Entry Type.

### **5.2.2.2 Sample parameter.properties File**

A sample `parameter.properties` file with all options included is as follows:

```
##Name: parameters.properties
##Location: import_bundle.zip/META_INF/

##Legend:
## a double ## indicates a comment
## a single # indicates a command which can be used in an import bundle
## the text between []'s is the set of valid values for the parameter with comma
separating individual values

## for assets:
## This parameter will retain the UUIDs of assets in the bundle rather than
randomly generating new UUIDs
#PreserveUUIDs=[true,false]
#Standard.saver.asset.props.duplicate.handling=[Create New Version, Ignore,
Overwrite]
```

```
#Standard.saver.asset.props.duplicate.matching=[UUID, NameVersion]

## for assettypes:
#Standard.saver.assettype.props.duplicate.handling=[Create New Version, Ignore,
Merge, Overwrite]
#Standard.saver.assettype.props.duplicate.matching=[UUID, Name]

## for relationship types:
#Standard.saver.relationshiptype.props.duplicate.handling=[Ignore, Overwrite]

## For CMF metadata
#Standard.saver.metadata.props.duplicate.handling=[Ignore, Overwrite]

## For CMF metadata entry types
#Standard.saver.metadataentrytype.props.duplicate.handling=[Ignore, Overwrite]
```

### 5.2.3 The Default Import Behavior

If the parameter `.properties` file does not exist in a `.zip` bundle, the default behavior on Import is as follows:

#### Asset Type

- An asset already exists, if an asset with the same name and version is in Oracle Enterprise Repository.
- Assets, which already exist, are then evaluated on the UUID.
  - Assets, which have the same name, version, and UUID, are ignored.
  - Assets, which have the same name and version, but a different UUID yield a new asset with a random UUID, same version, but a name that is versioned with a suffix of a "V" and a number (for example, Service Asset V2).
- An asset which does not already exist is created with the name and version specified in the import bundle. The UUID is randomly generated.
- When both assets in a relationship are included in the import bundle, the relationships are created.

#### Acceptable Value List

- An acceptable value list already exists if an acceptable value list with the same name is in Oracle Enterprise Repository.
- Acceptable Value lists, which already exist, are merged. The values, which appear in the import bundle, but not in Oracle Enterprise Repository are added to Oracle Enterprise Repository. The values in Oracle Enterprise Repository are not modified.
- An acceptable value list, which does not already exist, is created with the same name and contents specified in the import bundle.

#### Categorization Type

- A categorization type already exists if a categorization type with the same name is in Oracle Enterprise Repository.
- Categorization Types, which already exist, are merged. Categorization items, which appear in the import bundle, but not in Oracle Enterprise Repository are added to Oracle Enterprise Repository. Categorization items in Oracle Enterprise Repository are not modified.

- A categorization type, which does not already exist, is created with the same name and contents specified in the import bundle.

#### Relationship Type

- A relationship type already exists if a relationship type with the same name is in Oracle Enterprise Repository.
  - Relationship Types, which have the same name and direction, are ignored.
  - Relationship Types, which have the same name and a different direction, yield a new relationship type with the bundle specified direction and a name that is versioned with a suffix of a "V" and a number.
- A relationship type, which does not already exist, is created with the same name and direction specified in the import bundle.

#### Metadata Entries

- A metadata entry already exists if a metadata entry with the same UUID is in Oracle Enterprise Repository.
- Metadata Entries, which already exist in Oracle Enterprise Repository, are overwritten by the Metadata Entry in the Import Bundle.
- A Metadata Entry which does not already exist is created with the same UUID specified in the import bundle.

#### Metadata Entry Type

- A metadata entry type already exists if a metadata entry type with the same UUID is in Oracle Enterprise Repository.
- Metadata Entry Types, which already exist in Oracle Enterprise Repository, are overwritten by the Metadata Entry Type in the Import Bundle.
- A Metadata Entry Type, which does not already exist, is created with the same UUID specified in the import bundle.

## 5.3 Starting the Import/Export Tool

The Import/Export Tool can be launched from within Oracle Enterprise Repository from the Admin screen. The Import/Export Tool requires JDK 1.4.2 or later.

### 5.3.1 Launching from Oracle Enterprise Repository

The Import/Export Tool is launched from the Admin tab. Due to the nature of the Import/Export Tool and the amount of data required for a comprehensive backup, the tool is restricted to users who are assigned the Admin role, such as the built-in admin user. The link to launch the Import/Export tool is visible only to users with the Admin role.

The Import/Export section on the Admin tab is enabled and configured by a property in system settings. If the section is not visible for a user with the Admin role, then verify that the following system setting is enabled and set to True:

```
cmee.importexport.enabled
```

---



---

**Note:** Since Import/Export is a self-contained tool utilizing REX, you are required to log in again after launching it from Oracle Enterprise Repository.

---



---

### 5.3.2 Launching from Outside of Oracle Enterprise Repository

You may need to use the Import/Export tool when you aren't logged in to Oracle Enterprise Repository locally. You can save the Import Export tool and run it from your local machine.

To run the Import/Export utility from a local machine:

1. Follow the instructions in [Section 1.10, "Install Java Web Start"](#) to configure Java Web Start to run on the system
2. Click **Import Export** in the Admin sidebar.
3. Right click **Import/Export Client**, select **Save File** or your browser's equivalent, and then save the `impexp.jnlp` file
4. Move the `impexp.jnlp` file to the location from which you want it to run.
5. Run the `impexp.jnlp` file.

---



---

**Note:** You can also run the Import Export tool by saving it locally, and then using the `javaws` executable, located in `<JAVA_HOME>/jre/bin`. For example:

```
<JAVA_HOME>/jre/bin/javaws -verbose /path_to_file/impexp.jnlp
```

---



---

### 5.3.3 Initial Startup

Upon launching the Import/Export Tool, you may be prompted to enter connection information for Oracle Enterprise Repository, as shown in Figure 5–1.

---



---

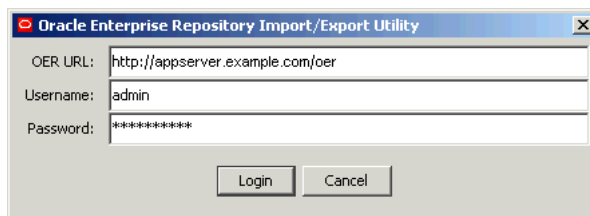
**Note:** You are not always prompted for credentials, except in the SSO environment.

---



---

**Figure 5–1** Oracle Enterprise Repository Import/Export Utility



The Oracle Enterprise Repository Import/Export utility, showing the OER URL, the Username, and Password, with the Login and Cancel buttons.

\*\*\*\*\*

- OER URL
  - Path to the Oracle Enterprise Repository installation, usually the same as the servlet path, as defined in the system settings. Example: `http://example.com/oer/`



- Username  
Name of a user who has been granted the Admin role within Oracle Enterprise Repository.
- Password  
Password for the user specified above.

To set a maximum time in milliseconds that an Import/Export tool can run before it is automatically killed by the job monitor, a new system setting must be configured:

```
cmee.extframework.impexp.monitor.maxruntime
```

## 5.4 Export Items from Oracle Enterprise Repository

This procedure is performed on the Oracle Enterprise Repository Admin screen.

1. Click **Import Export** in the Admin sidebar.
2. Click **Import/Export Client**. The Import/Export Client is displayed.
3. Click the **Export** tab, as shown in Figure 5–2.

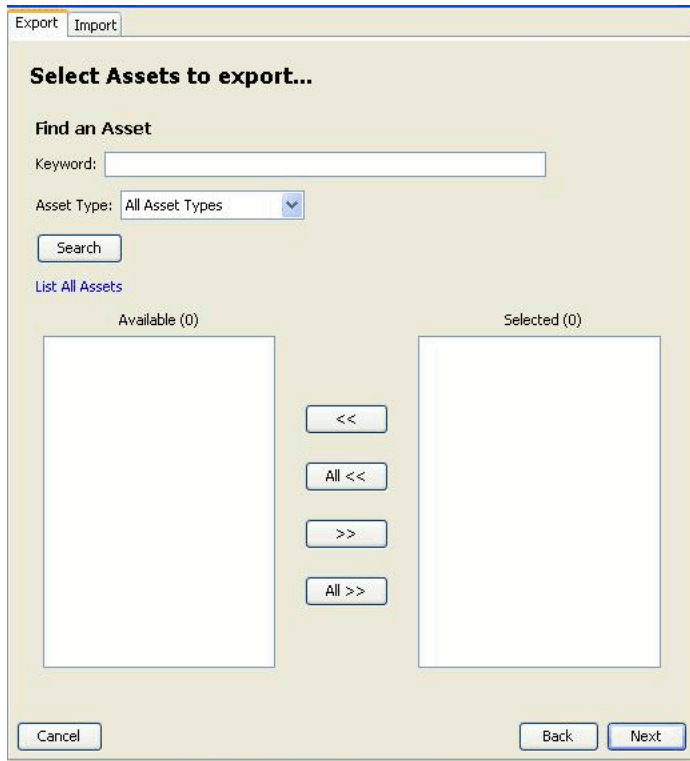
**Figure 5–2** *Export Tab*

The Export Tab, which asks you to select target and entity types to export. At the top is a list box to browse for target files, and in the middle a list of entities that you can select to export.

\*\*\*\*\*

4. Enter an appropriate file name in the **Select Target File** field or click **Browse** to select the directory into which the target file will be exported. Click **Next**.
5. Use the **Select Entities to Export** to identify the entities to be exported. This selection determines the specific behavior of the export client in subsequent steps.
6. The **Select Assets to Export** dialog is displayed, as shown in Figure 5–3.

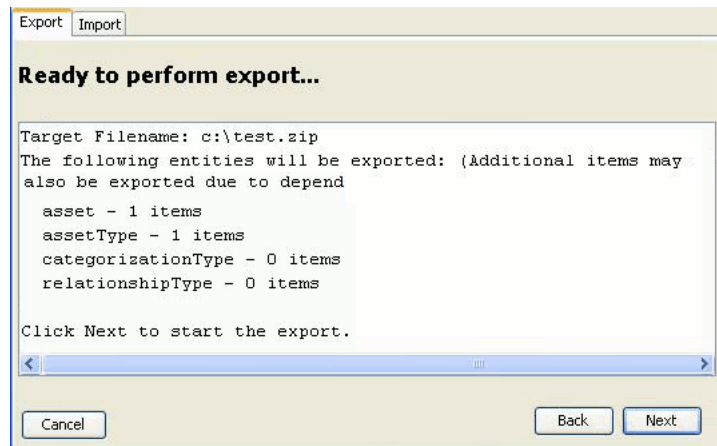
**Figure 5–3 Select Assets to Export Dialog**



The Export Tab, prompting you to select assets to export. Towards the top of the page is Find an asset, where you can enter a keyword to search for, a box for you to select an asset type, and Available and Selected columns for assets to be selected.

\*\*\*\*\*

7. Use **Search** or click **List All** to display a list of items in the **Available** column.
8. Use the arrow buttons to move selected items between the **Available** and the **Selected** columns.
9. Click **Next**. If more than one selection was made in Step 5 (Select Entities to Export), repeat Step 7 and Step 8 in the next Select dialog to select the next group of items for export.
10. Click **Next**. A list of the files to be exported is displayed in the **Ready to Perform Export** dialog, as shown in Figure 5–4.

**Figure 5-4 Ready to Perform Export Dialog**

The Ready to Perform Export Dialog showing the Target Filename test.zip with asset-1 items, assetType-1 items, categorizationType-0 items, relationshipType-0 items. The Dialog next says Click Next to start the export.

\*\*\*\*\*

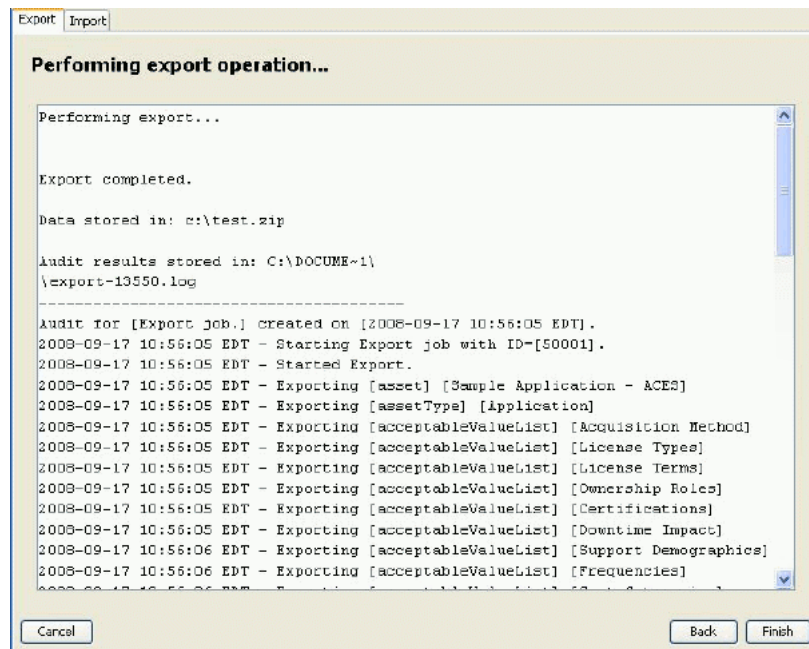
11. Click **Next**. A progress bar indicates the status of the export process.

---

**Note:** Variations in the size and complexity of the exported items affects the speed of this process and the behavior of the progress bar.

---

A summary of the export is displayed on completion, as shown in Figure 5-5.

**Figure 5-5 Performing Export Operation Dialog**

The Performing Export Operation dialog, showing completed export of text.zip, with an audit trail for the export job.

\*\*\*\*\*

On Microsoft Windows platforms, Windows Explorer opens to the folder where the archive was saved (assuming system permissions allow shell execution from the application).

## 5.5 Importing Items into Oracle Enterprise Repository

This procedure is performed on the Oracle Enterprise Repository Admin screen.

---



---

**Note:** This procedure also applied to importing Solution Packs into Oracle Enterprise Repository.

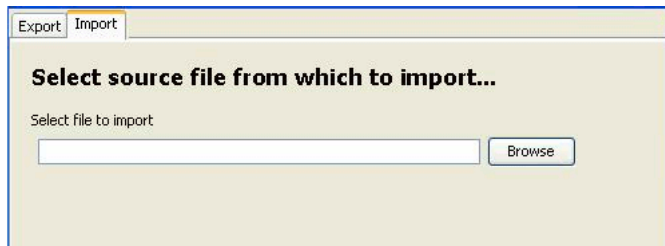
---



---

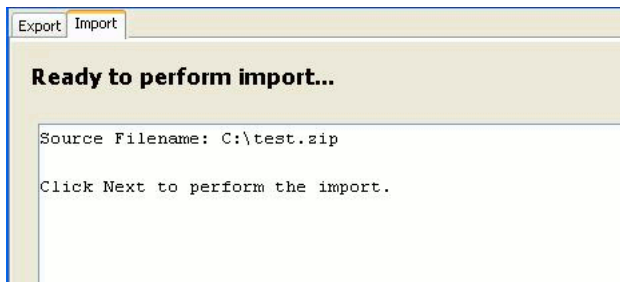
1. Click **Import Export** in the Admin sidebar.
2. Click **Import/Export Client**. The **Import/Export Client** is displayed.
3. Click the **Import** tab.
4. Enter the appropriate file name in the **Select File to Import** field or click **Browse** to select the source file for the import in [Figure 5-6](#).

**Figure 5-6 Select Source File From Which to Import Dialog**



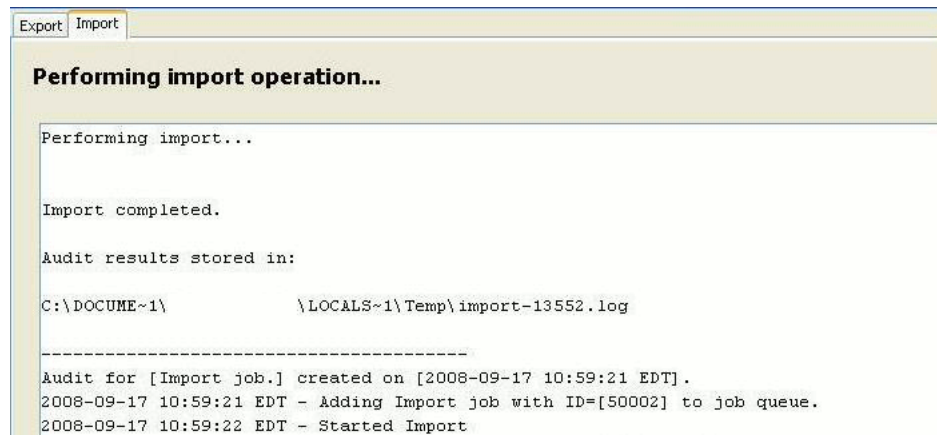
5. Click **Next**. The selected source file is listed in the Ready to Perform Import dialog is displayed, as shown in [Figure 5-7](#).

**Figure 5-7 Ready to Perform Import Dialog**



6. Click **Next**. A progress bar indicates the status of the import process. Variations in the size and complexity of the imported items affects the speed of this process and the behavior of the progress bar.

A summary of the import is displayed on completion, as shown in [Figure 5-8](#).

**Figure 5–8 Performing Import Operation Dialog**

---

**Note:** When running the Import/Export tool, importing larger data sets may cause available memory issues. To avoid, save the `impexp.jnlp` file locally by right clicking on the Import/Export Client link and selecting "Save link as..." option from the context menu. Edit the `impexp.jnlp` file to increase the default `max-heap-size` amount to better accommodate the size of the data sets being imported. If the machine importing the data set has enough available memory, change the `max-heap-size` parameter to a higher value, as in the following example: `<j2se version="1.6" max-heap-size="1024m" />`

---



---

---

## Configuring Reporting with BI Publisher

This chapter describes how to configure the Business Intelligence (BI) Publisher for reporting with Oracle Enterprise Repository.

---

---

**Note:** A restricted-use license for Oracle BI Publisher is included with OER. This license allows you to use the reports included with OER. If you want to create your own custom reports, you must acquire a license for the full Oracle Business Intelligence Enterprise Edition.

---

---

This chapter contains the following sections:

- [Section 6.1, "Deploy Oracle Enterprise Repository Pre-configured BI Publisher Content"](#)
- [Section 6.2, "Understanding the Security Model"](#)
- [Section 6.3, "Configuring the Oracle Enterprise Repository JDBC Data Source"](#)
- [Section 6.4, "Configuring the BI Publisher Server - Cache Expiration"](#)
- [Section 6.5, "Configuring the Oracle Enterprise Repository System Settings to Enable Reporting"](#)
- [Section 6.6, "Working with Reports"](#)

### 6.1 Deploy Oracle Enterprise Repository Pre-configured BI Publisher Content

Oracle Enterprise Repository is the entry point for all the reports. To host reports, you must configure BI Publisher and Oracle Enterprise Repository to interact with each other.

---

---

**Note:** These instructions assume that BI Publisher 11g is installed, which is dedicated only for hosting Oracle Enterprise Repository reports.

---

---

BI Publisher uses a repository internal to BI Publisher to host the users, roles, reports, and configuration for BI Publisher. This repository is replaced by the BI Publisher content which has been specifically developed for the integration with Oracle Enterprise Repository.

## 6.1.1 Configure BI Publisher with Oracle Enterprise Repository

---

---

**Note:** Do not attempt to migrate your 10g reports to 11g.

---

---

To deploy the Oracle Enterprise Repository-BI Publisher content, perform the following steps:

1. Find a packaged content archive for your database. You can find the reports in the Oracle Enterprise Repository Deployment in the `<ORACLE_HOME>/oer/modules/tools/solutions/reports` directory.  
In this directory you will find **12.1.3.0.0-OER-BIP11gReports.zip**. It contains three folders corresponding to each database. Each folder contains only the OER reports.
2. Stop your BI Publisher application server.
3. Copy the OER folder from the appropriate database folder into the BI Publisher Reports directory:  
`domain/config/bipublisher/repository/Reports`.
4. Restart your BI Publisher application server.
5. Check your BI Publisher configuration by opening the web console `http://hostname:port/xmlpserver`. Verify that the **OER** folder appears in the Shared Folders directory.

## 6.2 Understanding the Security Model

BI Publisher uses Enterprise Manager to manage users and roles rather than its own security model. For more information about the security model, see the *Oracle Fusion Middleware Administrator's Guide for Oracle Business Intelligence Publisher*.

### 6.2.1 The Administrator User

The Administrator user and password are determined during installation. This user and password are required to setup the database connection to OER as well as configure the OER user and role, and to setup the data sources in BI Publisher.

---

---

**Note:** In BI Publisher, the password is case sensitive.

---

---

### 6.2.2 The OER User and OER\_REPORTS Role

Use the BI Publisher WLS console to create an OER user. Apply the BI groups to this user. Verify the user account by logging into the BI Publisher console with the OER user credentials. View the reports in the Shared Folders->OER folder.

### 6.2.3 Set Local Time Zone for Report Footer

For the correct time zone to display on report footers, set the time zone in BI Publisher for the OER User account.

- Sign in as the OER User and select my account link to modify the Time Zone setting, located on the Signed In As `<username>` link at the upper right section of the BI Publisher console.



## 6.3 Configuring the Oracle Enterprise Repository JDBC Data Source

To update the JDBC information, perform the following steps:

1. Log into BI Publisher as an administrator.
2. Click the **Admin** tab.
3. In the Data Sources section, select OER - the **JDBC Connection** link.

---

**Note:** This is case-sensitive.

---

4. In the Update Data Source screen, change the connection string to point to the database for Oracle Enterprise Repository.

You can select either of the following Driver types for Oracle Database depending on your Oracle database version. The JDBC library file (JAR) is installed as part of the BI Publisher Server installation therefore no extra Jar file is required.

- Oracle 11g
- Oracle 12c

Example for Oracle JDBC Connection:

---

<b>Database Driver Class</b>	oracle.jdbc.OracleDriver
<b>Connect String</b>	jdbc:oracle:thin:@[HOST]:[PORT]:[DB_NAME]
<b>Example</b>	jdbc.oracle.thin:@machine.example.com:1521:oer

---

5. Click the **Test Connection** button to confirm that BI Publisher can access the database.
6. Click **Apply** to save the changes.

## 6.4 Configuring the BI Publisher Server - Cache Expiration

By default, all Oracle Enterprise Repository reports uses Document Caching to reduce database roundtrips on the generation of report data. While the data is cached, the rendered report may not always show the most recent activities in Oracle Enterprise Repository. Setting the cache expiration changes depending on how frequently BI Publisher needs to refresh the data for the second and subsequent rendering of a report. The default value in BI Publisher for the cache expiration is 30. To change this value, follow the following steps:

1. Log into BI Publisher as an administrator.
2. Click the **Admin** tab.
3. In the System Maintenance section, select the **Server Configuration** link.
4. In the Caching section, specify a Cache Expiration (in minutes) and click **Apply** apply the changes.

---

**Note:** When reports are rendered by BI Publisher, the result is dependent upon the data found for the report. Until data is present, the reports returned by BI Publisher may be inconsistent.

---

## 6.5 Configuring the Oracle Enterprise Repository System Settings to Enable Reporting

Oracle Enterprise Repository contains a report page with a series of links to a set of reports. These reports are hosted by a BI Publisher instance, which must be installed and configured to host the reports. All of the links in the Oracle Enterprise Repository Reports page are incomplete following an Oracle Enterprise Repository installation. There is a system setting that provides the protocol, hostname, port, web application name, and parent directory for all of the report links. The system setting must be configured prior to running any reports.

### Configure system settings

1. Log into Oracle Enterprise Repository as a user with the System Administrator or the Admin role.
2. In Oracle Enterprise Repository Administration page, click **System Settings**.
3. In the Search text box, enter `report` to filter the list of system settings.
4. In the Reports section, set **Enable Reports** (`cmee.reports.enabled`) to True.
5. Set the **Report Server URL** (`oracle.reports.server.url`) to `http://<HOSTNAME>:<PORT>/xmlpserver/OER`, where hostname and port match the BI Publisher installation.

If the Report Server URL property is not properly configured, then the Reports page displays the following message:

*Please set the system setting for Report Server URL. Reports cannot be run until this has been properly set.*

6. Set the system setting Report Server Endpoint URL (`oracle.reports.server.endpoint.url`) to `http://<HOSTNAME>:<PORT>/xmlpserver/services/PublicReportService` to enable OER users to access BI Publisher reports without needing an extra set of credentials.

OER users' access to reports is set via Basic Access Settings, as described in [Section 1.4, "Access Settings."](#)

7. Set the system settings Report Server Username and Report Server Password to the username and password of the OER BI Publisher account created in [Section 6.2.2, "The OER User and OER\\_REPORTS Role."](#)
8. Save the system settings.
9. Validate that the system settings are configured properly by navigating to the Reports section.
10. Run any report.

If the browser does not retain an active session to BI Publisher, then the BI Publisher's Login screen is displayed. After you have successfully logged into BI Publisher, the report is displayed.

Running the report opens a new browser window to the URL for BI Publisher. Comparing the URL generated by Oracle Enterprise Repository to the URL generated by BI Publisher should be an aid in diagnosing any configuration errors of the Report Server URL System Setting.

- Repeat the steps mentioned in this section, until the report runs successfully. If the report returns with "No Search Results Found", then run some keyword searches in the Assets tab of Oracle Enterprise Repository.

---

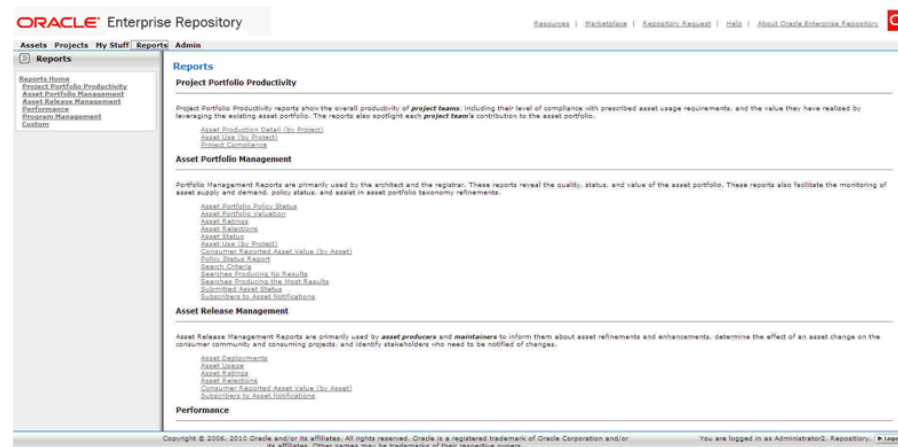
**Note:** If Internet Explorer returns a different report than the one requested, then change the Temporary Internet Files to check for changes on Every Visit.

---

## 6.6 Working with Reports

This section discusses the Oracle Enterprise Repository reports that are available in the Reports page, as shown in [Figure 6-1](#).

**Figure 6-1 Oracle Enterprise Repository Reports Page**



### 6.6.1 Understanding Report Types

This section describes the Oracle Enterprise Repository report types.

#### Project Portfolio Productivity Reports

Project Portfolio Productivity reports show the overall productivity of project teams, including their level of compliance with prescribed asset usage requirements, and the value they have realized by leveraging the existing asset portfolio. The reports also spotlight each project team's contribution to the asset portfolio. Project portfolio productivity reports include:

- Asset Production Detail (by Project)** (requires the asset production tracking feature to be enabled)

Shows the individual assets used in a project, as well as the hours saved by using the asset. Managers can use this report to reward groups or individuals for reusing assets; architects and registrars can use the report to spot emerging standards in a portfolio. Managers can also use this report to quantify the value produced for clients.

- Asset Use (by Project)**

Shows individual assets used in a project, as well as the hours saved by using the asset. Managers can use this report to reward groups or individuals for reusing assets; architects and registrars can use the report to spot emerging standards in a

portfolio. Managers can also use this report, with project leads, to quantify value produced for clients.

- **Project Compliance (assumes use of Compliance Templates)**

Shows the Compliance Templates applied to projects and whether the project is in compliance with the template.

### **Asset Portfolio Management Reports**

Used primarily by architects and registrars, these reports show the quality, status, and value of the asset portfolio. These reports also monitor asset supply and demand, as well as assist with asset portfolio taxonomy refinements. The reports included within this group are as follows:

- **Asset Portfolio Policy Status**

Shows a summary of the policy status (pass, fail, unknown) for an organization's entire asset portfolio. The report then enables users to view a list of assets for each status (for example, all of the assets that are failing). The detailed reports list all policies applied to each asset, and the status of all policy assertions.

- **Asset Portfolio Valuation**

Shows the number of assets used, how frequently each asset was used and the total savings in reuse time. Architects and registrars can use the report to find the most valuable and valued assets.

- **Asset Ratings**

Shows consumer ratings for assets. Architects and registrars can use the report to chart the perceived value of assets, while producers and maintainers can use the report as a feedback mechanism.

- **Asset Rejections**

Shows the reasons why an asset was rejected for use in a project. Architects and registrars can use this report to gauge the level of demand for specific assets, while producers and maintainers can use the report as a feedback mechanism.

- **Asset Status**

Shows the asset name, type and status (active, inactive, and so on). Registrars can use this report to determine the level of asset lifecycle maintenance required for the asset portfolio.

- **Asset Use (by Project)**

Shows individual assets used in a project, as well as the hours saved by using the asset. Managers can use this report to reward groups or individuals for reusing assets; architects and registrars can use the report to spot emerging standards in a portfolio. Managers can also use this report, with project leads, to quantify value produced for clients.

- **Consumer Reported Asset Value (by Asset)**

Shows the consumer reported savings associated with an asset, along with the predicted value of the assets. Managers can use this report to refine the numbers used to calculate predicted net hours. Architects and registrars can use this report to identify the most used and valuable assets in the registry. Producers and maintainers can use this report to determine actual asset value as realized by consumers and identify ongoing changes and enhancements to assets in the registry.

- **Policy Status Report**

Shows a summary of all policies in an organization's policy portfolio, and the number of assets that are passing, failing, or have an unknown status. This report also shows a detailed list of the assets that are governed by each policy, and the number of assertions that are passing, failing, or have an unknown status.

- **Search Criteria**

Shows the number of results of each search parameter. Architects and registrars can use this report to define taxonomies.

- **Searches Producing No Results**

Shows the search parameters in a search that produced no results. Ultimately, the report shows areas of unmet demand in the registry. Registrars and architects can use this report to refine taxonomies.

- **Searches Producing the Most Results**

Shows the search parameters in searches that returned the most results. This report can be used by architects and registrars to determine if users are finding required information in the registry. The report can also be used to refine taxonomies.

- **Submitted Asset Status**

Indicates asset status in the approval cycle. Registrars can use this report to determine scheduling for assets requiring work.

- **Subscribers to Asset Notifications**

Shows specific subscribers to asset information. This report can be used by architects and registrars to determine the level of demand on assets in the portfolio.

### **Asset Release Management Reports**

Used primarily by asset producers and maintainers, these reports inform these users about asset refinements and enhancements, determine the effect of an asset change to the consumer community and consuming projects. In addition, these reports identify stakeholders who must be notified about changes. Asset release management reports include:

- **Asset Deployments**

Shows the names of deployed assets and the projects with which they are associated. Used by producers, maintainers, and project leads to assess the effect of change to an asset.

- **Asset Usages**

Lists the assets selected for use in specific projects. Used by producers, maintainers, and project leads to assess the potential impact of changes to an asset.

- **Asset Ratings**

Shows consumer ratings for assets. Architects and registrars can use the report to chart the perceived value of assets, while producers and maintainers can use the report as a feedback mechanism.

- **Asset Rejections**

Lists the reasons an asset was rejected for use in a project. Architects and registrars can use this report to gauge the level of demand for specific assets, while producers and maintainers can use the report as a feedback mechanism.

- **Consumer Reported Asset Value (by Asset)**

Indicates the consumer reported savings associated with an asset, along with the predicted value of the assets. Managers can use this report to refine the numbers used to calculate predicted net hours. Architects and registrars can use this report to identify the most used and valuable assets in the registry. Producers and maintainers can use this report to determine actual asset value as realized by consumers and identify ongoing changes and enhancements to assets in the registry.

- **Subscribers to Asset Notifications**

Shows specific subscribers to asset information. This report can be used by architects and registrars to determine the level of demand on assets in the portfolio.

### **Performance Reports**

These reports allow individual users to quantify the value of their contributions to the organization. They also inform department heads about the performance of the producers, consumers, and project leads in their organizations. The performance reports presently included in Oracle Enterprise Repository include:

- **Asset Use (by project)**

Shows the assets used in a project, as well as the hours saved by using the asset. Managers can use this report to reward groups or individuals for reusing assets; architects and registrars can use the report to spot emerging standards in a portfolio. Managers can also use this report, with project leads, to quantify value produced for clients.

### **Program Management Reports**

Used primarily by various managers, these reports expose the value that the program brings to the organization (ROI) and indicates whether the program successfully meets goals, demonstrates program growth and progress and serves as the basis for incentives and governance. The program management reports include:

- **Active and Unapproved User Login**

Shows the names of active and unapproved users and the number of times that they logged in. The system administrator or security administrator can use this report to reallocate registry licenses.

- **Active and Unapproved Users with Zero Logins**

Shows the names of active and unapproved users who have never logged in. The system administrator or security administrator can use this report to reallocate registry licenses.

- **Asset Portfolio Valuation**

Shows the number of assets used, how frequently each asset was used and the total savings in reuse time. Architects and registrars can use the report to find the most valuable and valued assets.

- **Consumer Reported Asset Value (by Asset)**

Shows the consumer-reported savings associated with an asset, along with the predicted value of the assets. Managers can use this report to refine the numbers

used to calculate predicted net hours. Architects and registrars can use this report to identify the most used and valuable assets in the registry. Producers and maintainers can use this report to determine actual asset value as realized by consumers and identify ongoing changes and enhancements to assets in the registry.

- **Password Event Log**

Shows the password changes made for users. The report can be used by security administrators or system administrators to identify security breaches.

- **Submission Status (by Producer)**

Shows the names of producers, the number of assets submitted and registered, number of use instances, and the estimated time saved by asset consumers.

- **Subscribers to Asset Notifications**

Shows specific subscribers to asset information. This report can be used by architects and registrars to determine the level of demand on assets in the portfolio.

## 6.6.2 Adding Custom Reports

Oracle Enterprise Repository enables you to add custom reports to the Reports page. All custom Reports must be generated by a reporting engine external to Oracle Enterprise Repository and must be exposed by a URL. On the Reports page, the custom report can be given a name, description, and a link to it.

---



---

**Note:** A restricted-use license for Oracle BI Publisher is included with OER. This license allows you to use the reports included with OER. If you want to create your own custom reports, you must acquire a license for the full Oracle Business Intelligence Enterprise Edition.

---



---



---



---

**Caution:** If custom reports are added in this fashion, then the directory containing the reports must be backed up and protected during future upgrades of Oracle Enterprise Repository.

---



---

### 6.6.2.1 Overview

The Oracle Enterprise Repository Reports subsystem is built from a collection of .toc (table of contents), .jod (an XML representation of the internal report GUI), and .xml (Oracle Enterprise Repository Report definition) files. These files reside within the `WEBINF/config/reports` directory of the Oracle Enterprise Repository web application. To construct a custom report, a .toc file must be modified and a .xml file must be created. The .jod files have nothing to do with this customization and should not be modified.

Oracle provides a custom.toc file for the table of contents beneath the Custom section of the Reports page. A customreports.xml file is also included, and which must be copied and named to something for the report.

### 6.6.2.2 Configuration Steps

The steps to configure custom reports in Oracle Enterprise Repository are as follows:

1. Browse to the `WEB-INF/config/reports` directory for your deployed instance of Oracle Enterprise Repository.
2. Copy `customreports.xml` to a new name that represents the report. The extension must be `.xml` (For example, `CustomerReportA.xml`).
3. Edit `customreports.xml` to name it, describe it, and add a link to the report, as follows:
  - The `<report>` element has a parameter `name` that must refer to a unique, short name for the report. The file name is recommended (For example, `<report name='CustomerReportA'>`).
  - The value of the `<displayName>` element is the label shown in the Reports page within the Custom section (For example, `<displayName>Report A</displayName>`).
  - The value of the `<description>` element is the description shown in the Reports page (For example, `<description>This describes Report A.</description>`).
  - The value of the `<externalLink>` element is the URL for the report. This URL should execute the report. (For example, `<externalLink>http://www.example.com/reports/CustomerReportA?cmd=execute</externalLink>`).
4. Save the copied and modified `customreports.xml` file into the `WEB-INF/config/reports` directory. This is where Oracle Enterprise Repository looks for all report files. The report is now created.
5. Edit the `custom.toc` file to add the newly created report to the Reports area of Oracle Enterprise Repository.
  - The `<report>` element specifies the name of the report. Use identical text to what appears in the `.xml` file for the report. The file name is recommended. (For example, `<report name='CustomerReportA'>`)
6. Back up the `custom.toc` and `.xml` files for the report. These copies are required after upgrades since the upgrade resets the files.
7. Restart the application server.

### 6.6.3 Running a Report

This procedure is performed on the Reports page. You must install and configure the Oracle Business Intelligence Publisher (BI Publisher) to run a report.

1. Select the report from the list in the sidebar. The report form opens BI Publisher in a new browser window.
2. Login to the BI Publisher using the BI Publisher user name, which is different from the Oracle Enterprise Repository user name.
3. Select parameter values at the top of the report.
4. Click the calendar icon to launch a calendar to select From and To dates. The dates for each report default to a range of Year-To-Date.
5. Select other parameters as appropriate for the report.
6. Select a **Sort Order**.
7. Click the Template list to select an output format for the report. By default, the HTML template is displayed.



8. Click **View**. The report is generated.



# Part II

---

## Administering the Asset Lifecycle

This part describes advanced administration tasks for the Oracle Enterprise Repository lifecycle.

This part includes the following chapters:

- [Chapter 7, "Using the Type Manager"](#)
- [Chapter 8, "Configuring and Using Automated Harvesting"](#)
- [Chapter 9, "Using the Asset Editor"](#)



---

---

## Using the Type Manager

This chapter describes how to use the Type Manager to control the organization and display of information for each artifact as it appears in the Asset Editor and in the Asset Detail in Oracle Enterprise Repository.

This chapter includes the following sections:

- [Overview](#)
- [Launching the Type Manager](#)
- [Creating a New Type](#)
- [Configuring Tabs to Support Registration Workflow](#)
- [Tab Elements](#)
- [Adding Images for Type Manager Icons](#)
- [Unique ID](#)
- [Common Metadata Viewers](#)
- [Viewing and Editing Type XML Schemas](#)

### 7.1 Overview

Each artifact in Oracle Enterprise Repository falls under one of three general headings:

- Asset
- Compliance templates
- Policy

Each of these headings is represented by a variety of Types:

- Asset types
- Compliance templates types
- Policy types

These types are patterns, various instances of which are created to accommodate the diverse artifacts that can be managed within Oracle Enterprise Repository. For example:

- Application is an instance of an asset type.
- Project profile is an instance of a compliance template type.
- Regulatory compliance policy is an instance of a policy type.

Types determine both the display of information for, and the taxonomy of, all of the artifacts (assets, compliance templates, policies) in Oracle Enterprise Repository. Oracle Enterprise Repository ships with several types. These can be modified, or new types can be created using the Type Manager.

The Type Manager controls the organization and display of information for each artifact as it appears in the Asset Editor and in the Asset Detail.

## 7.2 Launching the Type Manager

This procedure is performed in the Asset Editor.

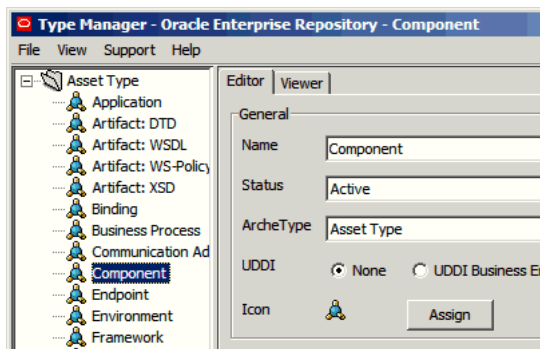
---

**Note:** The Asset Editor requires Java Web Start. See [Section 1.10, "Install Java Web Start"](#) for more information about installing and configuring Java Web Start.

---

1. Select **Manage Types** in the Actions menu. The Type Manager page is displayed, as shown in [Figure 7-1](#).

**Figure 7-1 Type Manager Page**



2. Select an asset type, compliance template, or policy from the tree to view/edit its configuration.

### 7.2.1 Type Manager Views

Each type has an Editor and a Viewer.

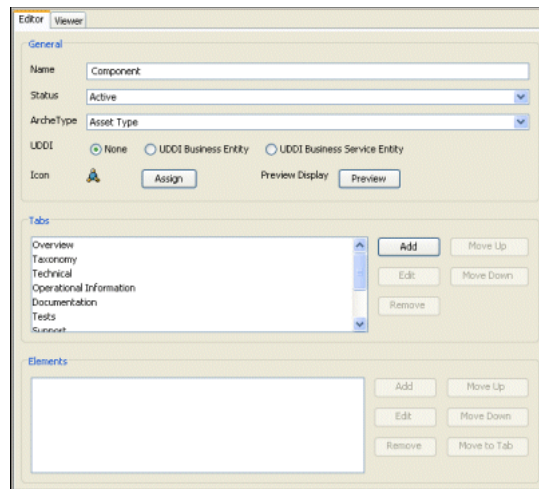
Information relevant to the configuration of the Editor and the Viewer views for each type is stored in XML files in the database. The Type Manager loads the XML files into a GUI, which enables the creation, modification, and deletion of types. After the type is saved, the data is assembled in XML and loaded into the database.

- [Section 7.2.1.1, "The Editor"](#)
- [Section 7.2.1.2, "The Viewer"](#)

#### 7.2.1.1 The Editor

The Editor controls the organization and display of information for assets/compliance templates of the particular type as they appear in the Asset Editor.

**Figure 7-2 Editor Tab**



The Editor display is divided into three sections:

- **General**

Includes the type name, the usage status selector, the type icon selector, and a Preview button, which displays the asset, compliance template, or policy type as seen through the Asset Editor.

- **Tabs**

Includes a list of available tabs for the type. These tabs correspond to tabs displayed in the Asset Editor view of an asset/compliance template. Click any listed tab to display its elements in the Elements section, just below the Tabs section.

- **Elements**

The elements of any tab listed in the Tabs section are displayed here when the item is clicked.

Several buttons appear to the right of the Tabs and Elements sections:

- Add
- Edit
- Remove
- Move Up
- Move Down

These actions can be applied to any item listed in the Tabs or Elements sections. Simply select the item, and then click the appropriate button.

The elements listed below are required for valid Types. They can be renamed, but cannot be removed from a type.

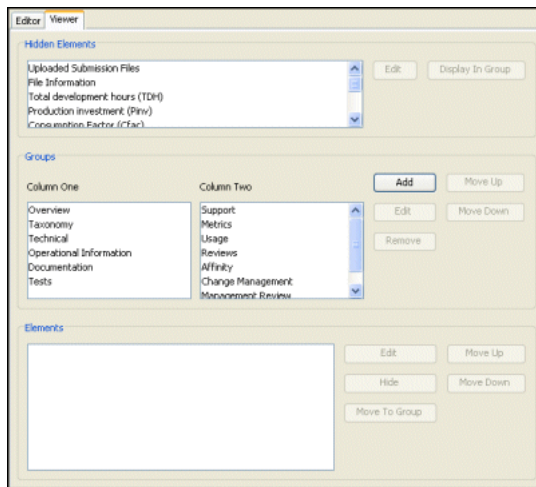
- Name
- Version
- Description
- File Information
- Notification Email

- Keywords

### 7.2.1.2 The Viewer

The Viewer controls the organization and display of information for assets, compliance templates, or policies of the particular type as they appear in the Assets section of Oracle Enterprise Repository (the general user view of asset/compliance template information).

**Figure 7–3 Viewer Tab**



The Viewer display is divided into three main sections:

- Hidden Elements
  - Lists all elements available for display.
  - Use the Edit button (on the right) to change the configuration of any listed element.
  - Use the Display in Group button to assign the element to one of the two columns in the main asset display in Oracle Enterprise Repository.
- Groups
  - Includes Column One and Column Two. These correspond to the two main columns in the main asset display in Oracle Enterprise Repository. Use the buttons on the right to configure the elements in the respective columns.
- Elements
  - Click any item in Groups -- Column One or Groups -- Column Two to display the respective item's elements in the Elements section. Use the buttons on the right to configure these elements.

---

**Caution:** Changes made to Types affect the display of all assets assigned to that type.

---

## 7.3 Creating a New Type

To preserve database integrity, only Types to which no assets are currently associated can be deleted.



---



---

**Note:** This action can alter reuse measurements.

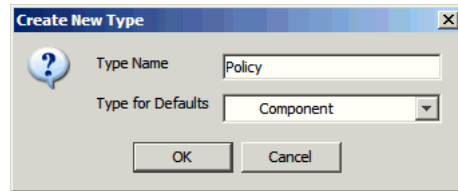
---



---

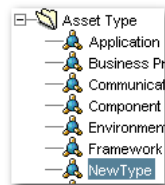
1. Select New in the File menu to open the Create New Type dialog is displayed, as shown in [Figure 7-4](#).

**Figure 7-4 Create New Type Dialog**



2. Enter the appropriate information in the **Type Name** text box.
3. In the Type for Defaults list, select one of the existing Types as the basic template for the new type.
4. Click **OK**. The new type appears in the tree, as shown in [Figure 7-5](#).

**Figure 7-5 Asset Type List**



## 7.4 Configuring Tabs to Support Registration Workflow

Tabs should be positioned on a Type in a logical sequence that accommodates the registration workflow. For example, in the Framework asset type (shipped with Oracle Enterprise Repository), it is intuitive to complete the General information first. The Documentation and Testing tabs, being very similar, are located near each other. The actual registration step, the last in the workflow process, is performed on the Administration tab. For that reason the Administration tab is system generated and intentionally fixed in its position.

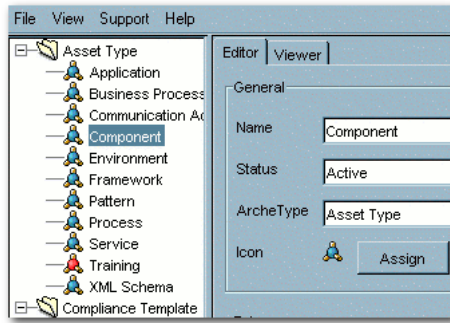
Whenever a tab is approved, a `TabApprovedEvent` event is triggered. This event, along with the tab name and certain other metadata, is sent to the workflows for processing.

### 7.4.1 Adding Tabs

You can add additional tabs in the Type Manager. Perform the following steps:

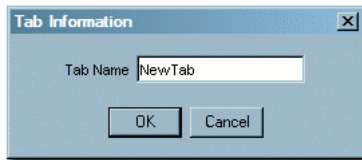
1. Select the Type to be modified, as shown in [Figure 7-6](#).

**Figure 7-6 Asset Type List**



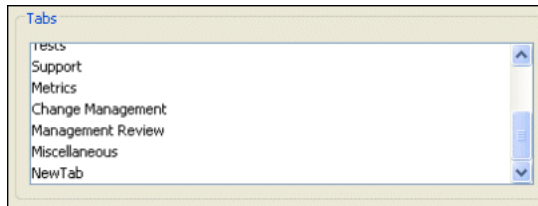
- In the Editor tab, click **Add** in the Tabs section. The Tabs dialog is displayed, as shown in [Figure 7-7](#).

**Figure 7-7 Tab Information Dialog**



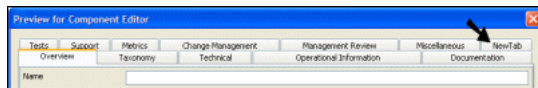
- Enter the appropriate information in the **Tab Name** text box.
- Click **OK**. The name for the new tab appears in the Tabs section list, as shown in [Figure 7-8](#).

**Figure 7-8 Tabs List**



- Click **Preview** in the General section. The preview page is displayed. The new tab is displayed, as shown in [Figure 7-9](#).

**Figure 7-9 Preview for Component Editor**



- Close the Preview page.
- Click **Save** to save the changes.

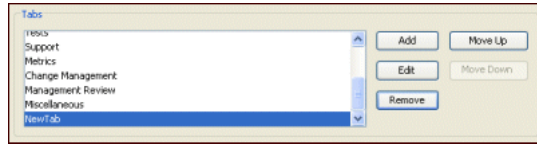
## 7.4.2 Change the Position of a Tab

To change the position of a tab, perform the following steps:

- Select the Type to be modified.

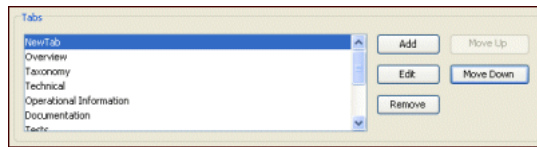
- In the Editor tab, select the tab to be moved.

**Figure 7-10 Tabs List**



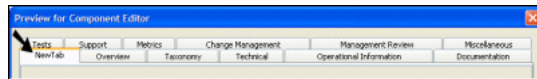
- Click **Move Up** or **Move Down**, as appropriate. Each click moves the tab one level. [Figure 7-11](#) shows the selected tab has moved from the bottom to the top of the list.

**Figure 7-11 Tabs List - Selected Tab at the Top**



- Click **Preview** in the General section. The preview dialog is displayed. Note new position of the tab in [Figure 7-12](#).

**Figure 7-12 Preview of the Component Editor**



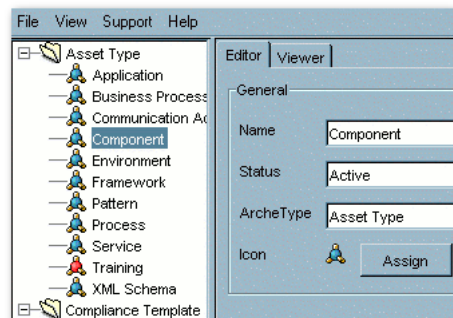
- Click **Save** to save the changes.
- Close the Preview window.
- Click **Save** to save the changes.

### 7.4.3 Editing Tabs

To edit a selected tab, perform the following steps:

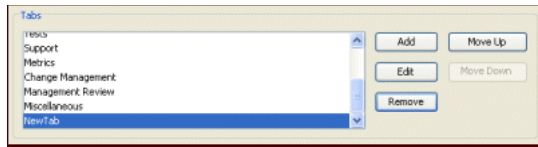
- Select the Type to be modified, as shown in [Figure 7-13](#).

**Figure 7-13 Asset Type List - Tab Selected**



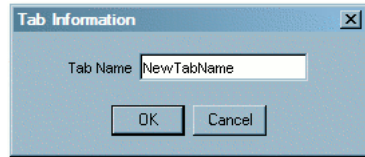
- On the Editor tab, select the tab to be edited, as shown in [Figure 7-14](#).

**Figure 7–14 Tabs List**



3. Click **Edit** and rename the tab. The Tab Information dialog is displayed, as shown in [Figure 7–15](#).

**Figure 7–15 Tab Information Dialog**



4. Click **OK**.
5. Click **Save** to save the changes.

## 7.4.4 Deleting Tabs

---

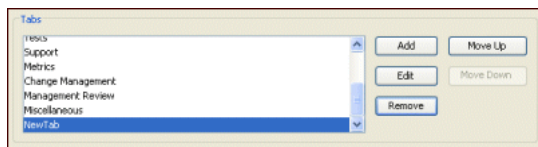
**Note:** Deleting a tab deletes all data associated with the tab.

---

To delete a tab, perform the following steps:

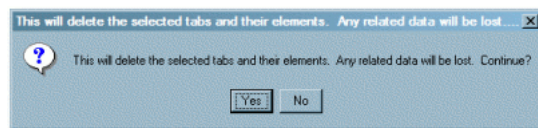
1. Select the Type to be modified.
2. On the Editor tab, select the tab to be deleted, as shown in [Figure 7–16](#).

**Figure 7–16 Tabs List**



3. Click **Remove**. A confirmation dialog is displayed, as shown in [Figure 7–17](#)

**Figure 7–17 Confirmation Dialog**



4. Click **Yes**. The tab is deleted, and no longer appears in the tabs list or in the preview.
5. Click **Save** to save the changes.

All tabs can be deleted from a type, except for those containing any of the following required elements:

- Name
- Version
- Description
- File information
- Notification Email
- Keywords

The system prevents the deletion of any tab containing any of these elements. To delete a tab containing any of these elements the elements must first be moved to another tab.

## 7.5 Tab Elements

A variety of Elements can be added to tabs to accommodate the management and presentation of asset metadata. Some of these Elements are single-use: they are used only once in a type. After they are assigned, single-use elements no longer appear in the list of available elements. The properties for each element type are listed below.

- Text field -- contains data strings
  - Display Name  
Element label that appears in the Asset Editor.
  - Required  
If checked, the field must be populated with data before the asset is saved.
  - Indexed  
If checked, the field appears in More Search Options for that type.
  - Maximum Length  
A validation rule for the maximum length of data in the field; accepts numeric entries.
- Numeric text field -- contains a single numeric value
  - Display Name  
Element label that appears in the Asset Editor.
  - Required  
If checked, the field must be populated with data before the asset is saved.
  - Indexed  
If checked, the field appears in More Search Options for that type.
  - Minimum Value  
A validation rule for the minimum value of data in the field; accepts numeric entries.
  - Maximum Value  
A validation rule for the maximum value of data in the field; accepts numeric entries.
  - Decimal Places

A validation rule for specifying the number of decimal places of data in the field; accepts numeric entries.

- Date field -- contains a single date value
  - Display Name  
Element label that appears in the Asset Editor.
  - Required  
If checked, the field must be populated with data before the asset is saved.
  - Indexed  
If checked, the field appears in More Search Options for that type.
- URL -- contains a string that links to a URL
  - Display Name  
Element label that appears in the Asset Editor.
  - Required  
If checked, the field must be populated with data before the asset is saved.
  - Indexed  
If checked, the field appears in More Search Options for that type.
- Check box -- contains a two-value check box that is either on or off.
  - Display Name  
Element label that appears in the Asset Editor.
  - Required  
If checked, the field must be populated with data before the asset is saved.
  - Checked by Default  
If Yes is selected, the check box is automatically checked on new assets of the same type.
- Lists -- contains a list of finite items
  - Display Name  
Element label that appears in the Asset Editor.
  - Required  
If checked, the field must be populated with data before the asset is saved.
  - Indexed  
If checked, the field appears in More Search Options for that type.
  - Acceptable Values  
List of available data that populates the list; the Configure button enables configuration of the values.
- Editable list -- contains items that are either data or numeric values
  - Display Name  
Element label that appears in the Asset Editor.
  - Required

- 
- Indexed
      - If checked, the field must be populated with data before the asset is saved.
    - Entry Field
      - If checked, the field appears in More Search Options for that type.
    - Acceptable Values
      - List of alphanumeric and numeric values; the Configure enables configuration of either alphanumeric or numeric values.
  - Multiple selection list -- contains data from the configured list from which items can be selected only once
    - Display Name
      - Element label that appears in the Asset Editor.
    - Required
      - If checked, the field must be populated with data before the asset is saved.
    - Indexed
      - If checked, the field appears in More Search Options for that type.
    - Acceptable Values
      - List of available data that populates the list; the Configure button enables configuration of new or existing values.
  - Table -- contains column and row information; columns correspond to elements, and rows correspond to data about the elements; available types include:
    - Text Field
    - Numeric Text Field
    - URL
    - Check Box
    - Drop Down Box
    - Table
      - \* Group Display Name
        - Element label that appears in the Asset Editor
      - \* Item Display Name
        - Data entry area
      - \* Required
        - If checked, the field must be populated with data before the asset is saved.
      - \* Elements
        - Each element is a column of the table and follows the same rules as the element type.
  - Categorization -- contains data relevant to the behavior of views in the Asset Editor; this can be used only once per type
    - Display Name
      - Element label that appears in the Asset Editor.
    - Categorization Type

List of various categorizations available in Oracle Enterprise Repository.

- Asset relationship -- contains data specific to the behavior of relationships; this element type can be used only once per type to search for and establish a relationship between assets
  - Display Name  
Element label that appears in the Asset Editor.
- Associated files -- contains data specific to the behavior of files; adding an associated create a link to the file
  - Display Name  
Element label that appears in the Asset Editor.
- Contact selection -- contains data specific to the behavior of contacts; this element can be used only once per type to enable the creation of a contact or use of an existing contact
  - Display Name  
Element label that appears in the Asset Editor.
- Producing projects -- contains data specific to the behavior of projects that produce assets as well as consume assets; this element can be used only once per type to enable producing projects.
  - Display Name  
Element label that appears in the Asset Editor.
- JAR introspection -- contains data specific to the behavior of JAR files; this element can be used only once per type to add a fully qualified class name to the contents of the element
  - Display Name  
Element label that appears in the Asset Editor.
- Requirement list/coupling -- contains data specific to the behavior of requirements; the element enables coupling based on the data (alphanumeric) entered in the element
  - Display Name  
Element label that appears in the Asset Editor.
- Vendor selection -- contains data specific to the behavior of vendors. This element can be used only once per type and the names correspond to the names of the vendors configured in the Configure Vendors option in the Asset Editor
  - Display Name  
Element label that appears in the Asset Editor.
- Uploaded submission files -- contains data specific to the behavior of uploaded files; this element can be used only once per type to generate data based on the contents of the directory named the same as the asset ID in the upload path
  - Display Name  
Element label that appears in the Asset Editor.
- SFID -- contains data relevant to the behavior of Automated Usage Tracking; this can be used only once per type



- Display Name  
Element label that appears in the Asset Editor.
- Unique Element -- contains data relevant to the behavior of Unique ID; this can be used only once per type
  - Display Name  
Element label that appears in the Asset Editor.
- Project(s) to which template is applied -- contains data relevant to the behavior of Compliance Templates; this can be used only once per type
  - Display Name  
Element label that appears in the Asset Editor.
- UDDI Service Key -- This is used with the Harvester Solution Pack to uniquely identify a service to be published to a UDDI Registry.
  - Display Name  
Element label that appears in the Asset Editor.
- UDDI Registries -- This is used with the Harvester Solution Pack to display a list of UDDI Registries that the asset has been published to.
  - Display Name  
Element label that appears in the Asset Editor.
- XPath Metadata Viewer -- This is the XPath Common Metadata Viewer. For more information, see [Section 7.8.1, "XPath Common Metadata Viewer"](#).
  - Display Name  
Element label that appears in the Asset Editor.
  - Metadata Type  
The Entry Type which contains the XML-Formatted Entry Type.
  - XPath  
The xpath used to find the metadata within the given Metadata Type.
- Introspection Standard Attributes Viewer -- This is used with the Harvester Solution Pack to display the metadata associated with the Harvester operations. It is usually added to asset types in the Harvester Solution Pack as an element called Harvester Properties. This viewer is also an example of the XSLT Common Metadata Viewers. For more details, see [Section 7.8.2, "XSLT Common Metadata Viewer"](#).
  - Display Name  
Element label that appears in the Asset Editor.
- Metadata Files -- This element performs like Asset Files. This element exposes XML-formatted metadata, which is attached to the asset by Harvester or Exchange Utility operations. The metadata is exposed to the Use/Download operations on the Asset. This element cannot be used with the Viewer portion of the type and it never appears in Asset Detail.
  - Display Name  
Element label that appears in the Asset Editor.

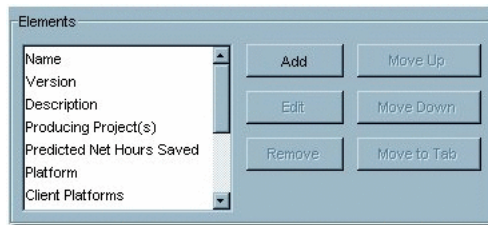
- Asset Files -- This element enables files to be exposed for asset Use/Download operations. Each file can be added by providing a name/url pair to the asset. This element cannot be used with the Viewer portion of the type and it never appears in Asset Detail. It is added to all asset types, by default, as an element called "File Information".
  - Display Name  
Element label that appears in the Asset Editor.
- WSDL Summary -- This is used with the Harvester Solution Pack to provide a link to the WSDL, which originally defined the service.
  - Display Name  
Element label that appears in the Asset Editor.

### 7.5.1 Working with Elements

Numeric Text elements contain numeric information and are often used with reports. The following example adds a Numeric Text element to the Overview tab of a Type to create a reporting field.

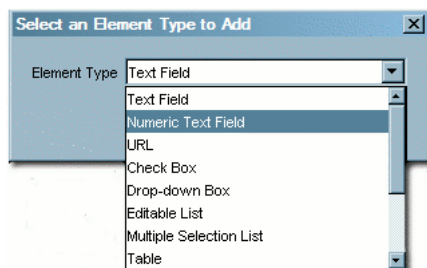
1. Select the Type to be modified.
2. On the Editor tab, click **Add** in the Elements section, as shown in [Figure 7-18](#).

**Figure 7-18 Elements Section**

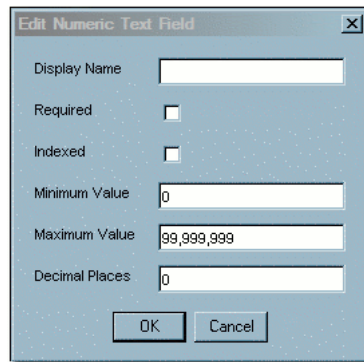


The Select an Element Type to Add dialog is displayed, as shown in [Figure 7-19](#).

**Figure 7-19 Select an Element Type to Add Dialog**



3. Select **Numeric Text Field** from the list.
4. Click **OK**. The Edit Numeric Text Field dialog is displayed.

**Figure 7–20 Edit Numeric Text Field Dialog**

5. Enter the appropriate information in the **Display Name** text box.
6. Checking **Required** means that assets of this type cannot be saved (in the Asset Editor) until the necessary data is supplied for all fields in this element.
7. Check **Indexed** to include the element fields in Oracle Enterprise Repository searches.
8. Enter the appropriate information in the Minimum Value and **Maximum Value** fields.
9. When finished, click **OK**.

## 7.6 Adding Images for Type Manager Icons

You can add images to Type Manager icons. This section includes the following topics:

- [Section 7.6.1, "Icon Image Requirements"](#)
- [Section 7.6.2, "View/Select Available Type Icons"](#)

### 7.6.1 Icon Image Requirements

- Size: 16x16 pixels
- Format: GIF format
- Save to: `images/tree/enterprise/asset` subdirectory in the host directory for Oracle Enterprise Repository installation files.

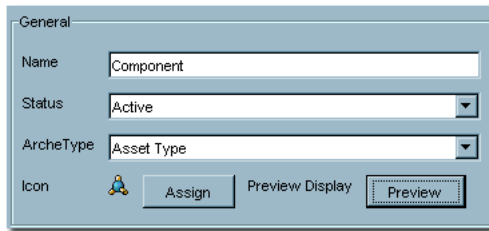
To add icon elements:

1. Locate the `icons.xml` file in the `images/tree/enterprise/asset` subdirectory.
2. Edit `icons.xml` as necessary to add icon elements for the new GIF images.
3. If it is open, close and reopen the Asset Editor.

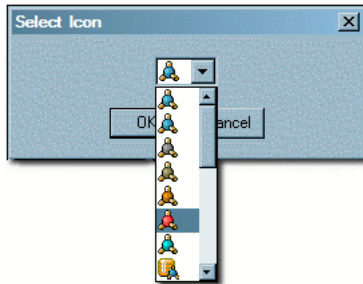
### 7.6.2 View/Select Available Type Icons

This procedure is performed in the Type Manager.

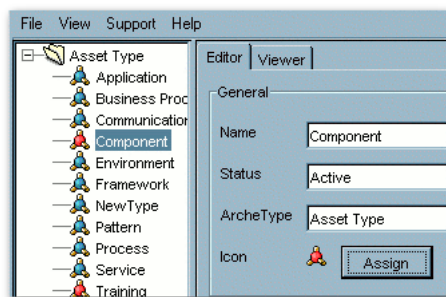
1. Select a Type.
2. On the Editor tab, click **Assign** in the General section, as shown in [Figure 7–21](#).

**Figure 7–21 General Section**

The Select Icon dialog is displayed, as shown in [Figure 7–22](#).

**Figure 7–22 Select Icon Dialog**

3. Select an icon from the list.
4. Click **OK**.
5. Click **Save** to save the changes. The new icon appears next to the Type in the list in the sidebar, as shown in [Figure 7–23](#).

**Figure 7–23 Asset Type List - New Type**

## 7.7 Unique ID

Every asset submitted to Oracle Enterprise Repository is automatically tagged with an Oracle Enterprise Repository-specific ID number, which is used in the organization and management of assets. However, in certain environments or situations it may be advantageous for an organization to assign an additional custom identification key to specific assets. For example, a Unique ID key might be used to enforce uniqueness when relating an asset to a resource or asset in a system external to Oracle Enterprise Repository.

Oracle Enterprise Repository's Unique ID feature enables users with access to the Type Manager to add the Unique ID element to selected asset and/or compliance template

types. Only one Unique ID element can be added per type. This Unique ID element then appears in the Asset Editor display of assets of the selected types.

Users with access to the Asset Editor can assign a Unique ID key to these assets/compliance templates by adding the appropriate information in the Unique ID text box.

The uniqueness of each ID key can be enforced across the entire system, or limited to enforcement within a specific type. For example, when enforced across the entire system, no two assets, regardless of type, can be assigned the same Unique ID key. When enforced within a type, no two assets of the same type can be assigned the same key. However, the same Unique ID can be assigned multiple assets of different types. Within the Asset Editor, any attempt to save an asset with a Unique ID key that has already been assigned to another asset results in an error message.

The specific Type Manager configuration of the Unique ID element determines whether the element and information contained therein is visible to users in the asset detail display for individual assets in Oracle Enterprise Repository.

### 7.7.1 Adding the Unique ID Element to a Type

This procedure is performed in the Type Manager.

---



---

**Note:** The properties associated with the Unique ID element must be enabled in the System Settings section of the Admin screen.

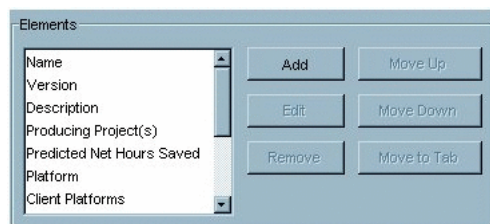
---



---

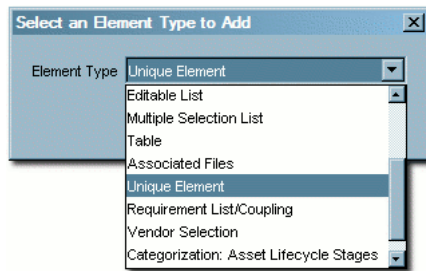
1. Select the Type to be modified.
2. Select the tab to which the Unique ID element is to be added.

**Figure 7-24 Elements Section**

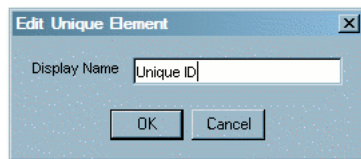


The elements assigned to that tab appear in the Elements section, as shown in [Figure 7-24](#).

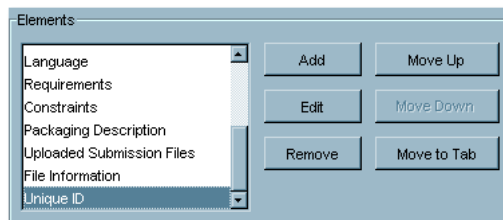
3. Click **Add** in the Elements section. The Select an Element Type to Add dialog is displayed, as shown in [Figure 7-25](#).

**Figure 7–25 Select an Element Type to Add Dialog**

4. Select Unique Element from the list.
5. Click **OK**. The Edit Unique Element dialog is displayed, as shown in [Figure 7–26](#).

**Figure 7–26 Edit Unique Element Dialog**

6. Enter the appropriate information in the **Display Name** text box in the Edit Unique Element dialog.
7. Click **OK**. The Unique ID element now appears in the Elements list, as shown in [Figure 7–27](#).

**Figure 7–27 Elements List - Unique ID**

8. Click **Save**.

## 7.7.2 Information Necessary for Custom Reporting

To include the Unique ID element in a custom report, the schema information for this element is required. This metadata is stored in a table which can be joined to the Assets table and is populated based on the value of certain System Settings. The location of the metadata and rules that control it are specified in this section.

- **Table Name:**  
UniqueElementValues
- **Fields:**
  - assetid: ID of asset (foreign key to assets.id)
  - assettypeid: ID of assettype (foreign key to assettypes.id)
  - value: Unique value (varchar 4000)

---



---

**Note:** A record appears in the `uniqueelementvalues` table only if the asset's Unique ID element (as it appears in the Asset Editor) contains a value.

---



---

### Properties

Two properties are associated with the unique element:

- `cmee.unique-element.enable`
  - Enable/disable field for uniqueness.
- `cmee.unique-element.per-assettype`
  - True = uniqueness enforced within the asset type
  - False = uniqueness enforced across the entire system.

---



---

**Note:** Repeatedly toggling this property from true to false can result in unexpected system behavior.

---



---

### Sample Scenarios

- `cmee.unique-element.per-assettype` is set to true
  - Uniqueness is enforced within the asset type.
    - \* Multiple assets of different asset types can share the same Unique ID.
    - \* Assets of the same asset type cannot share the same Unique ID.
- `cmee.unique-element.per-assettype` is set to false
  - Uniqueness is enforced across the system.
    - \* No two assets, regardless of asset type, can share the same Unique ID.
    - \* Prior Unique ID assignments are unaffected by this property change. However, attempts to save changes to assets can be rejected if the asset shares a Unique ID.

## 7.8 Common Metadata Viewers

There are two elements for Asset Types, Compliance Templates, and Policies that allow users to view XML-formatted metadata that is attached to an asset. One is based on XPath expressions and the other is based on XSLT transformations. After it is added to an asset type, the metadata can be viewed in the Asset Editor and the Asset Detail views. Although the metadata appears in the asset editor, it is for reference only. These viewers do not make the metadata editable.

This section includes the following topics:

- [Section 7.8.1, "XPath Common Metadata Viewer"](#)
- [Section 7.8.2, "XSLT Common Metadata Viewer"](#)

### 7.8.1 XPath Common Metadata Viewer

When added to an asset type you must specify a Common Metadata type and an XPath. When viewing an asset, if the asset has common metadata of that type, the

XPath is evaluated against the common metadata and the resulting items are displayed in the asset as a simple list.

## 7.8.2 XSLT Common Metadata Viewer

These viewers are imported into Oracle Enterprise Repository. After they are imported, the viewers appear as elements that can be added to asset types. There is no configuration of the viewer apart from giving it a name. After they are added to an asset type, assets of that type that contain common metadata compatible with the viewer displays the metadata as defined by the XSLT of the viewer.

## 7.9 Viewing and Editing Type XML Schemas

The Type Manager includes features that allow users to view and edit the XML schemas for Asset Types, Compliance Templates, and Policy Types.

- Export Schema
  - Displays the XML schema for the selected type.
  - Available only to users who are assigned the Admin role.
  - Accessed through the File menu in the Type Manager.
- XML Editor
  - Enables editing of XML source for the Editor and Viewer displays of all types, as seen through the Type Manager.
  - Available to users in the Admin role, and to others, based on specific System Settings.
  - Accessed through the Support menu in the Type Manager.

### 7.9.1 Configuring XML Schemas

This section describes the configuring options available:

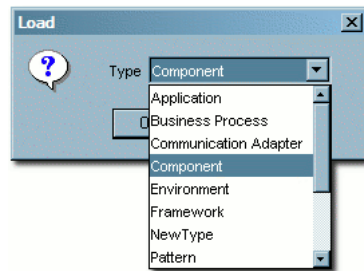
- The Export Schema viewer is available only to users assigned to the Admin role.
- The Support menu displayed is determined by the following conditions:
  - If the user has Admin privileges.
    - \* The XML Editor option is accessible in the Support menu.
  - If the property `cmee.asseteditor.allow.manual.proxy` is enabled.
    - \* The manual proxy configuration option is accessible in the Support menu.

### 7.9.2 Using the Export Schema Feature

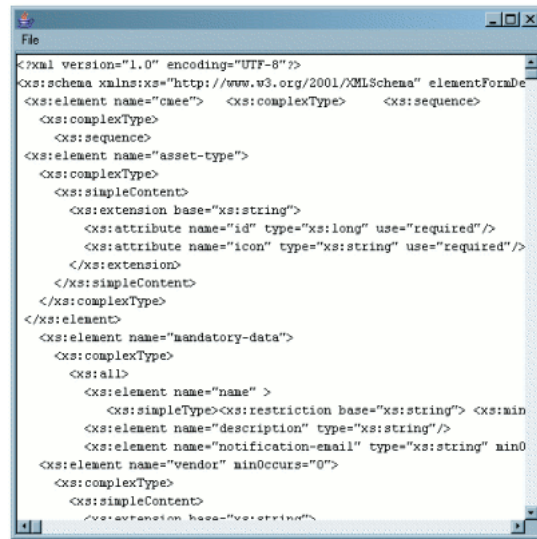
This procedure is performed in the Type Manager.

1. Select **Export Schema** from the File menu. The Load dialog is displayed, as shown in [Figure 7-28](#).
2. Select an XML schema type from the Type list.



**Figure 7-28 Load Dialog**

3. Click **OK**. The XML schema display is displayed, as shown in [Figure 7-29](#).

**Figure 7-29 XML Schema Display**

The display window does not allow editing of the schema. However, the schema can be saved to the local file system for later use. (Click **Save** in the window's **File** menu.)

---

**Note:** The schema can also be copied/pasted into a local document to use for validation of Type XML documents.

---

### 7.9.3 Using the Type XML Editor

The XML Editor enables unvalidated modifications of the XML source for Asset Types, Compliance Templates, and Policy Types.

---

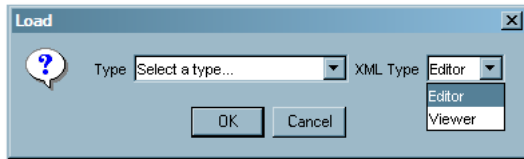
**Caution:** Improper changes to the XML source can render all types inoperable.

---

1. In the Type Manager, select **XML Editor** from the Support menu. The following warning message displays:  
*Changes made in the XML Editor are not validated and can render a Type unavailable!*
2. Click **OK**. A blank dialog is displayed.

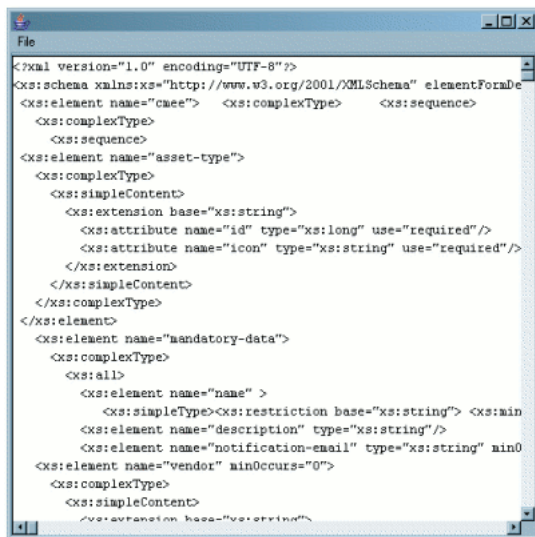
3. Click **OK**. The XML schema display opens.
4. Select **Load** from the File menu in the blank window.
5. In the Load dialog, as shown in [Figure 7-30](#), select the type of XML schema to be edited from the Type list.

**Figure 7-30 Load Dialog**



6. Select **Editor** or **Viewer** from the XML Type list. These selections refer to the Editor and Viewer tabs in the Type Manager.
7. Click **OK**. The selected schema is displayed, as shown in [Figure 7-31](#).

**Figure 7-31 XML Schema Display**



8. Edit the schema as necessary.
9. Save the changes by selecting **Save** from the File menu.

---

---

# Configuring and Using Automated Harvesting

This chapter describes how to get started with Harvester. The tool used to populate Oracle Enterprise Repository is called the Harvester. The Harvester reads metadata from Oracle products and standards-based files. This includes Oracle SOA Suite, Oracle Service Bus, composites generated by Oracle BPA, and standard BPEL, WSDL, XSD, XSLT, BPMN 2.0 and WS-Policy files.

The Harvester automatically creates assets, populates asset metadata, and generates relationship links based on the information in the artifact files.

The files are harvested as they are deployed to the runtime environment. The Harvester can be used from the command line and within Ant and the Weblogic Scripting Tool (WLST).

This chapter contains the following sections:

- [Getting Started with Harvester](#)
- [Configuring the Harvester](#)
- [Harvesting of Artifacts](#)

## 8.1 Getting Started with Harvester

This section describes how to get started with Harvester and its use in various high level use cases.

This section contains the following topics:

- [Section 8.1.1, "Prerequisites"](#)
- [Section 8.1.2, "Harvester Functionality"](#)
- [Section 8.1.3, "Artifacts/Products Version Matrix"](#)

### 8.1.1 Prerequisites

Before using the Harvester, you must perform the following prerequisites:

- Import the Harvester Solution Pack, which includes the required Asset Types and Relationships, into Oracle Enterprise Repository. Import the `<OER_Home>/modules/tools/solutions/12.1.3.0.0-OER-Harvester-Solution-Pack.zip` file.

For more information about importing the Oracle Enterprise Repository solution packs, see *Oracle Fusion Middleware Installation Guide for Oracle Enterprise Repository*.

- If you wish to harvest from Oracle Service Bus, install the harvester to the same `FMW_HOME` to which Oracle Service Bus is installed or to a shared file system accessible to the Oracle Service Bus installation. See [Section 8.2.2.2, "Selecting the Artifacts to Harvest for the Command Line"](#) for more information.

---

**Note:** If you wish to harvest projects from Oracle Service Bus, you must use the `osb-harvest.bat` or `osb-harvest.sh` utilities instead of the usual `harvest.bat` or `harvest.sh` Harvester utilities. This requires Oracle Service Bus 12c to be installed. See [Chapter 8.2.5.3, "Harvesting Web Services from Oracle Service Bus"](#) for more information about the OSB Harvester.

---

- When harvesting assets from deployed applications such as Oracle SOA Suite or Oracle Service Bus, or Web Services, then the application must be deployed on WebLogic 11g or higher.
- For complete details about harvesting from Oracle Application Integration Architecture, see "How to Set Up Oracle AIA Content Harvesting" in *Oracle Fusion Middleware Developer's Guide for Oracle Application Integration Architecture Foundation Pack*.
- Harvester requires Java SDK version 6 or higher.

## 8.1.2 Harvester Functionality

You can use the Harvester to:

- Publish standard SOA artifacts such as SCA Composites, BPEL, WSDL, WADL, XSD, and XSLT, BPMN 2.0 and WS-Policy (from Oracle and other vendors) to Oracle Enterprise Repository.
- Publish WSDL and WADL services as a single API asset rather than the full SCA model, greatly reducing the number of assets to be governed. Harvest a WSDL or WADL with the `-simple_model` harvester option (described in [Table 8-4](#)) to create a single API asset.
- Capture dependencies between the artifacts for impact analysis.
- Handle nested WSDL and XSD files.
- Create abstract Interface assets and concrete deployment / Endpoints.
- Capture BPEL partnerlink dependencies and relating them to Interfaces.
- Capture BPEL PM dependencies to Transformations.
- Store or reference artifact content in Oracle Enterprise Repository for reusability.
- Calculate Software File IDs (SFIDs) for artifacts for duplicate detection.
- Publish artifacts from nested directories, remote servers, zip files, jar files, SOA composites, or Oracle BPEL suitcases.
- Publish artifacts in a transactional fashion.
- Publish artifacts from command line, and from automated Ant build process using the Harvester Ant task.
- Publish Oracle Service Bus project artifacts such as Proxy Service, Business Service, XQuery, MFL, XSLT, WS-Policy.
- Integrate the harvesting process as part of WLST deployment scripts.

- Harvest into SSL enabled SOA server. For more information about how to enable SSL on SOA server, and export and import certificates into a client, see *Oracle Fusion Middleware Security Guide*.

---

**Note:** For more information about importing/exporting certifications, see the Security Guides available at

[http://download.oracle.com/docs/cd/E15523\\_01/index.htm](http://download.oracle.com/docs/cd/E15523_01/index.htm)

---

### 8.1.3 Artifacts/Products Version Matrix

Table 8–1 describes the version matrix of the artifacts/products that is supported with Harvester.

**Table 8–1 Artifacts/Products Version Matrix**

Artifact/Product	Version
BPEL	1.1 and 2.0
WSDL	1.1
WADL	1.1.5
SCA	0.95
JCA Adapter	1.5
XSD	1.0
XSLT	1.0
Oracle BPEL PM	10.1.3.4
Oracle SOA Suite	11g and 12c (12.1.3 and 12.2.1)
WS-Policy	1.2
Oracle Service Bus	11g and 12c (12.1.3 and 12.2.1)
WebLogic Web Services	11g and 12c (12.1.3 and 12.2.1)
BPMN	2.0

---

**Note:** The harvester does not support harvesting end-to-end JSON or Native REST services in Oracle SOA Suite and Service Bus, introduced in the 12.2.1 release. Harvesting of other artifacts from the 12.2.1 releases of these products is supported.

---

## 8.2 Configuring the Harvester

This section describes how you can configure Harvester and use it within an integrated development environment.

This section contains the following topics:

- [Section 8.2.1, "Obtaining the Harvester"](#)
- [Section 8.2.2, "Configuring the Harvester for the Command Line"](#)
- [Section 8.2.3, "Invoking the Harvester Using the Repository.Submit Ant Task"](#)
- [Section 8.2.4, "Invoking Harvester from WLST"](#)

- [Section 8.2.5, "Runtime Harvesting Details"](#)
- [Section 8.2.6, "Performing Optional Harvester Configuration"](#)
- [Section 8.2.7, "Configuring Logging for the Harvester"](#)
- [Section 8.2.8, "Transaction Handling in Harvester"](#)
- [Section 8.2.9, "Advanced Configuration"](#)

## 8.2.1 Obtaining the Harvester

The Harvesters are automatically unzipped into the following directories when installing Oracle Enterprise Repository:

- `<FMW_HOME>/oer/tools/harvester`
- `<FMW_HOME>/oer/tools/osbharvester`

The Harvester is also available in `12.1.3.0.0-OER-Harvester.zip`, which is bundled with the Oracle Enterprise Repository 12c installation, at the following location:

`<FMW_HOME>/oer/modules/tools/solutions/12.1.3.0.0-OER-Harvester.zip`

The OSB Harvester is also available in `12.1.3.0.0-OSB12c-Harvester.zip`, which is bundled with the Oracle Enterprise Repository 12c installation, at the following location:

`<FMW_HOME>/oer/modules/tools/solutions/12.1.3.0.0-OSB12c-Harvester.zip`

This manual refers to `<FMW_HOME>/oer/tools/harvester` or `<FMW_HOME>/oer/tools/osbharvester` as the `<Harvester Home>` directory, depending on which harvester you are running.

[Table 8–2](#) describes the relationship matrix for the product and the corresponding version.

**Table 8–2 Product and Version Matrix for Harvester**

Product and Version	Harvest	IDE Search, Browse, and Consume
Service Bus 11g and 12c (12.1.3 and 12.2.1)	OER 12c	Yes
SOA Suite 11g and 12c (12.1.3 and 12.2.1)	OER 12c	Yes
AIA Foundation Pack 11g	AIA	No
Standards based artifacts (WSDL 1.1, XSD, BPEL, XSLT, BPMN 2.0)	OER 12c	Varies

---

**Note:** The harvester does not support harvesting end-to-end JSON or Native REST services in Oracle SOA Suite and Service Bus, introduced in the 12.2.1 release. Harvesting of other artifacts from the 12.2.1 releases of these products is supported.

---

### parameter.properties Used by Harvester Solution Pack

The `parameter.properties` file determines the behavior of the Harvester Solution Pack during the Import operation into Oracle Enterprise Repository. You must not alter or modify this file in any way. The objects in the Solution pack are imported according to the behaviors mentioned in the list:

- Harvester Solution Pack's `/META-INF/parameter.properties`

```

PreserveUUIDs=true
Standard.saver.assettype.props.duplicate.handling=Merge
Standard.saver.assettype.props.duplicate.matching=UUID
Standard.saver.relationshiptype.props.duplicate.handling=Overwrite
Standard.saver.metadata.props.duplicate.handling=ignore
Standard.saver.metadataentrytype.props.duplicate.handling=ignore

```

- Summary of Harvester Solution pack behavior

- **Assets**

Assets, which already exist in Oracle Enterprise Repository, (same UUID) are merged.

Assets are added with the UUID stated in the import bundle.

- **Asset Types**

Asset Types are considered duplicates if the UUID already exists in Oracle Enterprise Repository.

If new elements and/or tabs have been added to an asset type in the solution pack, then these elements and tabs are added to the existing asset type in Oracle Enterprise Repository.

New asset types in the import bundle are added to Oracle Enterprise Repository.

- **Relationship Types**

Relationship Types, which have the same name and direction, are replaced in Oracle Enterprise Repository with the information in the import bundle.

- **Categorization Types**

New categorization entries in Categorization Types, that have the same name as Categorization Types in Oracle Enterprise Repository, are added to that Type using the information in the import bundle.

New Categorization Types are added to Oracle Enterprise Repository.

- **Acceptable Value Lists (AVLs)**

New values in AVLs that have the same name as AVLs in Oracle Enterprise Repository, are added to that AVL using the information in the import bundle.

New AVLs are added to Oracle Enterprise Repository.

- **CMF Metadata Entry types**

Duplicate Metadata Entry types (same UUID) are ignored. (The entities that are in the solution pack are ignored.)

New Metadata Entry Types are added to Oracle Enterprise Repository.

- **CMF Metadata Entries**

Duplicate Metadata Entries (same UUID) are ignored. (The entities that are in the solution pack are ignored.)

New Metadata Entries are added to Oracle Enterprise Repository.

For more information about parameter.properties, see [Section 5.2.2, "The parameter.properties File"](#).

## 8.2.2 Configuring the Harvester for the Command Line

You can configure the Harvester from the command-line. Organizations can easily bootstrap their Enterprise Repository using the Harvester from the command-line. Command-line harvesting allows organizations to harvest:

- individual files
- directories of files
- remote files
- files residing in artifact stores
- files residing on Oracle MetaData Store (MDS)

This section describes the tasks you need to perform to configure the Harvester for the command-line:

- [Section 8.2.2.1, "Setting Repository Connection Information for the Command Line"](#)
- [Section 8.2.2.2, "Selecting the Artifacts to Harvest for the Command Line"](#)

### 8.2.2.1 Setting Repository Connection Information for the Command Line

Open the XML file `HarvesterSettings.xml` located at `<Harvester Home>` and modify the following XML to point the Harvester to an Oracle Enterprise Repository instance with the correct credentials:

```
<repository>
  <uri>http://localhost:7101/oer</uri>
  <credentials>
    <user>smith</user>
    <password>*****</password>
    //To ensure security, the password must be encrypted.
    //The password encryption tool (encrypt.bat/encrypt.sh) allows you to encrypt
    // the passwords that are stored in the Harvester configuration
    // (HarvesterSettings.xml) file.
  </credentials>
  <timeout>30000</timeout>
</repository>
```

---

---

**Note:** It is recommended that you run the Harvester as a user with the Basic Access Settings for Assets - View, Edit, Accept, and Register.

---

---

### 8.2.2.2 Selecting the Artifacts to Harvest for the Command Line

The Harvester can be run from the command line using the `harvest.bat` or `harvest.sh` utilities.

---

---

**Note:** You must use the OSB Harvester if you are harvesting projects from Oracle Service Bus. The OSB Harvester is installed into the `<FMW_HOME>/oer/tools/osbharvester` directory and can be run using the `osb-harvest.bat` or `osb-harvest.sh` utilities. The OSB Harvester must be installed into the same `FMW_HOME` to which OSB 12c is installed.

See [Section 8.2.5.3, "Harvesting Web Services from Oracle Service Bus"](#) for more information about using the OSB Harvester.

---

---



The `harvest.bat` and `harvest.sh` utilities can be run from the server to which Oracle Enterprise Repository is installed. If you want to run the harvester from a different client, obtain the `12.1.3.0.0-OER-Harvester.zip` file from the `<FMW_HOME>/oer/modules/tools/solutions` directory and extract the zip to any client that has a valid `JAVA_HOME` environment variable set, as discussed in [Table 8-3](#).

The `osb-harvest.bat` and `osb-harvest.sh` utilities can be run from the server to which Oracle Enterprise Repository is installed. If you want to run the harvester from a different client:

- Install Oracle Service Bus into a `<FMW_HOME>` directory on the client machine.
- Ensure that the `JAVA_HOME` environment variable is set on the client machine, as shown in [Table 8-3](#).
- Obtain the `12.1.3.0.0-OSB12c-Harvester.zip` file from the `<FMW_HOME>/oer/modules/tools/solutions` directory on the machine to which Oracle Enterprise Repository is installed. Extract the file to the `<FMW_HOME>/oer/tools` directory on the target machine. The full path to the harvester after extracting the files should be `<FMW_HOME>/oer/tools/osbharvester`. The OSB Harvester must be installed into the same `FMW_HOME` to which OSB 12c is installed.

Before running `harvest.bat`, `harvest.sh`, `osb-harvest.bat`, or `osb-harvest.sh`, ensure that the environment variables mentioned in [Table 8-3](#) are set. In Windows, from a command window, you can type "set X" to see the value of the variable X, and "set X=abc" to set the value of X to "abc".

**Table 8-3 Command Line Script**

Environment Variable	Description
<code>JAVA_HOME</code>	Ensure that the <code>JAVA_HOME</code> environment variable points to an installed java runtime (JRE) or SDK. For Oracle Service Bus introspection, this must be Java version 6 or higher.
<code>BEA_HOME</code>	Ensure that the <code>BEA_HOME</code> environment variable points to the installation directory containing Oracle Service Bus server, only if you plan to harvest projects from Oracle Service Bus. For example, <code>C:\bea</code> .
<code>JAVA_OPTS</code>	Optionally, set your <code>JAVA_OPTS</code> parameter to add any additional java parameters that are necessary. For example, if you need to use an HTTP proxy server, set the value to <code>-Dhttp.proxyHost=www-proxy.yourhost.com -Dhttp.nonProxyHosts="*.yourhost.com  localhost"</code>  See Also: <a href="http://java.sun.com/javase/6/docs/technotes/guides/net/proxies.html">http://java.sun.com/javase/6/docs/technotes/guides/net/proxies.html</a>

[Table 8-4](#) shows the options that can be specified using the Harvester command line utility:

**Table 8-4 Command Line Options for the Harvester**

Harvester Options	Description
<code>-settings&lt;file&gt;</code>	Specifies the configuration settings XML file.
<code>-url &lt;URL&gt;</code>	Specifies the URL of Oracle Enterprise Repository.
<code>-user &lt;OER/OAC User Name&gt;</code>	Specifies the user name of the Oracle Enterprise Repository user.
<code>-asset_version</code>	Version to be applied to all assets created in the model

**Table 8–4 (Cont.) Command Line Options for the Harvester**

Harvester Options	Description
-artifact_store	Specifies the name of Oracle Enterprise Repository artifact store to look in. The artifact store must be created beforehand in the Oracle Enterprise Repository Asset Editor. If specified, the -file argument is resolved relative to the artifact store URL.  When specifying a URI relative to an Artifact Store, the URI must resolve to a file such as a .wsdl or .zip file. The URIs that point to directories are not supported.  For more information about artifact stores, see <a href="#">Section 8.2.3.2, "Specifying Parameters as Nested Elements"</a> .
-file <filename or URL>	Specifies the file or directory to be harvested. This can be a filename or URL to the file.
-file_type <type>	Specifies the file type of the file to be harvested. If not specified, then the type is derived from the file extension. This must correspond to one of the filetypes in the config/plugins folder. By default, the following are supported: .bpel, .mfl, .policy, .wsdl, .xsd, .xquery, .xslt.
-remote_url <URL>	Specifies the running server from which to harvest the remote project, instead of from a file.
-remote_username <username>	Specifies the username to connect to the remote server.
-remote_server_type <type>	Specifies the type of remote server. Remote server could include either of the following: SOASuite, SOASuite11g, OSB, WLS.  <b>Note:</b> Use SOASuite11g when running against a SOA Suite 11g server. Use SOASuite when running against a SOA Suite 12c server.
-remote_project <type>	Specifies the name of remote project to harvest, instead of a file. If omitted, all of the projects on the server are harvested. In the case of Oracle SOA Suite, this should be the name of the composite plus revision, for example, MyComposite_rev1. In the case of WLS, this should be the Application Name, as seen in the WebLogic Administration console and Enterprise Manager.
-soa_partition <type>	Specifies the name of partition for Oracle SOA Suite and by default, it uses the soa "default" partition.
-mds_settings <file>	Specifies the location of an adf-config.xml file to initialize MDS. If omitted, defaults to the adf-config.xml file that comes with Harvester.
-deployment_status <status>	Specifies the deployment status that is set on the created assets. It takes either design-time or runtime value.
-version	Specifies the print version information.
-preview	Specifies if the harvest can be run on a preview mode. If set to true, then runs harvester in preview mode. A detailed information about successes and failures are logged, and no changes are committed to Oracle Enterprise Repository.  If set to false or no value is specified, then runs harvester in production mode. A detailed information about successes and failures are logged, and the changes are committed to Oracle Enterprise Repository.
-producing_project <project>	Will apply all assets created from the harvest to this producing project.  The project must already exist in OER.
-registration_status	Specifies registration status to set on created Assets in OER. Acceptable values are: Unsubmitted, PendingReview, UnderReview, and Registered.
-simple_model	Creates single-asset models when set to true. This requires the API asset type.
-apply_registration_status_to_existing_assets	Applies the registration status to existing assets in the introspected model when set to true.
-categorization_type	Specifies which categorization type to apply to the assets created from the introspection. You must also specify the categorization.

**Table 8–4 (Cont.) Command Line Options for the Harvester**

Harvester Options	Description
-categorization	Specifies which categorization to apply to the assets created from the introspection. You must also specify the categorization type.
-apply_categorization_to_existing_assets	Applies the specified categorization to existing assets in the introspected model when set to true.
-help	Displays the online help for the Harvester command line utility.

An example of a remote harvest is as follows:

```
harvest.bat -remote_url mysoasuiteserver:8001 -remote_username weblogic
-remote_server_type SOASuite -remote_project MyComposite_rev1.0 -soa_
partition department1
```

To ensure security, passwords are not passed via the command line and they must be encrypted. The only way to set passwords for Oracle Enterprise Repository and remote servers is by storing them in the Harvester configuration (HarvesterSettings.xml) file. The password encryption tool (encrypt.bat/encrypt.sh) allows you to encrypt the passwords that are stored in the file. For more information about password encryption, see [Chapter 12, "Password Encryption"](#).

[Figure 8–1](#) shows the command line utility options and online help displayed by the harvest.bat -help command.

**Figure 8–1 Harvest Command Line Utility Options**

```
usage:
harvest <settings !-url -user ! (other params)>
-!settings <file>           : Configuration Settings XML file
-!url <URL>                 : OER/OCB URL
-!user <User Name>         : OER/OCB User Name
-!asset_version <assetversion> : Version to be applied to all assets created in the model
-!artifact_store <store>    : <OER Only> Name of OER Artifact Store to look in.
                             : If specified, the -file argument will be
                             : resolved relative to the
                             : artifact store URL
-!file <filename or URL>    : File or directory to be harvested.
                             : This can be a filename or URL to the file.
-!file_type <type>         : File type of the file to be harvested.
                             : If not specified, it will be derived from
                             : the file extension.
-!remote_url <URL>         : Remote server from which to harvest the
                             : remote project, instead of from a file.
-!remote_username <username> : Username to connect to the remote server
-!remote_server_type <type> : Type of remote server.
                             : One of: SOASuite, OER, WES
-!remote_project <type>    : Name of remote project to harvest from the
                             : remote server. If omitted, all of the
                             : projects on the server will be harvested.
-!soa_partition <type>    : Name of partition for SOASuite.
-!adf_settings_file       : <OER Only> Location of an adf-config.xml file to initialize MDS.
                             : If omitted, will default to the adf-config.xml
                             : that comes with Harvester
-!deployment_status <status> : <OER Only> Deployment status to set on created assets
                             : Must be one of: (design-time, run-time).
-!version <version>       : Print version information
-!preview <preview>      : If true: run harvester in preview mode.
                             : Detailed information about successes and failures
                             : will be logged, and
                             : no changes will be committed to OER.
                             : If false or not specified: run harvester in production mode.
                             : Detailed information about successes and failures
                             : will be logged, and
                             : the changes WILL be committed to OER.
-!producing_project <project> : <OER Only> Will apply all assets created from the harvest to this producing project.
                             : The project must already exist in OER.
-!registration_status : <OER Only> Registration status to set on created Assets in OER.
-!single_model : If true: Creates single asset models. Requires OPI asset type.
-!apply_registration_status_to_existing_assets : <OER Only> Set to "true" to apply the registration status to existing assets in the introspected model.
-!categorization_type : <OER Only> Categorization type to apply to assets created from the introspection. Must also specify the categorization.
-!categorization : <OER Only> Categorization to apply to assets created from the introspection. Must also specify the categorization type.
-!apply_categorization_to_existing_assets : <OER Only> Set to "true" to apply the specified categorization to existing assets in the introspected model.
-!help : Harvester Help.
```

Command line options are required only when harvesting from an Oracle SOA Suite 11g server; the -remote\_server\_type SOASuite11g option must be included if this is the case. See [Section 8.2.5.1, "Harvesting from Oracle SOA Suite Server"](#) for more information.

If the options are omitted, then the Harvester uses the information in the HarvesterSettings.xml file in the <Harvester Home> directory, where harvest.bat resides. If options are specified on the command line, then these override the settings in HarvesterSettings.xml.

To point to the artifacts to be harvested using the HarvesterSettings.xml file in the <Harvester Home>, modify the following XML:

```
<query>
```

```

    <fileQuery>
      <rootDir>C:\samples</rootDir>
      <files>BPEL</files>
    </fileQuery>
  </query>

or

<query>
  <fileQuery>
    <files>http://remote/server/my_generated_wsdl</files>
    <fileType>.wsdl</fileType>
  </fileQuery>
</query>

```

### 8.2.3 Invoking the Harvester Using the repository.submit Ant Task

You can invoke harvester as an Ant Task to ensure that all deployment information is stored in Oracle Enterprise Repository at deployment time. You can use the `repository.submit` Ant task provided with the Harvester to harvest and import metadata into Oracle Enterprise Repository. This task can be defined in the `harvest-tasks.xml` file, which is located in the `<Harvester Home>` directory (the directory into which you unzipped `12.1.3.0.0-OER-Harvester.zip`).

This section contains the following topics:

- [Section 8.2.3.1, "Specifying Parameters for the repository.submit Ant Task"](#)
- [Section 8.2.3.2, "Specifying Parameters as Nested Elements"](#)
- [Section 8.2.3.3, "Running the Harvester from Ant"](#)
- [Section 8.2.3.4, "Using the Third-Party Tasks"](#)

#### 8.2.3.1 Specifying Parameters for the repository.submit Ant Task

[Table 8–5](#) shows parameters that can be specified for the `repository.submit` Ant task in the `harvest-tasks.xml` file:

**Table 8–5 Parameters for the repository.submit Ant Task**

Attribute	Description	Required
<code>repositoryURL</code>	Repository instance to connect to.	Yes, unless specified by a property.
<code>repositoryUsername</code>	Username to log into Oracle Enterprise Repository.	Yes, unless specified by a property.
<code>repositoryPassword</code>	<p>Password to log into Oracle Enterprise Repository.</p> <p>To ensure security, the password must be encrypted.</p> <p>The Oracle Enterprise Repository Web console has a tool to encrypt passwords:</p> <p><code>http://&lt;host&gt;:&lt;port&gt;/&lt;domain&gt;/diag/encryptstrings.jsp</code></p>	Yes, unless specified by a property.
<code>timeout</code>	Number of seconds before calls to Oracle Enterprise Repository will time out.	No. Defaults to 300 (5 minutes).
<code>failOnError</code>	Fails the entire build script if the Oracle Enterprise Repository operation results in an error.	No. Defaults to "true."

**Table 8–5 (Cont.) Parameters for the repository.submit Ant Task**

Attribute	Description	Required
errorProperty	Name of the Ant property to set if the repository operation results in an error. This is only useful if failOnError is set to false. If there is an error, the specified property is set to true. Otherwise, the specified property will remain unset.	No.
debug	Display debug information from the Oracle Enterprise Repository task regardless of the Ant -debug setting. Debug information will also be displayed if you pass the -debug parameter to the Ant runtime.	No. Defaults to "true."
description	Harvester Description to associate with each asset created in Oracle Enterprise Repository. This is visible in the Harvester Properties. For more information about harvester properties, see <a href="#">Section A.7, "Harvester Properties"</a> .	No.
namespace	Namespace with which to prefix each asset created in Oracle Enterprise Repository.	No.
version	Harvester Version to associate with each asset created in Oracle Enterprise Repository. This is visible in the "Harvester Properties."	No.
beaHome	The BEA_HOME directory of an installed Oracle Service Bus server.	Yes, when harvesting Oracle Service Bus project jars.
settingsFile	Location of settings XML file. Must conform to BPEL_Harvester_Settings.xsd. This file configures what Harvester classes are mapped to which file types, and how entity and relationships are mapped to types in Oracle Enterprise Repository.  Optionally, you can specify this attribute or the settingsURL attribute, but not both.	No. Defaults to settings XML that is bundled with the Ant task.
settingsURL	Location of settings XML file. Must conform to BPEL_Harvester_Settings.xsd. This file configures what Harvester classes are mapped to which file types, and how entity and relationships are mapped to types in Oracle Enterprise Repository.  Optionally, you can specify this attribute or the settingsFile attribute, but not both.	No. Defaults to settings XML that is bundled with the Ant task.
preview	This option sets the Harvester to Preview mode. The Harvester displays all assets to be created or modified, however, no transaction is committed. To enable, set value to "true".	No. Defaults to false.

### 8.2.3.2 Specifying Parameters as Nested Elements

#### FileSet

FileSets are used to select sets of files to harvest. One or more fileSets must be specified.

The Harvester examines all the files selected by the fileSet, including files in .zip format (including .zip, .jar, and .ear files, for example).

In the Ant repository.submit task shown in [Example 8–1](#), the Harvester examines all the files and directories under the fileSet directory (the /tmp/components directory) and imports them into Oracle Enterprise Repository.

**URI**

URI identifies a file to introspect. One or more uris can be specified.

This can be an "http:" URL that points to a remote file, or a "file:" URL that points to a local file, or any other URL supported by java.

This can point to a single file, or a file in zip format that includes zips, jars, ears, etc. This could include exported Oracle Service Bus project jars. Oracle Service Bus projects can be exported from Oracle Service Bus Workshop, using the `Export|Oracle Service Bus|Configuration Jar` command.

The `artifactStore` attribute is the name of an Oracle Enterprise Repository artifact store to look in. The artifact store must be created beforehand in the Oracle Enterprise Repository Asset Editor. If specified, then the URI is resolved relative to the artifact store URL. When specifying a URI relative to an Artifact Store, the URI must resolve to a file such as a `.wsdl` or `.zip` file. The URIs that point to directories are not supported.

The `fileType` attribute indicates the file type of the file to be harvested. If not specified, then the type is derived from the file extension. This must correspond to one of the file types in the `config/plugins` folder. By default, the following are supported: `.bpel`, `.mfl`, `.policy`, `.wsdl`, `.xsd`, `.xquery`, `.xslt`.

**Example 8-1 Specifying the Files to Harvest with the Ant repository.submit Task**

```
<repository.submit repositoryurl="http://server.example.com:8080/oer"
    repositoryusername="myuser"
    registrypassword="*****"
registrationStatus="Registered"
applyRegistrationStatusToExistingAssets="true"
    //To ensure security, the password must be encrypted.
    //The password encryption tool (encrypt.bat/encrypt.sh) allows you to encrypt
    // the passwords that are stored in the Harvester configuration
    // (HarvesterSettings.xml) file.
    settingsFile="..MyCustomSettings.xml">
<fileset dir="/tmp/components/">
    <include name="**/*" />
</fileset>
<applyCategorization type="Classification" value="Raw"
applyToExistingAssets="false" />
<applyCategorization type="AssetFunction" value="XML Standards"
applyToExistingAssets="true" />
    <uri filetype=".wsdl">http://remote/server/my_generated_wsdl</uri>
</repository.submit>
```

**Registration Status and Categorizations**

You can specify the registration status for assets created from the Harvester in the `HarvesterSettings.xml` file. Acceptable values are:

- **Unsubmitted**
- **Submitted - Pending Review**
- **Submitted - Under Review**
- **Registered**

---

**Note:** The acceptable values for setting the `registrationStatus` of harvested assets in the `HarvesterSettings.xml` file differ from the acceptable values used when harvesting via the command line. See the `-registration_status` entry in [Table 8-4](#).

---

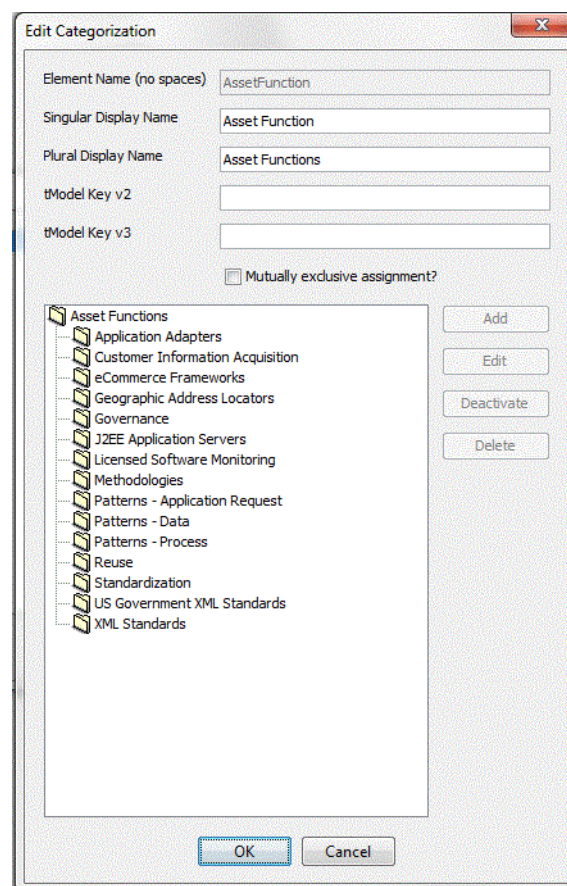


By default, a registration status is applied to newly created assets. You can also specify the registration status for assets that are created from a previous harvest of the same artifact. To enable, set `applyRegistrationStatusToExistingAssets` to **true**, as shown in [Example 8-1](#).

You can apply categorizations to assets created from the harvest. This requires that you know the internal names of the categorization types, which are shown in the Configure Categorizations tab of the Asset Editor. The required value is `Element Name`.

1. Open the Asset Editor.
2. Select **Actions -> Configure Categorizations**.
3. Select the **Categorization Type** and select **Edit**. The **Element Name** field displays the value, see [Example 8-2](#).

**Figure 8-2 Edit Categorizations**



Element Name refers to the field in the Configure Categorizations window in the Asset Editor. Use the value of that field in the `HarvesterSettings.xml`.

Registration Status and Categorizations can also be applied at the command line. The command line parameters are `-registration_status`, `-apply_registration_status_to_existing_assets`, `-categorization_type`, and `-categorization`.

### RemoteProject

`RemoteProject` specifies that the Harvester should read a project from a remote server rather than from a file. The `uri` attribute indicates the running server from which to

harvest the remote project. The username and password attributes indicate login information for the remote server.

The `serverType` attribute indicates the remote server type, which could be either SOASuite or Oracle Service Bus or WebLogic Server. The `projectName` element indicates the name of remote project to harvest, instead of a file. If omitted, all of the projects on the server are harvested. In the case of Oracle SOA Suite, this should be the name of the composite plus revision, for example, `MyComposite_rev1`. In the case of WLS, this should be the Application Name, as seen in the WebLogic Administration console and Enterprise Manager.

### **Example 8–2 Specifying Parameters as Nested Elements - RemoteProject**

```
<repository.submit repositoryurl="http://server.example.com:8080/oer"
    repositoryusername="myuser"
    registrypassword="*****"
    //To ensure security, the password must be encrypted.
    //The password encryption tool (encrypt.bat/encrypt.sh) allows you to encrypt
    // the passwords that are stored in the Harvester configuration
    // (HarvesterSettings.xml) file.
    settingsFile="..MyCustomSettings.xml">
  <remoteProjects uri="http://mywlsserver:7001" username="admin" password="*****"
  serverType="SOASuite" soaPartition="${partiton}">
    //To ensure security, the password must be encrypted.
    //The password encryption tool (encrypt.bat/encrypt.sh) allows you to encrypt
    // the passwords that are stored in the Harvester configuration
    // (HarvesterSettings.xml) file.
    <projectName>MySOAComposite1_rev1.0</projectName>
    <projectName>MySOAComposite3_rev2.0</projectName>
  </remoteProjects>
</repository.submit>
```

### **Producing Projects**

You can specify multiple projects tagged as producing projects for all assets that are created from the harvest. The projects must already exist in OER.

Within the `<repository.submit>` element, include:

- `<producingProject name="MyProjectName"/>`
- `<producingProject name="MyOtherProjectName"/>`

Producing projects may also be set within the `HarvesterSettings.xml` file, in the same way.

#### **8.2.3.3 Running the Harvester from Ant**

To import the Harvester Ant tasks, include a line, as follows, in your Ant XML:

```
<taskdef file=${harvester.dir}/harvest-tasks.xml/>
```

where `harvester.dir` is the `<harvester home>` directory to which you unzipped 12.1.3.0.0-OER-Harvester.zip.

When running from the command line, ensure that the Harvester libraries are available to Ant's classpath. Harvester includes a script called `runant` that sets up the ant classpath correctly. This script must be used to launch ant when using the harvester ant tasks. For example:

```
runant -f mybuild.xml
```



(where `mybuild.xml` is your ant build XML).

Any arguments passed to `runant` are passed along to `ant`. Before running `harvest.bat` or `harvest.sh`, ensure that the required environment variables are set. Refer to [Table 8–6](#) for the list of environment variables that you need to set.

**Table 8–6 Command Line Script**

Environment Variable	Description
JAVA_HOME	Ensure that the <code>JAVA_HOME</code> environment variable points to an installed java runtime (JRE) or SDK. For Oracle Service Bus introspection, this must be Java version 6 or higher.
ANT_HOME	Ensure that the <code>ANT_HOME</code> environment variable points to an installation of Apache Ant, version 1.6.2 or higher. For Oracle Service Bus introspection, ensure that the Ant version is 1.6.5 or higher.
BEA_HOME	Ensure that the <code>BEA_HOME</code> environment variable points to the installation directory containing Oracle Service Bus server, if you plan to harvest projects from Oracle Service Bus. For example, <code>C:\bea</code> .
JAVA_OPTS	Optionally, set your <code>JAVA_OPTS</code> parameter to add any additional java parameters that are necessary. For example, if you need to use an HTTP proxy server, set the value to <code>-Dhttp.proxyHost=www-proxy.yourhost.com -Dhttp.nonProxyHosts=*.yourhost.com  localhost</code>  See Also: <a href="http://java.sun.com/javase/6/docs/technotes/guides/net/proxies.html">http://java.sun.com/javase/6/docs/technotes/guides/net/proxies.html</a>

### 8.2.3.4 Using the Third-Party Tasks

The `runant` script uses `ant`'s *old* launcher: `org.apache.tools.ant.Main`. The *new* launcher uses a `URIClassLoader` interface, which interferes with the usage of custom `URIStrreamHandlers` of the Oracle Service Bus.

The old `ant` launcher does not support automatic discovery of custom `ant` tasks in the `ant/lib` directory. When defining custom tasks with `<taskdef>`, you must specify a `classpath` attribute.

For example, the following is an example of how the `<taskdef>` should NOT be defined:

```
<taskdef id="ant-contrib"
resource="net/sf/antcontrib/antcontrib.properties"/>
```

The following is an example of how the `<taskdef>` should be defined:

```
<taskdef id="ant-contrib"
resource="net/sf/antcontrib/antcontrib.properties"
classpath="${ant.home}\lib\ant-contrib.jar"/>
```

## 8.2.4 Invoking Harvester from WLST

WLST (WebLogic Scripting Tool) is a command-line scripting interface that system administrators can use to manage WebLogic Server instances. WLST supports Oracle Service Bus 10g and SOA Suite 11g.

For more information about WLST, see *Oracle Fusion Middleware WebLogic Scripting Tool Command Reference*.

You can invoke `harvester` from WLST to ensure that all deployment information is stored in Oracle Enterprise Repository at deployment time. Perform the following steps to invoke `harvester` from WLST:

1. Copy the `oer.py` file from the harvester installation folder to the `<BEA_HOME>\wlserver\common\wlst\lib` directory.
2. In the command window, enter the following commands:
  - Initialize the WLST classpath: run `<BEA_HOME>\wlserver\server\bin\setWLSEnv.cmd`
  - Initialize the Harvester classpath: cd to the `<harvester>` folder and run `<harvester>\setenv.bat`
  - Invoke WLST: `java weblogic.WLST<scriptname>`
3. To invoke WLST from ant, perform the following steps:
  - In the command prompt, change to the `<harvester>` directory by using the `cd` command.
  - Enter `runant.bat -f<antscriptname>`, where the ant script invokes the `weblogic.WLST java` class using ant's `<java>` task.
  - The `<java>` task should set `fork = true`.
  - The `<java>` task should include the environment's CLASSPATH (`runant.bat` will set CLASSPATH to include the harvester classes)
  - The `<java>` task should also include on the classpath any libraries referenced by the WLST script.

The sample ant script is as follows:

```
<property environment="env"/>
<target name="harvest">
<java classname="weblogic.WLST" fork="true">
<arg line="\${domain.import.script}"/>
<arg line="\${import.config.file}"/>
<classpath refid="extra.class.path"/> <!--extra jars used by WLST script, e.g.
OSB jars-->
<classpath path="\${env.CLASSPATH}"/>
</java>
</target>
```

The `oer.harvest()` method takes one argument: a dictionary of command line parameters. These are the same parameters that are available from the harvester command line.

For more information about using the harvester from command line, see [Section 8.2.2.2, "Selecting the Artifacts to Harvest for the Command Line"](#).

A sample usage is as follows:

```
argMap = {}
argMap['-harvester_home'] = '.'
argMap['-bea_home'] = 'c:/bea'
argMap['-remote_url'] = 'http://mywlserver:7001'
argMap['-remote_username'] = 'admin'
argMap['-remote_password'] = '*****'
//To ensure security, the password must be encrypted.
//The password encryption tool (encrypt.bat/encrypt.sh) allows you to encrypt
// the passwords that are stored in the Harvester configuration
// (HarvesterSettings.xml) file.
argMap['-remote_server_type'] = 'SOASuite'
argMap['-remote_project'] = 'MySOAComposite_rev1.0'
argMap['-soa_partition'] = 'department1'
oer.harvest(argMap)
```

The `-harvester_home` argument must point to the location of the harvester installation. The `-bea_home` argument is required for Oracle Service Bus harvesting.

## 8.2.5 Runtime Harvesting Details

As assets move from development to testing, staging, and production environments, you may want to harvest these assets into Oracle Enterprise Repository, so that Oracle Enterprise Repository has the most up-to-date endpoints. Oracle Enterprise Repository associates new endpoints to the existing assets in Oracle Enterprise Repository. If the WSDL is modified as the asset moves through its lifecycle, the updated WSDL is also associated with the existing asset in Oracle Enterprise Repository.

This section describes the runtime harvesting details for different servers. This section contains the following topics:

- [Section 8.2.5.1, "Harvesting from Oracle SOA Suite Server"](#)
- [Section 8.2.5.2, "Harvesting Web Services from WebLogic Server"](#)
- [Section 8.2.5.3, "Harvesting Web Services from Oracle Service Bus"](#)
- [Section 8.2.5.4, "Harvesting Deployed Services"](#)

### 8.2.5.1 Harvesting from Oracle SOA Suite Server

You need the following WLS security roles when harvesting from Oracle SOA Suite server:

- Admin
- Operator
- Monitor

The `-remote_url` parameter should point to the port of the soa-infra managed server. The default value of this in Oracle SOA Suite is 8001.

The soa-infra managed server must be up and running. Harvesting connects to the MDS database as part of the remote harvesting. The MDS database must be running and accessible from the machine where harvesting is taking place.

The SSL/HTTP protocol is supported. Harvester connects to the server MBeans via the t3 protocol.

The harvester automatically saves the value of `-remote_url` in the `Deployment URI` property, in the assets harvested from the runtime servers.

An example for harvesting from Oracle SOA Suite 12c server is as follows:

```
harvest.bat -remote_url mysoasuiteserver:8001 -remote_username weblogic
-remote_server_type SOASuite -remote_project MyComposite_rev1.0 -soa_
partition default
```

An example for harvesting from Oracle SOA Suite 11g server is as follows:

```
harvest.bat -remote_url mysoasuiteserver:8001 -remote_username weblogic
-remote_server_type SOASuite11g -remote_project MyComposite_rev1.0 -soa_
partition default
```

To ensure security, passwords are not passed via the command line and they must be encrypted. The only way to set passwords for Oracle Enterprise Repository and remote servers is by storing them in the Harvester configuration

(HarvesterSettings.xml) file. The password encryption tool (encrypt.bat/encrypt.sh) allows you to encrypt the passwords that are stored in the file. For more information about password encryption, see [Chapter 12, "Password Encryption"](#).

The Oracle SOA Suite server has also implemented partitions to further categorize composites during runtime. This enables you to select the partition to use, along with the composite name, in the harvester. The parameter name in the HarvesterSettings.xml file is called soaPartition. A sample HarvesterSettings.xml file is as shown below:

```
<remoteQuery>
  <serverType>SOASuite</serverType>
  <projectName>MyComposite_rev1.0</projectName>
  <uri>http://remotehost:8001/</uri>
  <credentials>
    <user>weblogic</user>
    <password>password</password>
  </credentials>
  <soaPartition>partition_name</soaPartition>
</remoteQuery>
```

---

**Note:** For Oracle SOA Suite 11g, you must set the <serverType> element to SOASuite11g. Additionally, you must include the -remote\_server\_type SOASuite11g command line argument when harvesting from Oracle SOA Suite 11g server.

---



---

**Note:** Partitions should be used only with SOA servers of version 11g R1 PS2 or later. You should comment the **soaPartition** element, if you are using a version earlier to PS2.

---

By default, the Harvester harvests using the partition\_name option set to "default". If it is not specified, then partition "default" comes out-of-the-box by SOA deployments.

---

### Harvesting SOA Suite Composites in a Cluster

The OHS server is deployed as a part of the WebTier label into the Middleware Home and is associated with the domains configured within this Middleware Home, as well.

There is a number of URI's that would normally be assigned to the OHS server's mod\_wl\_2x.so HTTP server module, some of which can be seen below (taken from the OHS mod\_wl\_ohs.conf file and modified in order to make it generic for external consumption):

#### LoadModule weblogic\_module

```
"/opt/Oracle/Middleware/wlserver_10.3/server/plugin/linux/x86_64/mod_wl_22.so"
<IfModule weblogic_module>
  ConnectTimeoutSecs 10
  ConnectRetrySecs 2
  Debug ALL
  WLogFile /tmp/weblogic.log
  DebugConfigInfo ON
  WLSocketTimeoutSecs 2
  WLIOTimeoutSecs 300
  Idempotent ON
  FileCaching ON
```

```

        KeepAliveSecs 20
        KeepAliveEnabled ON
        DynamicServerList ON
        WLProxySSL OFF
    </IfModule>

    <Location /soa-infra>
        SetHandler weblogic-handler
        WebLogicCluster slc01ese.example.com:23049,wls02.example.com:15980
    </Location>
    <Location ~ "/bea_wls_internal/iiop/Client*">
        SetHandler weblogic-handler
        WebLogicCluster wls01.example.com:23049,wls02.example.com:15980
    </Location>
    <Location ~ "/bea_wls_internal/HTTPClnt*">
        SetHandler weblogic-handler
        WebLogicCluster wls01.example.com:23049,wls02.example.com:15980
    </Location>
    <Location /console>
        SetHandler weblogic-handler
        WebLogicHost wls01.example.com
        WebLogicPort 19617
    </Location>
    <Location /em>
        SetHandler weblogic-handler
        WebLogicHost wls01.example.com
        WebLogicPort 19617
    </Location>
    <Location /bea_wls_internal>
        SetHandler weblogic-handler
        WebLogicHost wls01.example.com
        WebLogicPort 19617
    </Location>
    <Location /jndi>
        SetHandler weblogic-handler
        WebLogicHost wls01.example.com
        WebLogicPort 23049
    </Location>
    <Location /HTTPClnt>
        SetHandler weblogic-handler
        WebLogicHost wls01.example.com
        WebLogicPort 23049
    </Location>

```

These special URI patterns allow for JMS and JMX requests to be routed through the OHS WebLogic module.

Other than any changes that you had applied to code, these are the only customizations that had been applied to the SOA clustered installation.

### 8.2.5.2 Harvesting Web Services from WebLogic Server

You do not need any WebLogic Server security roles when harvesting from WebLogic Server.

The `-remote_url` parameter should point to the port of the WLS admin server. The default value of this in Weblogic is 7001. The WLS admin server must be up and running. The SSL/HTTP protocol is supported. Harvester connects to the server MBeans via the t3 protocol.

The harvester automatically saves the value of `-remote_url` in the `Deployment URI` property, in the assets harvested from the runtime servers.

### 8.2.5.3 Harvesting Web Services from Oracle Service Bus

You need the Admin WLS security role when harvesting Web Services from Oracle Service Bus.

You must use the OSB Harvester to harvest web services from an Oracle Service Bus 12c server. The OSB harvester is installed in the `<FMW_HOME>/oer/tools/osbharvester` directory.

The OSB Harvester can be run from the command line using the `osb-harvest.bat` or `osb-harvest.sh` utilities. See [Section 8.2.2, "Configuring the Harvester for the Command Line"](#) for information about configuring the OSB Harvester.

You can use the harvester that ships with Service Bus 11g to harvest non-simple assets from Service Bus 11g servers. The following note describes the process for harvesting simple model API assets (resulting from setting the `simple_model` property, as described in [Table 8-4](#)) from a Service Bus 11g server.

---



---

**Note:** When harvesting simple model assets from a Service Bus 11g server using the `osb-harvest.bat/.sh` utility, the Service Bus harvester must be run from the same Middleware home into which Service Bus 12c is installed. Even though the Service Bus server is 11g, the Harvester requires the Service Bus 12c libraries. There are multiple ways to accomplish this; the easiest way is to copy the Service Bus harvester to an existing Service Bus 12c server.

If Service Bus 12c is installed on a separate client or in a separate Middleware home, you can update the `setenv.sh` or `setenv.bat` script to account for this location.

---



---

The `-remote_url` parameter should point to the port of the WLS admin server for the Oracle Service Bus domain. The default value of this in Weblogic is 7001. The WLS admin server must be up and running. The SSL/HTTP protocol is supported. Harvester connects to the server MBeans via the `t3` protocol.

The harvester automatically saves the value of `-remote_url` in the `Deployment URI` property, in the assets harvested from the runtime servers.

An example of harvesting Web Services from Oracle Service Bus is as follows:

```
osb-harvest.bat -remote_url myosbserver:7001 -remote_username weblogic
-remote_server_type OSB -remote_project MyOSBProject
```

To ensure security, passwords are not passed via the command line and they must be encrypted. The only way to set passwords for Oracle Enterprise Repository and remote servers is by storing them in the Harvester configuration (`HarvesterSettings.xml`) file. The password encryption tool (`encrypt.bat/encrypt.sh`) allows you to encrypt the passwords that are stored in the file. For more information about password encryption, see [Chapter 12, "Password Encryption"](#).

### 8.2.5.4 Harvesting Deployed Services

In addition to harvesting APIs from SOA Suite and Service Bus, you can also harvest deployed services using a URL to a deployed WSDL file. The example below will harvest a Weather service from the internet:

```
./harvest.sh -file
"http://example.com/WeatherWebService/Weather.asmx?WSDL"
```

This method can be used to harvest secure endpoints from Oracle API Gateway (OAG). To get the WSDL URL from OAG, make sure the OAG Policy Studio publishes its WSDL to clients by checking "Allow the Gateway to publish WSDL to clients" checkbox on the WSDL tab in the Service Handler for the web service.

## 8.2.6 Performing Optional Harvester Configuration

You can optionally modify these additional configuration settings in the XML file `HarvesterSettings.xml` in the `<Harvester Home>` directory:

- `<harvesterDescription>`: A description about the harvesting performed. This information is stored in the **Harvester Properties** of the assets created in Oracle Enterprise Repository
- `<harvesterVersion>`: A version of the harvesting performed. This information is stored in the **Harvester Properties** of the assets created in Oracle Enterprise Repository.
- `<namespace>`: A namespace that is added to abstract (non-artifact) Oracle Enterprise Repository assets that are created during harvesting. The namespace is used in duplicate detection. If left empty, then this is set based on information from Oracle SOA Suite and Oracle Service Bus projects when available. That is, generally, the best practice, so override this with caution.
- `<workDir>`: A temporary directory where the zip and jar files are unzipped. By default, the system temp directory is used.
- `<triggerEvent>`: The state of the Oracle Enterprise Repository events for use by Workflow. If true, then trigger Oracle Enterprise Repository events for use by Workflow.

For more information about the harvester properties, see [Section A.7, "Harvester Properties"](#).

### 8.2.6.1 Oracle Metadata Service URLs

Harvester supports reading files from Oracle's MDS (Metadata Service). MDS is used in SOA Suite 11g to store shared data such as SOA Suite framework schemas, shared policies, and deployed composites.

Oracle MDS files are referenced through the `oramds:` URL protocol. Harvester contains the `adf-config.xml` file, which is used to configure the connection to MDS. This file can point to a local MDS store, as used for framework schemas in JDeveloper. It can also point to a remote MDS store, to access deployed composites, shared policies, and framework schemas. The `adf-config.xml` file that ships with harvester contains examples of both usages.

For local MDS:, modify the `metadata-path` property in `adf-config.xml` and set it to match the location of your JDeveloper installation. For remote MDS:, comment the section marked `file store (JDev)` and uncomment the section marked "DB store (SOA Suite server)" in `adf-config.xml`. Also, modify the `jdbc-url`, `jdbc-userid`, and `jdbc-password` properties to point to the MDS database of your Oracle SOA Suite server.

You can also harvest from command line, as mentioned in the example below:

```
harvest.bat -file oramds:/policies/oracle/wsmtom_policy
```

Oracle JDeveloper 11g is a browser-based application that allows you to view the contents in MDS. In JDeveloper, click **File, New**. In the New Gallery dialog, click **Connections** and then **SOA-MDS Connection** to view the MDS connections.

Alternatively, harvester can be pointed to a different `adf-config.xml` file, through the `<mdsSettingsFile>` element in `HarvesterSettings.xml` or in the Harvester ant task.

For more information about configuring MDS, see Oracle ADF XML Files in the *Oracle Fusion Middleware Fusion Developer's Guide for Oracle Application Development Framework*.

### 8.2.6.2 Classpath URLs

Harvester supports reading files from the java classpath. These files are referenced through the classpath: URL protocol. These files are located inside any jar file or directory in Harvester's classpath. For example, `classpath:/META-INF/wsdl/ServiceException.wsdl`.

## 8.2.7 Configuring Logging for the Harvester

The Harvester uses log4j for logging the detailed tasks performed and the log file is placed in the `<Harvester Home>` directory. The logging options can be changed by updating the `log4fl.properties` file located in the `<Harvester Home>` directory.

## 8.2.8 Transaction Handling in Harvester

By default, the harvester makes all of its changes to Oracle Enterprise Repository in a single transaction. Transactional operations in the harvester function is based on the following rules:

- Oracle Enterprise Repository 12c servers only supports one transaction at a time. If a transaction cannot be started, then the harvester informs you about it.
- There is a set of timeouts associated with the transactions. These will terminate a harvester operation if the timeouts are exceeded.
  - The setting, `cmee.extframework.impexp.monitor.rex.maxidle`, specifies the maximum amount of time the transaction is kept alive if it loses connectivity with the client. This allows the transaction to be cancelled if the client-side harvester process is killed.
  - The setting, `cmee.extframework.impexp.monitor.maxruntime`, specifies the maximum amount of time that the entire transaction will take.

It is possible that a single operation will exceed the `cmee.extframework.impexp.monitor.rex.maxidle` (`maxidle`) setting and cause all subsequent operations to fail. When this occurs, the last error in the log will state that *'An error occurred while attempting to rollback because a transaction has not been started.'* This is because the transaction automatically rolls back when the statement exceeds the `maxidle` and the transaction is no longer available when the client application attempts a forceful rollback, having detected errors. The result is that no data, from the harvester operation, is added to Oracle Enterprise Repository.

## 8.2.9 Advanced Configuration

This section contains the advanced configuration options for Harvester:



## Adapters and Applications

Harvester contains a configuration file called `adapters.xml`, that governs which Adapter and Application assets are supported. It has the following structure:

- **adapter**

```
<adapter name="AQ Adapter"
  jcaSpecBinding="oracle.tip.adapter.aq.inbound.AQDequeueActivationSpec"/>
<adapter name="AQ Adapter"
  jcaSpecBinding="oracle.tip.adapter.aq.outbound.AQEnqueueInteractionSpec"/>
<adapter name="PeopleSoft Adapter"
  jcaSpecBinding="com.ibi.afjca.cci.IWAFInteractionSpec"
  jcaCSAdapterName="PeopleSoft"/>
<adapter name="PeopleSoft Adapter"
  namespace="http://xmlns.oracle.com/Enterprise/" />
<adapter name="PeopleSoft Adapter" namespace="http://peoplesoft.com/" />
```

- **adapterAsset**

```
<adapterAsset name="AQ Adapter">
</adapterAsset>
<adapterAsset name="PeopleSoft Adapter" applicationName="PeopleSoft
  Application">
</adapterAsset>
```

Each `adapterAsset` element must correspond to an Adapter asset and/or Application asset that exists in Oracle Enterprise Repository. The Harvester does not create Adapter assets or Application assets. These must be imported via an Oracle Enterprise Repository solution pack or created manually in Oracle Enterprise Repository. The `12.1.3.0.0-OER-Harvester-Solution-Pack.zip` contains all of the Adapter and Application assets that are configured in the default `adapters.xml` file.

Each `adapter` element indicates an Adapter Configuration and/or Application Deployment asset that is created. The name attribute must match the name of a corresponding `adapterAsset` element. Each `adapter` element must have either the `jcaSpecBinding` or `namespace` attribute present.

- **jcaSpecBinding**

The `jcaSpecBinding` attribute corresponds to the fully-qualified classname of a JCA `interactionSpec` or `activationSpec` class, as appears in WSDLs created by Oracle SOA Suite. If `jcaSpecBinding` is present, then an Adapter Configuration asset is created.

---

**Note:** An Application Deployment asset may also be created if the corresponding `adapterAsset` element has the `applicationName` attribute present.

---

- **jcaCSAAdapterName**

The `jcaCSAAdapterName` attribute is optional. It may be specified if the JCA `interactionSpec/activationSpec` is not enough to distinguish the adapter type. If specified, then it must match the `cs.AdapterName` attribute in the `jca:address` section of the WSDL.

- **namespace**

The `namespace` attribute corresponds to the `targetNamespace` of a WSDL created by Oracle SOA Suite, using the Oracle BSE framework. If `namespace` is present,

then an Adapter Configuration asset will NOT be created, only an Application Deployment gets created.

If an adapter or application has several possible JCA classes or XML namespaces, then multiple adapter elements can correspond to one `adapterAsset`. Each `adapterAsset` element indicates an adapter and/or application asset that is created. The name attribute must match the name of a corresponding adapter element. Only one adapter and/or application asset is created in Oracle Enterprise Repository for each `adapterAsset` element.

- **applicationName**

The `applicationName` attribute, if present, indicates that an application asset should also be created. Each property element will result in the specified data being set in the Harvester Properties section for the adapter asset created.

For more information about the harvester properties, see [Section A.7, "Harvester Properties"](#).

## 8.3 Harvesting of Artifacts

This section describes how you can configure Harvester and use it within an integrated development environment.

This section contains the following topics:

- [Section 8.3.1, "Harvesting Artifacts"](#)
- [Section 8.3.2, "Detecting Duplicate Artifacts"](#)
- [Section 8.3.3, "Downloading Harvested Artifacts"](#)
- [Section 8.3.4, "Searching Harvested Assets"](#)
- [Section 8.3.5, "Previewing of the Created Assets in Harvester"](#)
- [Section 8.3.6, "Best Practices"](#)
- [Section 8.3.7, "Known Issues"](#)

### 8.3.1 Harvesting Artifacts

The metadata can be submitted to Oracle Enterprise Repository either from the command line or using an Ant task.

The Harvester scans for artifacts and harvests those artifacts to detect the dependencies that exist between them. The following sections describe the artifacts:

- [Section 8.3.1.1, "BPEL"](#)
- [Section 8.3.1.2, "WSDL"](#)
- [Section 8.3.1.3, "WADL"](#)
- [Section 8.3.1.4, "XSD"](#)
- [Section 8.3.1.5, "XSL"](#)
- [Section 8.3.1.6, "Harvesting Oracle SOA Suite"](#)
- [Section 8.3.1.7, "JCA Adapters"](#)
- [Section 8.3.1.8, "Applications via Oracle BSE \(WSDL/SOAP\)"](#)
- [Section 8.3.1.9, "Proxy Service or Business Service"](#)

- [Section 8.3.1.10, "Oracle Service Bus \(OSB\) Project"](#)
- [Section 8.3.1.11, "BPMN"](#)

The Harvester creates entities for these artifacts in Oracle Enterprise Repository and creates the relationships between them.

The artifacts created in Oracle Enterprise Repository are stored in the created asset's File Information. The locally stored harvested artifacts store the physical file in Oracle Enterprise Repository whereas for the remotely stored harvested artifacts (<http://>), Oracle Enterprise Repository points to the URL location of that artifact. Locally stored harvested artifacts, by default, will not be published to OSR by the Exchange Utility. Only remotely stored artifacts will be published.

[Figure 8-3](#) shows the asset types created by the Harvester and the relationships between them and [Figure 8-4](#) describes a sample Asset Model diagram.

---

---

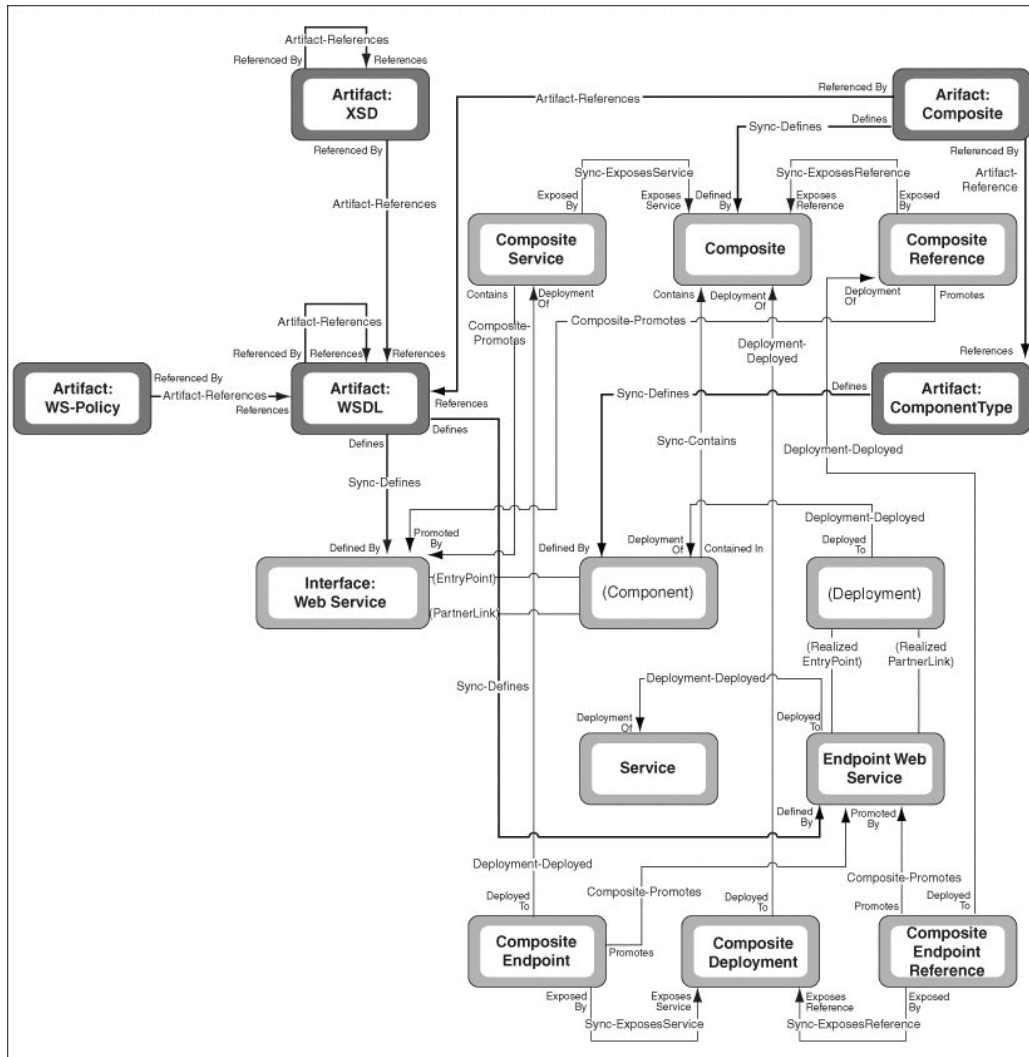
**Note:** Relationship role names are listed in the order of Primary role, then Secondary role.

In the diagrams which follow, the Primary role is at the tail of each arrow, and the Secondary role is at the head of each arrow.

---

---

Figure 8-3 Asset Types Created by the Harvester

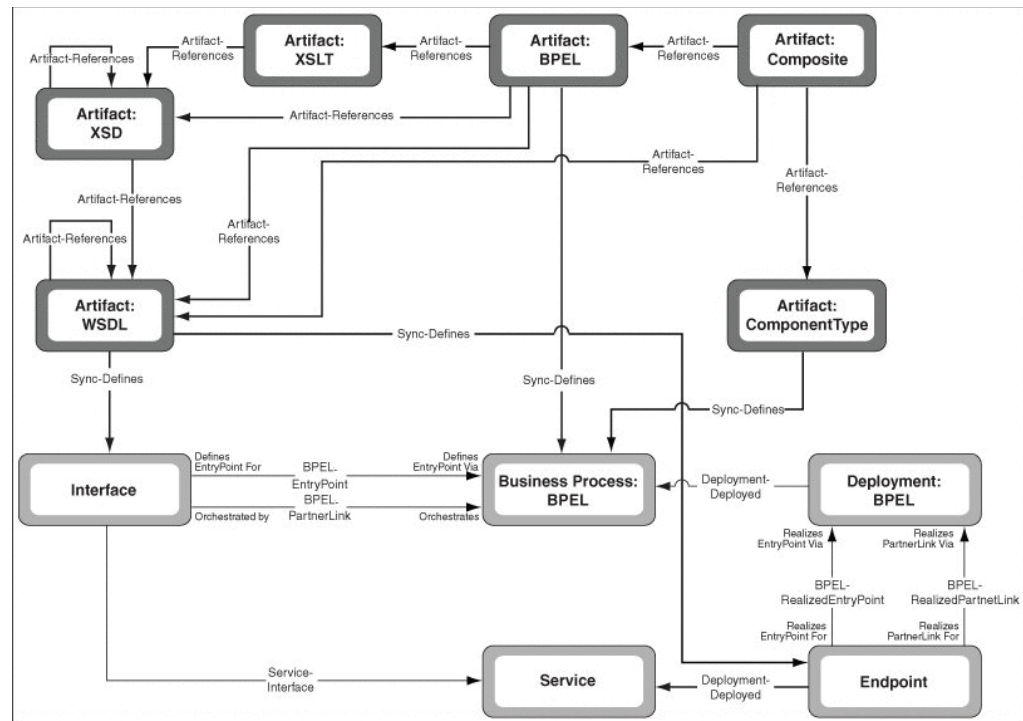


### 8.3.1.1 BPEL

When a BPEL artifact is submitted to Oracle Enterprise Repository, it will result in the following in Oracle Enterprise Repository:

- A Business Process asset (of type: Business Process:BPEL) is directly related to the artifact asset of type Artifact:BPEL.
- A BPEL artifact asset (of type Artifact:BPEL) is created that contains the BPEL artifact contents.
- The Business Process is related to a BPEL artifact asset using the "Defined by" relationship.
- For every partner link in the BPEL flow, the Business Process will also be related to interface assets (of type Interface).
- The BPEL artifact asset is related to WSDL and XSLT artifact (if a transformation is performed in the flow) assets. These are the WSDL that contain the definitions of the partnerlinks and the entry points for the Business Process.

Figure 8–4 Asset Model



### 8.3.1.2 WSDL

When a WSDL artifact is submitted to Oracle Enterprise Repository, it will result in the following in Oracle Enterprise Repository:

- If the WSDL contains a Service, a Service asset (of type: "Service") is created.
- An interface asset (of type: "Interfaces") is created for the port type.
- The Service is related to the Interface asset using the "Contains interface" relationship.
- An Endpoint asset is created for the port.
- The Service asset is related to the Endpoint asset using the "Deployed to" relationship.
- WSDL artifact asset (of type "Artifact:WSDL") is created that contains the WSDL artifact contents.
- If the WSDL artifact imports WSDLs and imports / includes XSDs, it is related to those WSDL and XSD artifact assets using the "References" relationship.

### 8.3.1.3 WADL

When a WADL artifact is submitted to Oracle Enterprise Repository, it will result in the following in Oracle Enterprise Repository:

- If the WADL contains a Service, a Service asset (of type: "Service") is created.
- An interface asset (of type: "Interfaces") is created for the port type.
- The Service is related to the Interface asset using the "Contains interface" relationship.
- An Endpoint asset is created for the port.

- The Service asset is related to the Endpoint asset using the "Deployed to" relationship.
- WADL artifact asset (of type "Artifact:WADL") is created that contains the WADL artifact contents.
- If the WADL artifact imports WSDLs and imports / includes XSDs, it is related to those WSDL and XSD artifact assets using the "References" relationship.

#### 8.3.1.4 XSD

When a XSD artifact is submitted to Oracle Enterprise Repository, it will result in the following in Oracle Enterprise Repository:

- XSD artifact asset (of type "Artifact:XSD") is created that contains the XSD artifact contents.
- If the XSD artifact imports / includes XSDs, it is related to those XSD artifact assets using the "References" relationship.

#### 8.3.1.5 XSL

When a XSL artifact is submitted to Oracle Enterprise Repository, it will result in the following in Oracle Enterprise Repository:

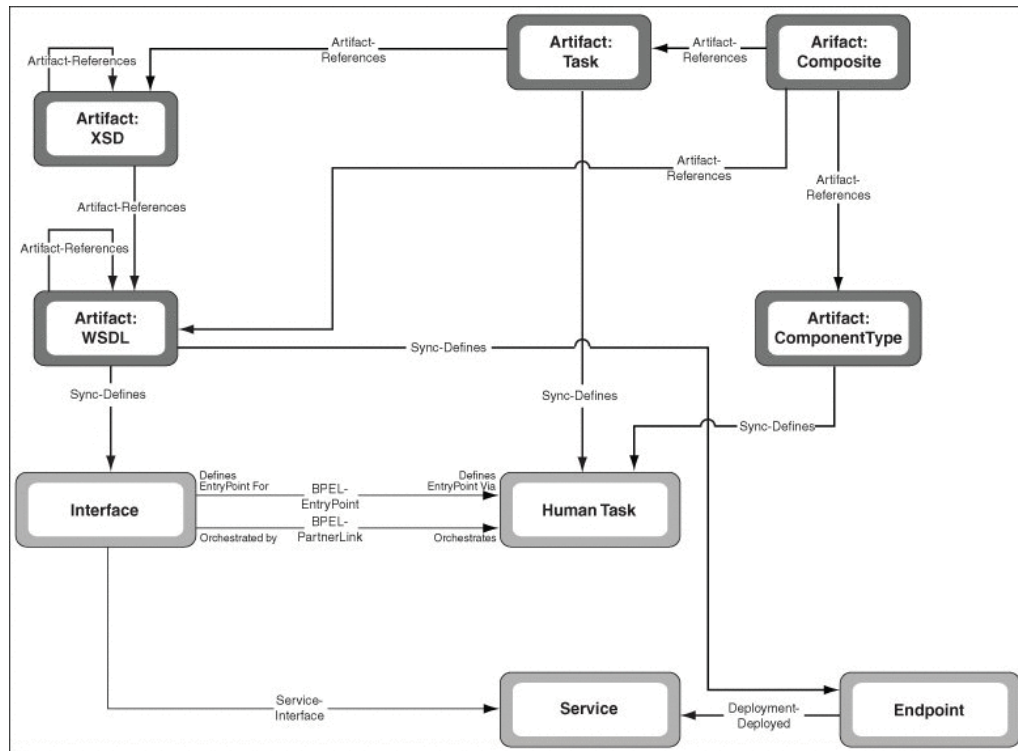
- XSL artifact asset (of type "Artifact:XSL") is created that contains the XSL artifact contents.
- If the XSL artifact references XSDs and WSDLs as source and target for the transformation, it is related to those XSD and WSDL artifact assets using the "References" relationship.

#### 8.3.1.6 Harvesting Oracle SOA Suite

The section describes harvesting an Oracle SOA Suite project.

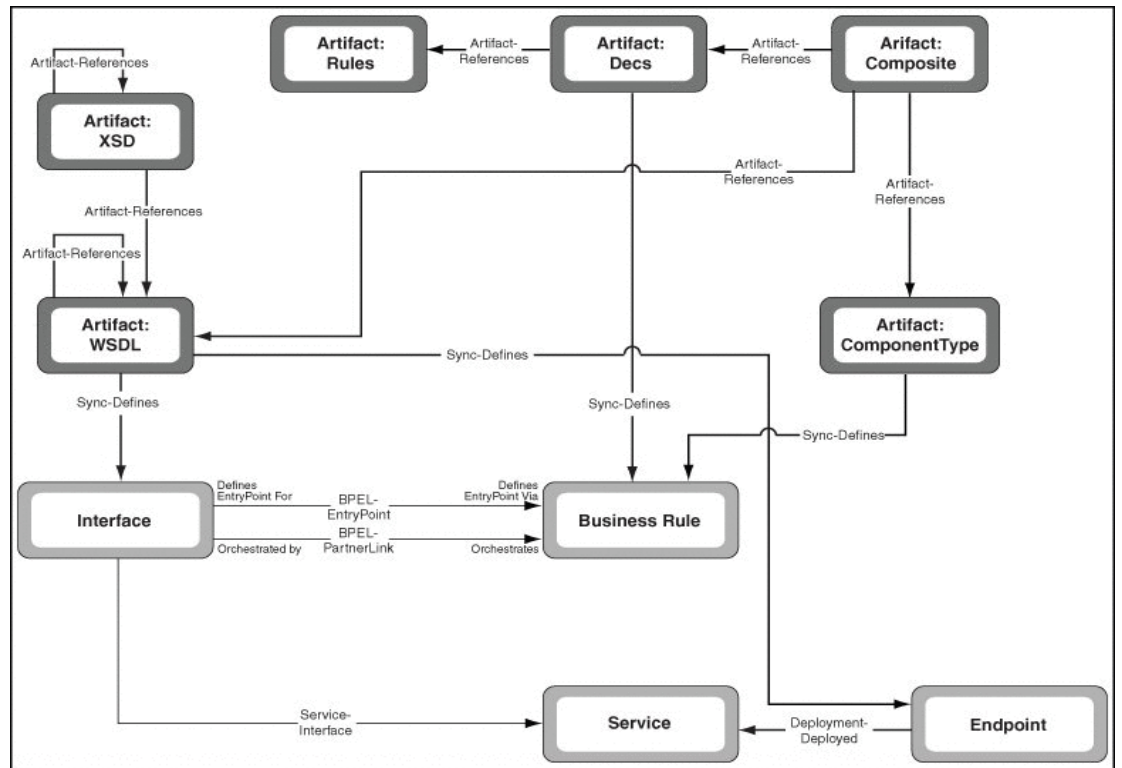
- Composite: When an Oracle SOA Suite project folder or its `composite.xml` file is introspected, it results in the following actions in Oracle Enterprise Repository:
  - Composite asset is created.
  - Any services exposed by the composite results in Composite Service assets in Oracle Enterprise Repository. These are related to the Composite by the Sync-ExposesService relationship.
  - Any references exposed by the composite results in Composite Reference assets in Oracle Enterprise Repository. These are related to the Composite by the Sync-ExposesReference relationship.
  - The `composite.xml` file becomes an asset of the Artifact: Composite type and relates to the Composite by the Sync-Defines relationship.
- Composite Deployment: When an Oracle SOA Suite project folder or its `composite.xml` file is introspected, it results in the following actions in Oracle Enterprise Repository:
  - Composite Deployment asset is created.
  - Any services exposed by the composite deployment results in Composite Deployment Service assets in Oracle Enterprise Repository. These are related to the Composite Deployment by the Sync-ExposesService relationship.
  - Any references exposed by the composite will result in Composite Deployment Reference assets in Oracle Enterprise Repository. These are



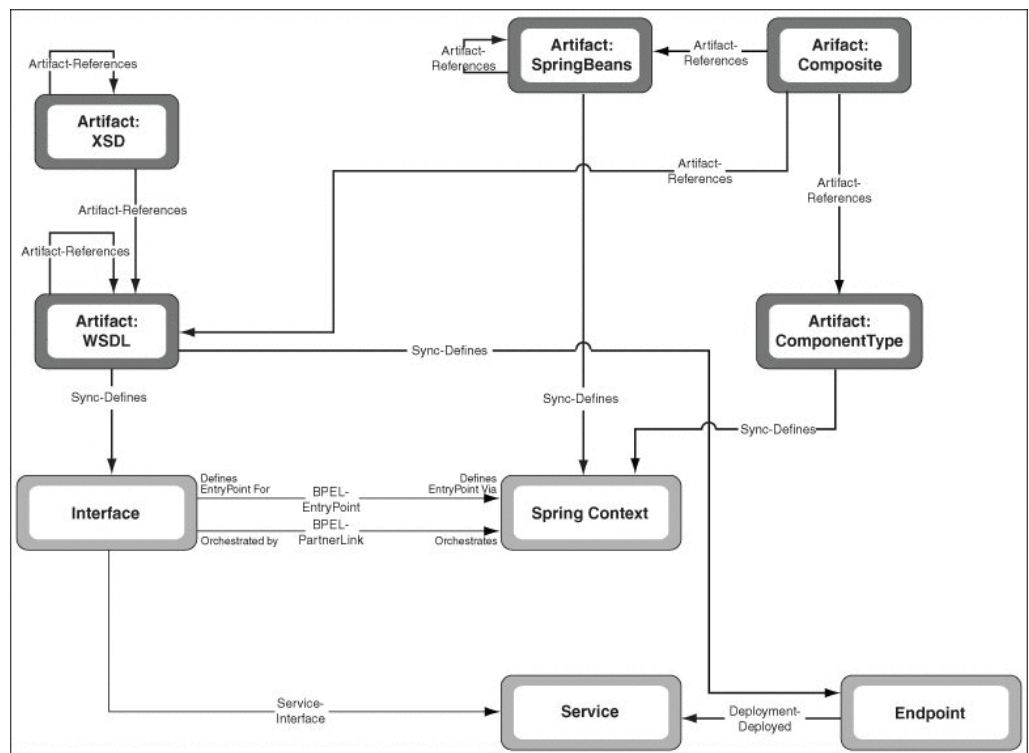


- Any Business Rule should result in a Business Rule asset.
  - \* It relates to an `Artifact:Decs` artifact by the `Sync-Defines` relationship. This relates to any dependent `Artifact:Rules` artifacts by the `Artifact-References` relationship. The `Artifact:Rules` artifacts are related to any dependent `Artifact:XSD` artifacts by the `Artifact-References` relationship.





- Any Spring Context component should result in a Spring Context asset.
- \* It is related to an `Artifact:SpringBeans` artifact by the `Sync-Defines` relationship. This is related to other `Artifact:SpringBeans` artifacts by the "Artifact-References" relationship.



- Component-level Relationships:
  - Any component services offered by a component results in an Interface asset. This is related to the component-level asset by the BPEL-Entry point relationship (except for Mediators, in which case it is related by the Service-Interface relationship).
    - \* In the case of `<interface.ws>` in the `composite.xml`, the Service is related to an Interface asset. In the case of `<interface.java>` in the `composite.xml`, the Service is related to an Interface with attribute `Interface Type = Java`.
    - \* If the component service is promoted to the composite level, the Service asset is related to the corresponding Composite Service asset by the Composite-Promotes relationship.
    - \* If the `composite.xml` defining that the component service uses the Web Service or WSIF binding, an Endpoint: Web Service asset is created. This will have the Binding Type attribute set to `Web Service` or `WSIF`. If the service is promoted, this is related to the Composite Deployment Service asset by the Composite-Promotes relationship.
    - \* If the `composite.xml` defining that the component service uses the EJB, or ADF binding, an Endpoint asset is created. This will have the Binding Type attribute set to `ADF Service` or `EJB Service`. If the service is promoted, this is related to the Composite Deployment Service asset by the Composite-Promotes relationship.
    - \* If the component service uses a JCA or B2B adapter, an Adapter Configuration asset is created. For JCA adapters, this is related to the corresponding `Artifact: JCA` asset. This is related to the corresponding Adapter asset. If the service is promoted, the Adapter Configuration is related to the Composite Deployment Service asset by the Composite-Promotes relationship.
  - Any non-promoted component references used by a component will result in an Interface asset. Any component reference will not have a new Service created for it. However, if the corresponding Service assets already exist in Oracle Enterprise Repository or in the current introspection, they are related to the Component Reference. This rule prevents duplicate Service assets from being created in the namespace of both the referring project and the project that exposes it. This is related to the component-level asset by the Partner-Link point relationship (except for Mediators, in which case it is related by the Service-Fronts relationship).
    - \* For a promoted component reference, if a corresponding Service asset is found in OER, the Service asset is related to the corresponding Composite Reference asset by the Composite-Promotes relationship.
    - \* For a deployed promoted component reference using the Web Service or WSIF binding, if a corresponding Endpoint: Web Service is found in OER, this is related to the Composite Deployment Reference asset by the Composite-Promotes relationship.
    - \* For a deployed promoted component reference using the ADF or EJB Binding, if a corresponding Endpoint is found in OER, this is related to the Composite Deployment Reference asset by the Composite-Promotes relationship.
    - \* For a deployed promoted component reference using the JCA or B2B adapter, if a corresponding Adapter Configuration asset is found in OER,

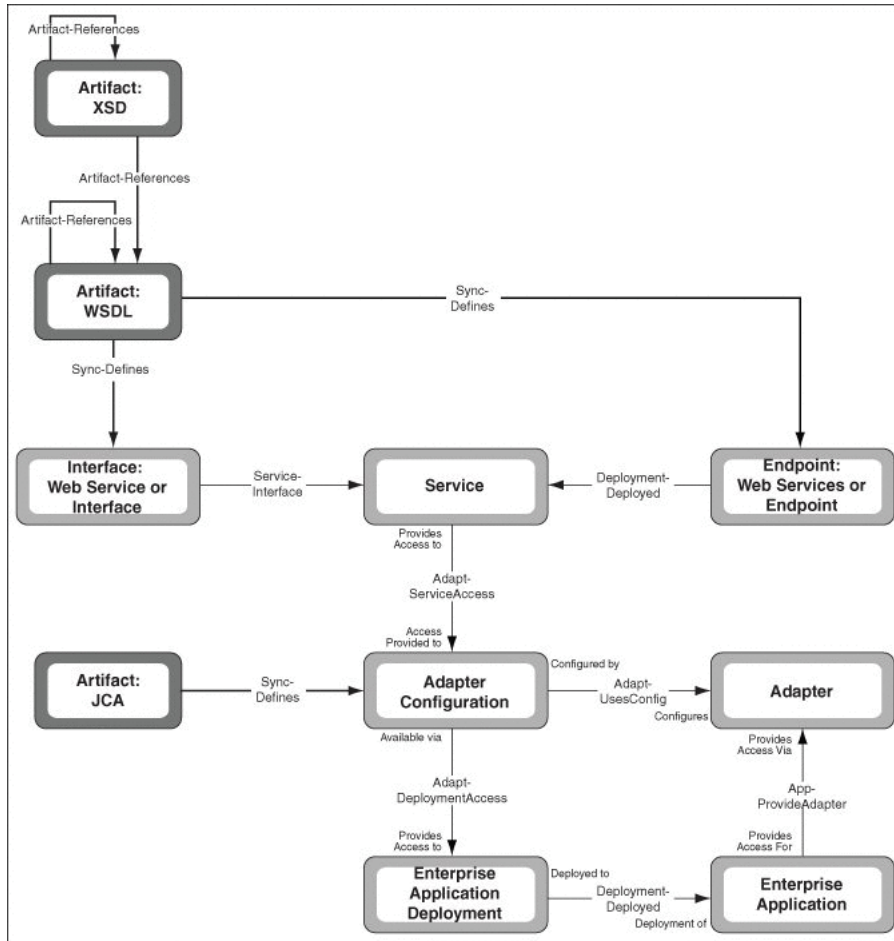
---

this is related to the Composite Deployment Service asset by the Composite-Promotes relationship.

### 8.3.1.7 JCA Adapters

When a WSDL is harvested from Oracle SOA Suite or Oracle Service Bus, which uses JCA Adapter extensions, it will result in the following actions in Oracle Enterprise Repository:

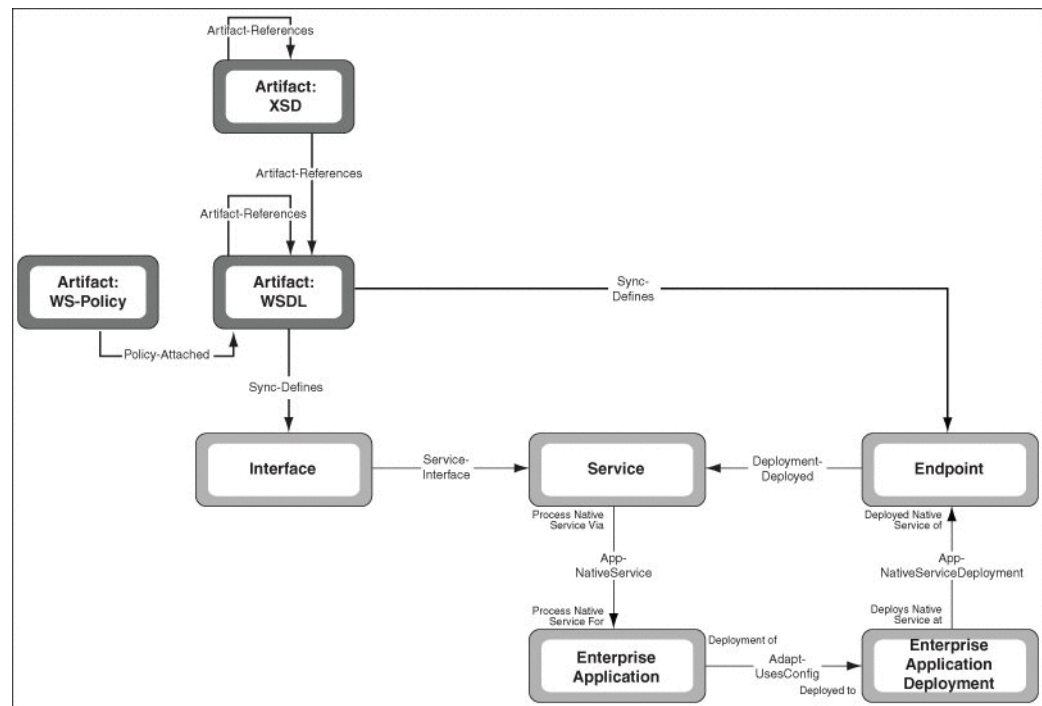
- Harvester attempts to locate an Adapter asset in Oracle Enterprise Repository, as configured in the Harvester's `adapters.xml` file.
- An Adapter Configuration asset is created that contains the binding and connectivity information used by a specific service.
  - It is related to the Adapter asset.
  - It is related to the Service asset that is using the adapter configuration.
- Harvester attempts to locate an Application asset in Oracle Enterprise Repository, as configured in the Harvester's `adapters.xml` file.
  - Application assets are supported, by default, for all the following adapters:
    - \* Oracle Applications (a.k.a. E-Business Suite)
    - \* PeopleSoft
    - \* Siebel
    - \* SAP
    - \* JDE One World
- If the Service has an Endpoint, then an Application Deployment asset is created.
  - It is related to the Application asset.
  - It is related to the Endpoint asset.



### 8.3.1.8 Applications via Oracle BSE (WSDL/SOAP)

When a BPEL process or application is harvested that uses BSE (Business Service Engine) to connect to an external application, it results in the following actions in Oracle Enterprise Repository:

- Harvester attempts to locate an Application asset in Oracle Enterprise Repository, as configured in the Harvester's `adapters.xml` file.
- If the Service has an Endpoint, then an Application Deployment asset is created.
  - It is related to the Application asset.
  - It is related to the Endpoint asset.



### 8.3.1.9 Proxy Service or Business Service

When a Proxy Service or Business Service is submitted to Oracle Enterprise Repository, it will result in the following in Oracle Enterprise Repository:

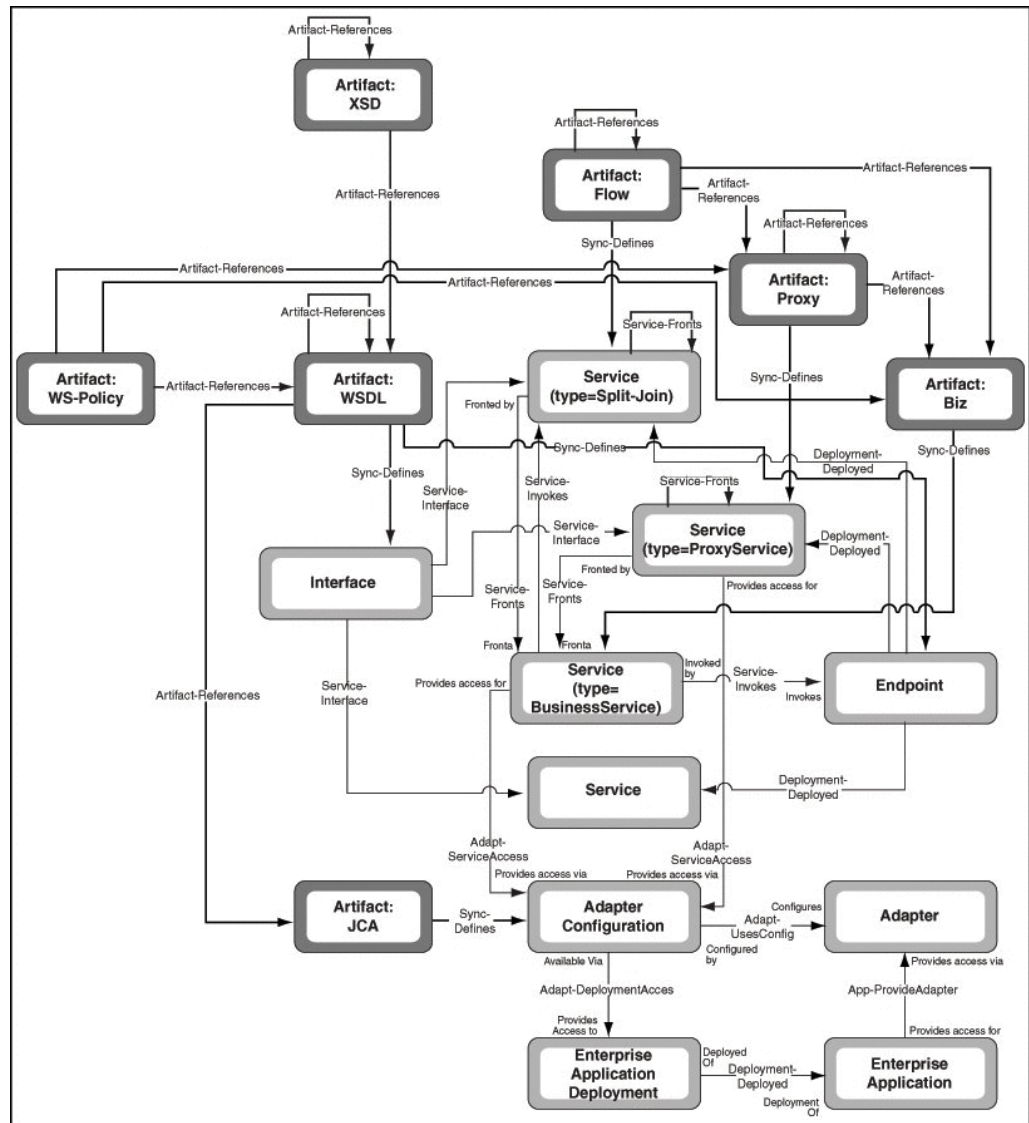
- A Service asset is created, with Service Type= "Proxy Service" or "Business Service".
- The service is related to the Interface asset using the relationship 'Contains interface'.
- If the Proxy or Business Service is of type *Web Service* in Oracle Service Bus.
  - An "Interface" asset is created for the port type.
  - An "Endpoint" asset is created for the port.
    - \* This will have an EndpointURI property visible in the Overview section in the Oracle Enterprise Repository Web UI.
  - The Service asset is related to the "Endpoint" asset using the relationship "Deployed to".
  - A WSDL artifact asset (of type "Artifact: WSDL") is created that contains the WSDL artifact contents.
- If the Proxy or Business Service is not of type *Web Service* in Oracle Service Bus
  - An Interface asset is created for the port type.
  - An Endpoint asset is created for the port.
    - \* This will have an EndpointURI property visible in the Harvester Properties section in the Oracle Enterprise Repository Web UI. For more information about harvester properties, see [Section A.7, "Harvester Properties"](#).
  - The Service asset is related to the Endpoint asset using the relationship "Deployed to".

- There is no WSDL associated with this service. So it will not be available to integrate with UDDI using the RR-XU utility. And the WSDL can not be consumed via the Oracle Enterprise Repository Web UI's "Use|Download" feature, or the "Service Consumption Framework" feature in Oracle Eclipse Workshop.
- If the WSDL artifact imports WSDLs and imports / includes XSDs, it is related to those WSDL and XSD artifact assets using the relationship 'References'.
- A Proxy Service is related to Business Services or Proxy Services that it fronts, via the "Fronts" relationship.
- A Business Service is related to external services and flows that it invokes, via the "Invokes" relationship.
- Artifacts that define the interface in Oracle Service Bus (`Artifact:WSDL`, `Artifact:XSD`, or `Artifact:MFL`) is related to the Interface asset via the "Defined by" relationship.
- Artifacts that are used to implement the service in Oracle Service Bus (`Artifact:XQuery`, `Artifact:XSLT`, `Artifact:MFL`, or `Artifact:WS-Policy`) is related to the `Artifact:Proxy` or `Artifact:Biz` asset via the "Artifact-References" relationship.

#### **8.3.1.10 Oracle Service Bus (OSB) Project**

An Oracle Service Bus project can be submitted to Oracle Enterprise Repository from the command line. See [Section 8.2.5.3, "Harvesting Web Services from Oracle Service Bus"](#) for information about harvesting from OSB.

The harvester will handle Proxy Services, Business Services, and Flows. It will also handle artifacts such as BPEL, WSDL, XSD, XSLT, XQuery, WS-Policy and MFL, and harvest the artifacts to detect the dependencies that exist between these artifacts. The harvester will create entities in Oracle Enterprise Repository and create relationships between them.



**Note:** When harvesting, use the harvesters that are more appropriate to the content being harvested. For example, when harvesting Oracle Service Bus projects, use the Oracle Service Bus harvester.

### 8.3.1.11 BPMN

When you submit a BPMN artifact to Oracle Enterprise Repository, it will result in the following in Oracle Enterprise Repository:

- A Business Process asset (of type Business Process:BPMN) is directly related to the artifact asset of type Artifact: BPMN.
- A BPMN artifact asset (of type Artifact:BPMN) is created that contains the BPMN artifact contents.
- The Business Process is related to a BPMN artifact asset using the "Defined by" relationship.

## 8.3.2 Detecting Duplicate Artifacts

The Harvester stores information about files such as WSDLs, BPELs, BPMNs, and XSDs as artifacts in Oracle Enterprise Repository. To avoid storing the same artifact file twice, the Harvester will calculate a Software File ID (SFID) for each artifact when it is stored. Before submitting a new artifact, the SFID can be compared against existing SFIDs in the repository to check for duplicates.

The SFID calculated is an MD5 hash. Some level of canonicalization is performed before calculating the SFID. In particular, if the artifact file is XML, it is canonicalized using the Canonicalizer class in the Apache XML Security library. This canonicalizes according to the W3C Canonical XML standard (see [www.w3.org/TR/xml-c14n](http://www.w3.org/TR/xml-c14n)), which includes canonicalizing the text encoding, line breaks, whitespace, comments, and attribute ordering. Some extra canonicalization not specified in the W3C standard is performed, including normalizing of namespace prefixes, normalizing the order of the elements in WSDLs, removing documentation elements, and inlining any included/imported files.

## 8.3.3 Downloading Harvested Artifacts

The Harvester creates artifact bundles that may be downloaded from the Harvested Assets. The artifact bundles for these assets are stored in zip files. For example, for an Endpoint, a WSDL file and its associated XSD files are stored in relative locations within the zip payload.

When one artifact imports another artifact (for example, a WSDL imports a XSD), it always refers to the child artifact relative to the parent. For example, if `MyWSDL.wsdl` is located in `c:\temp` and if the child XSD that is being imported resides in `c:\temp\schemas\MyXSD.xsd`, the parent `MyWSDL.wsdl` imports the child using the relative path `./schemas/MyXSD.xsd`. When the bundle is downloaded, the child artifact should be created in a folder called `schemas` relative to the parent so that the parent can resolve the child.

For more information about Using/Downloading Assets, see "Using/Downloading Assets" in the *Oracle Fusion Middleware Developer's Guide for Oracle Enterprise Repository*.

## 8.3.4 Searching Harvested Assets

You can search for harvested assets using either of the following two methods:

- [Section 8.3.4.1, "Using Consumption Method"](#)
- [Section 8.3.4.2, "Using Metadata"](#)

### 8.3.4.1 Using Consumption Method

You can search for harvested services based on a consumption method. In the Basic Search section, when you can select WSDL Ports from the Consumption Method list, as shown in [Figure 8-5](#), the search returns the services that have a WSDL summary, an Endpoint defined in the WSDL, and a scope set to **global**. This list also can be used under the More Search Options.



**Figure 8–5 Assets Pane - Consumption Method**

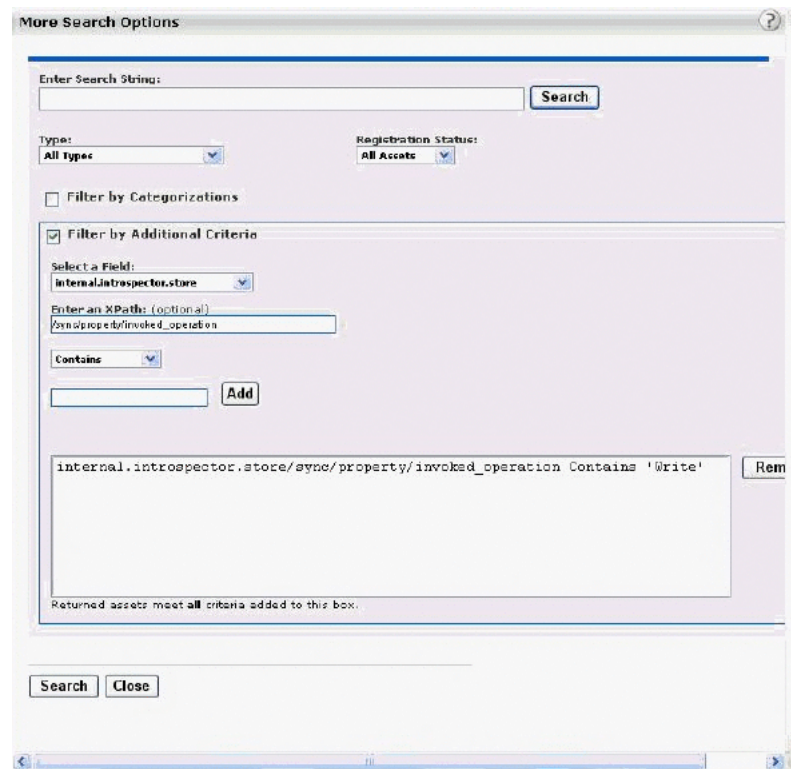
The screenshot shows a web interface titled "Assets". At the top, there are links for "Submit an Asset" and "Edit / Manage Assets". Below this is a "Search" section with a text input field labeled "Enter Search String" and a help icon. Underneath is a "Type" dropdown menu set to "All Types" with a "..." button. The "Consumption Method" section features a dropdown menu with "Select a consumption method" selected, a list of options including "WSDL Ports", and a "Registered" dropdown menu. Below that is an "Asset Function" dropdown menu set to "All Asset Functions" with a "..." button. At the bottom, there is a "More Search Options" link and a "Search" button.

### 8.3.4.2 Using Metadata

The Harvester tags each asset with properties that can be used for querying.

Figure 8–6 shows how to query for Business Processes that invoke the Write operation. To get the search screen, click **More Search Options** in the main page of the Oracle Enterprise Repository Web console. In the More Search Options dialog, perform the following:

- **Filter by Additional Criteria:** Select the option.
- **Select a Field:** Choose `internal.introspector.store` from the list.
- **Enter an XPath:** Type `"/"` in the text field. A list of available fields is displayed.

**Figure 8–6 Viewing the More Search Options Dialog in Oracle Enterprise Repository**

The following search criteria are available:

- Harvester Description
- Harvester Version
- Harvester Namespace
- Harvested by
- Invoked Operations of Business Processes

### 8.3.5 Previewing of the Created Assets in Harvester

You can use the **Preview** feature to view the created assets in harvester. You can use the Preview feature using either of the following methods:

#### Using Command Line

At the command line or in an ant task, if you add `-preview true`, then the Harvester runs and displays a list of all of the assets that would be created, but it does not actually commit the changes.

#### Using the HarvesterSettings.xml File

You can also set the preview mode in the `HarvesterSettings.xml` file as follows:

```
<introspection preview="true">
  <reader>com.oracle.oer.sync.plugin.reader.file.FileReader</reader>
  <writer>com.oracle.oer.sync.plugin.writer.oer.OERWriter</writer>
</introspection>
```

## 8.3.6 Best Practices

This section describes best practices for the Harvester. It contains the following topics:

- [Section 8.3.6.1, "Recommended Privileges for Harvesting"](#)
- [Section 8.3.6.2, "Do Not Override the Namespace Parameter"](#)
- [Section 8.3.6.3, "Namespaces in WSDL Files"](#)
- [Section 8.3.6.4, "Harvesting Completed Work"](#)
- [Section 8.3.6.5, "Harvesting and Maintenance Releases of XSD"](#)
- [Section 8.3.6.6, "Harvesting Metadata throughout the Asset Lifecycle"](#)
- [Section 8.3.6.7, "Downloading WSDL Artifacts"](#)
- [Section 8.3.6.8, "Harvesting Retired Endpoints"](#)
- [Section 8.3.6.9, "Harvesting SOA Projects"](#)
- [Section 8.3.6.10, "Harvesting a File in the Harvester Folder"](#)
- [Section 8.3.6.11, "Running Harvester and Exchange Utility on the Same Computer"](#)
- [Section 8.3.6.12, "Harvesting Composite Service WSDL"](#)
- [Section 8.3.6.13, "Understanding Versioning of Assets"](#)

### 8.3.6.1 Recommended Privileges for Harvesting

Only Registrars or individuals with the authority to view all the assets in Oracle Enterprise Repository should harvest assets. If individuals do not have permission to view all assets in the repository, they may harvest assets that already exist and unintentionally duplicate assets.

### 8.3.6.2 Do Not Override the Namespace Parameter

Harvester offers you with the ability to set the project namespace for the assets being harvested. The project namespace is used in detecting duplicate assets, as different namespaces will result in different assets. In most cases, you are recommended not to override this parameter.

However, harvester automatically sets the project namespace based on the Oracle SOA Suite or Oracle Service Bus project name when harvesting from those products, which is the recommended behavior.

This is especially important to follow, when you harvest the same project and revision from multiple sources. Changing or overriding the namespace leads to duplicate assets in OER.

### 8.3.6.3 Namespaces in WSDL Files

In the WSDL files that you harvest, it is recommended that you use a unique namespace for each unique interface, service, and endpoint.

Correlation to existing assets in the Oracle Enterprise Repository is done through QNames, so if you make significant changes to interface, service, or endpoint assets and do not change the QNames, you will overwrite the existing asset with the modified asset.

[Table 8–7](#) shows the correlation of Oracle Enterprise Repository assets with WSDL structure:

**Table 8–7 Correlation of Oracle Enterprise Repository Assets with WSDL Structures**

Repository Asset	WSDL Structure
Service	/definition/service/@name
Endpoint	/definition/service/port/@name
Interface	/definition/portType/@name

### 8.3.6.4 Harvesting Completed Work

It is recommended that you harvest only work that is completed or near completion. If you regularly harvest from a development environment, the Oracle Enterprise Repository can become cluttered with outdated versions of assets.

Oracle Enterprise Repository promotes only services that have endpoints and hosted WSDL to Oracle Service Registry. In addition, only services with concrete WSDL can be consumed through JDeveloper. It is recommended that you harvest from the runtime environment to obtain the endpoints and WSDL location.

### 8.3.6.5 Harvesting and Maintenance Releases of XSD

Some schema development patterns involve the maintenance release of schemas that fix defects or add minor structures but do not change the namespace of the schema. It should be recognized that subsequent harvesting of slightly modified schema artifacts can have the effect of creating a significant number of new artifact assets in the repository. Oracle Enterprise Repository correlates artifact assets based on a hash, or Software File ID (SFID), of the contents of the artifacts. The SFID is calculated over the contents of each artifact after all imports and includes have been inlined.

Consequently, a change in an XSD that is imported by a WSDL will result in both a new XSD artifact and a new WSDL artifact.

This is particularly important to keep in mind when considering schemas that are widely used throughout the enterprise. For example, consider a low-level schema such as `customer.xsd` that is widely imported by other schemas, WSDLs, XSLTs, BPELs. A material change to `customer.xsd`, and a subsequent re-harvesting of all of an enterprise's artifacts (for example, some kind of regular batch harvesting) would result in a large number of similar artifact assets in the repository that reference `customer.xsd` either directly or indirectly.

### 8.3.6.6 Harvesting Metadata throughout the Asset Lifecycle

Harvester can be invoked from several places in the asset lifecycle and each invocation gathers artifacts and metadata from different sources. The following list describes the recommendation on how to use harvester during an asset's lifecycle to keep Oracle Enterprise Repository up to date:

1. Deployment time

Immediately after deployment, assets should be published to Oracle Enterprise Repository. At deployment, the production endpoints are harvested and the asset's metadata is updated with the endpoint information. An organization may have multiple endpoints supporting their development, test, staging, and production environments. Endpoints, when harvested into Oracle Enterprise Repository, may also go through a governance process. For example, development and testing endpoints may be made available to the entire consumer community, but production endpoints may not be provided until a service provider and a service consumer negotiate a service contract.

**Recommended approach:** Invoke the harvester from the deployment script (WLST, ant, or command-line).

Invoke harvester at the end of deployment. Harvest the deployed artifacts from the running server, using the `-remote_server` argument.

## 2. Runtime

Once the project is deployed and running, runtime metrics are harvested and associated with Oracle Enterprise Repository assets. There are several options available for harvesting runtime metrics. One alternative is the EM-Integration, which gathers runtime metrics from Enterprise Manager and updates Oracle Enterprise Repository. For more information, see *Oracle Enterprise Manager Connectors Integration Guide*.

Another alternative is to rely upon third-party runtime monitoring tools that publish runtime metrics to UDDI registries. Once in the UDDI Registry, the metrics can be propagated back to Oracle Enterprise Repository using the Oracle Registry Repository Exchange Utility.

### 8.3.6.7 Downloading WSDL Artifacts

When consuming a WSDL artifact in Oracle Enterprise Repository, you have the ability to download the WSDL file. In many cases, the location of this file is on a remote location. Also, the URL for that file may end in `?wsdl`. In these cases, when saving that WSDL, the browser may, by default, give the file the extension as `.xml`. Then, you should replace `.xml` with `.wsdl`. A WSDL file can be hosted on an HTTP/HTTPS or FTP server.

### 8.3.6.8 Harvesting Retired Endpoints

When endpoint is changed, either retired or moved to another location, then it is recommended to retire endpoint from Oracle Enterprise Repository and remove endpoint for Oracle Service Registry. Then, reharvest the same WSDL service with a new endpoint.

---

**Note:** If you re-run the Exchange Utility publisher, it will not delete retired endpoint from Oracle Service Registry.

---

### 8.3.6.9 Harvesting SOA Projects

As a best practise for the SOA life cycle, it is recommended to harvest any SOA projects to Oracle Enterprise Repository, deploy to the SOA server, run the Exchange Utility to publish to Oracle Service Registry, and then consume published assets from Oracle Enterprise Repository or Oracle Service Registry.

### 8.3.6.10 Harvesting a File in the Harvester Folder

It is recommended not to harvest a file that is located within the unzipped Harvester folder, `12.1.3.0.0-OER-Harvester.zip`. Currently, this is not supported.

### 8.3.6.11 Running Harvester and Exchange Utility on the Same Computer

When publishing a service to Oracle Service Registry, it is recommended that you run Exchange Utility and Harvester on the same computer so they resolve to the same URL. You can also run these on separate computers where the access points in a WSDL are not fully formed, but you must ensure that they resolve to the same URL on both

computers. If the access points are resolved to different URL for Harvester and Exchange Utility, then the service fails to publish with error.

### 8.3.6.12 Harvesting Composite Service WSDL

It is recommended that a WSDL of exposed composite service should never be harvested separately by itself. It is always best to harvest as whole Composite Project to get the complete and correct SOA asset model in Oracle Enterprise Repository.

### 8.3.6.13 Understanding Versioning of Assets

When defining a harvesting strategy, it is important to understand that different harvesters yield different results, for example:

- Using OER harvester on a running SOA infra or SOA composite (MDS), gets complete composite and relationships
- Using OSB harvester on OSB gets relationships between proxy and back end business services

When harvesting a service multiple times, OER will attempt to determine is the service is the same as the last time it was harvested. In Oracle Enterprise Repository 12c, two entities are determined to be the same in the following ways:

- Composites are determined to be the same they have the same name/version pair (may add new relationships, deployments, etc for same composite).
- Services are determined to be the same if they share the same QName, as identified in WSDL that is introspected (version number specified in the WSDL has no effect).
- WSDLs, XSDs, composite.xml files, and other file types are considered the same artifact if they have the same SFID "fingerprint" (SFID is calculated on canonicalized artifact).

If OER finds a match in harvesting, it assumes that it already has the entity (and it may overwrite description, and add relationships to other entities to preserve the latest). If OER does not find a match, it assumes that the entity is a distinct, new entity (If artifacts have same name, but different content, it may append "-1" to make new asset name unique).

If your objective is to "refresh" the same asset, for example, for minor bug fixes or enhancements, it is recommended that you:

- Keep the endpoint address the same or add PortType for new interface on same service
- Keep the QName of the service the same in the WSDL
- Keep the Name/Version pair for composites the same (and do not issue a new composite revision in SOA Suite)

---

---

**Note:** Artifacts that are modified are simply added incrementally to the endpoints they relate to, and latest artifacts are served to consumers of endpoints. If another artifact has the same name, then OER appends -1 to the new artifact name. If user consumes the WSDL for a given service endpoint, they get the last submitted artifact for the corresponding endpoint.

---

---

If your objective is to "version" an asset by issuing a new asset with a new version number, it is recommended that you:

- Optionally add PortType for new interface on same service.
- Optionally create previous/next relationship.
- Optionally enable notification on previous/next relationship.
- Run side-by-side until consumers have migrated; then inactivate and retire the old asset.

The recommended best practice is to do a mixture of both refreshing and versioning, depending on the expectations you set with your asset consumers.

### 8.3.7 Known Issues

This section describes the following known Harvester issues:

- [Section 8.3.7.1, "Asset Types Must be Present in the System"](#)
- [Section 8.3.7.2, "Two Versions of an Asset Type"](#)
- [Section 8.3.7.3, "Do Not Delete the Harvester-Specific Metadata Entries in a in a Harvested Asset"](#)
- [Section 8.3.7.4, "Supported Remote Server Types"](#)
- [Section 8.3.7.5, "Using Incorrect Encrypted Password"](#)
- [Section 8.3.7.6, "View in Repository Option"](#)

#### 8.3.7.1 Asset Types Must be Present in the System

As a prerequisite to using Harvester features, the asset types must be present in the system. The necessary asset types are installed with 12.1.3.0.0-OER-Harvester-Solution-Pack.zip.

#### 8.3.7.2 Two Versions of an Asset Type

If some of the existing asset type names in your Oracle Enterprise Repository have the same names as the asset types installed with 12.1.3.0.0-OER-Harvester-Solution-Pack.zip, the asset type names for the Harvester will have a version number appended to them. This does not affect the functioning of the Harvester asset types and harvester refers to these asset types based on its UUID.

#### 8.3.7.3 Do Not Delete the Harvester-Specific Metadata Entries in a in a Harvested Asset

When the Harvester creates assets during the harvesting process, it attaches metadata entries to the asset of metadata entry Type: `internal.inspector.store` and `internal.harvester.manifest.store`. Do not modify or delete these metadata entries. Doing so can cause the system to behave unpredictably.

Note that it is not possible to delete these metadata entries using Oracle Enterprise Repository user interface.

#### 8.3.7.4 Supported Remote Server Types

Currently, the Harvester only supports connecting to the remote Oracle SOA Suite, Oracle Service Bus, and WebLogic Server projects that are running on WebLogic 11g or 12c.

### 8.3.7.5 Using Incorrect Encrypted Password

If you configure the `HarvesterSettings.xml` file with incorrect encrypted password, then a long stack trace is displayed, which is as follows:

```
log4j:WARN No appenders could be found for logger
(org.apache.axis.i18n.ProjectResourceBundle).
log4j:WARN Please initialize the log4j system properly.
@ com.oracle.oer.sync.framework.MetadataIntrospectionException:
@ com.oracle.oer.sync.framework.MetadataIntrospectionException: Unable to read
plugin file:
C:\Drive-E\XU-harvest-tools\dec16\OER-Harvester\.\plugins\biz.introspector
    at
@ com.oracle.oer.sync.framework.MetadataManager.init(MetadataManager.java:308)
@      at com.oracle.oer.sync.framework.Introspector.<init>
Introspector.java:188)
    at
@ com.oracle.oer.sync.framework.Introspector.main(Introspector.java:395)
@ Caused by: com.oracle.oer.sync.framework.MetadataIntrospectionException:
Unable
to read plugin file:
C:\Drive-E\XU-harvest-tools\dec16\OER-Harvester\.\plugins\biz.introspector
    at
@ com.oracle.oer.sync.framework.impl.DefaultPluginManager.processIntrospector(De
faultPluginManager.java:127)
    at
@ com.oracle.oer.sync.framework.impl.DefaultPluginManager.<init>(DefaultPluginMa
nager.java:73)
    at
@ com.oracle.oer.sync.framework.MetadataManager.init(MetadataManager.java:306)
    ... 2 more
Caused by: java.lang.IllegalArgumentException: The char '0x12' in
'java.lang.IllegalArgumentExcep
tion: The char '0x12' in 'E/?&#8597;?&#8597;H?rd' is not a
valid XML character.' is not a valid XML character.
    at
@ org.apache.axis.components.encoding.AbstractXMLEncoder.encode(Abstrac
tXMLEncoder.java:110)
    at org.apache.axis.utils.XMLUtils.xmlEncodeString(XMLUtils.java:131)
    at org.apache.axis.AxisFault.dumpToString(AxisFault.java:366)
    at org.apache.axis.AxisFault.printStackTrace(AxisFault.java:796)
    at
com.oer.log4j.spi.ThrowableInformation.getThrowableStrRep(Throw
ableInformation.java:42)
    at
com.oer.log4j.spi.LoggingEvent.getThrowableStrRep(LoggingEvent.java:217)
    at
com.oer.log4j.WriterAppender.subAppend(WriterAppender.java:298)
    at
com.oer.log4j.RollingFileAppender.subAppend(RollingFileAppender.java:294
)
    at com.oer.log4j.WriterAppender.append(WriterAppender.java:157)
    at
com.oer.log4j.AppenderSkeleton.doAppend(AppenderSkeleton.java:251)
    at
com.oer.log4j.helpers.AppenderAttachableImpl.appendLoopOnAppenders(Appen
derAttachableImpl.java:57)
    at com.oer.log4j.Category.callAppenders(Category.java:255)
    at com.oer.log4j.Category.forcedLog(Category.java:445)
    at com.oer.log4j.Category.log(Category.java:882)
    at
```



```
@ com.oracle.oer.sync.framework.logger.DefaultMetadataLogger.error(DefaultMetadataLogger.java:98)
    at
@ com.oracle.oer.sync.plugin.writer.oer.OERConnectionCache.getAuthToken(OERConnectionCache.java:138)
    at
@ com.oracle.oer.sync.plugin.writer.oer.OERAssetQueries.getToken(OERAssetQueries.java:82)
    at
@ com.oracle.oer.sync.plugin.writer.oer.OERAssetQueries.assetTypeQueryByUUID(ALERAssetQueries.java:159)
    at
@ com.oracle.oer.sync.framework.MetadataManager.putAssetType(MetadataManager.java:204)
    at
@ com.oracle.oer.sync.framework.impl.DefaultPluginManager.processIntrospector(DefaultPluginManager.java:104)
    ... 4 more
```

### 8.3.7.6 View in Repository Option

From Oracle JDeveloper, when you click the **View in Repository** option, it opens up the asset details in JDeveloper, which is for read-only purpose and the links within this page may not work.

As a workaround, you need to login to Oracle Enterprise Repository console and click links for further information.



---

---

## Using the Asset Editor

This chapter describes how to use the Asset Editor in Oracle Enterprise Repository to configure categories, relationships, rejection reasons, repositories, and vendors.

This chapter includes the following sections:

- [Overview](#)
- [Launching the Asset Editor](#)
- [Configuring System Options](#)
- [Advanced Asset Submission](#)
- [Asset/Compliance Template/Policy Migration](#)
- [Understanding the Asset Registration Process](#)
- [Understanding Asset Status](#)
- [Deleting an Asset](#)

### 9.1 Overview

The Asset Editor is used to administer assets and to configure assets and types. Categories, relationships, rejection reasons, repositories, and vendors are also configured within the Asset Editor.

The specifics of configuration depend on your organization's guidelines and may differ slightly from the options discussed here.

### 9.2 Launching the Asset Editor

This procedure is performed on the Oracle Enterprise Repository Assets screen.

---

---

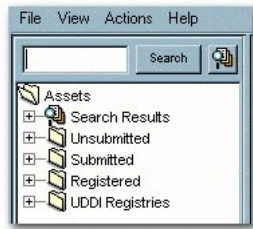
**Note:** The Asset Editor requires Java Web Start. See [Section 1.10, "Install Java Web Start"](#) for more information about installing and configuring Java Web Start.

---

---

1. Click **Edit/Manage Assets**. The Asset Editor is displayed, as shown in [Figure 9-1](#).

**Figure 9–1 Asset Editor**



## 9.3 Configuring System Options

A number of system options can be configured from the Actions menu in the Asset Editor.

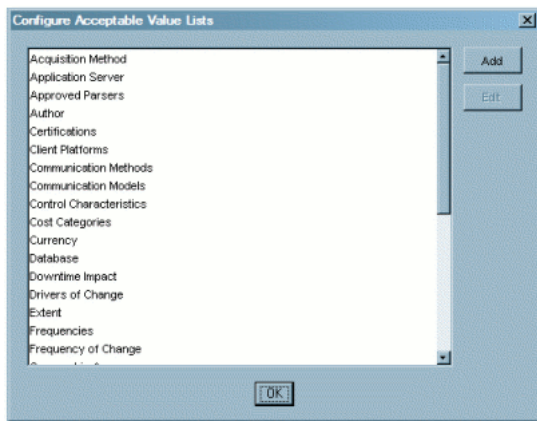
- [Section 9.3.1, "Configure Acceptable Value Lists"](#)
- [Section 9.3.2, "Configure Artifact Stores"](#)
- [Section 9.3.4, "Configure Categorizations"](#)
- [Section 9.3.5, "Configure Relationships"](#)
- [Section 9.3.6, "Configure Rejection Reasons"](#)
- [Section 9.3.7, "Configure Vendors"](#)

### 9.3.1 Configure Acceptable Value Lists

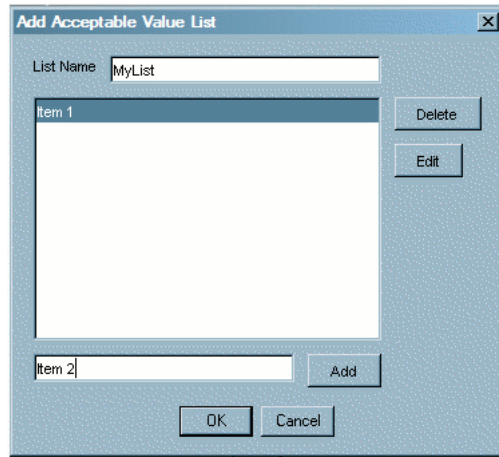
Defines/identifies the single- and multiple-selection lists used as metadata elements in asset displays.

1. Select **Configure Acceptable Value Lists** from the Actions menu. The Configure Acceptable Value Lists dialog is displayed, as shown in [Figure 9–2](#).

**Figure 9–2 Configure Acceptable Value Lists**



2. Click **Add**. The Add Acceptable Value List dialog is displayed, as shown in [Figure 9–3](#).
3. Enter an appropriate name in the **List Name** text box.

**Figure 9–3 Add Acceptable Value List Dialog**

4. Enter an appropriate list item name in the text box next to the Add button.
5. Click **Add**.
6. Repeat Steps 4 and 5 as necessary to add additional list items.
7. When finished, click **OK**. The new list appears in the Configure Acceptable Value Lists window.
8. To edit any Acceptable Value List, select the list from the Configure Acceptable Value Lists window.
9. Click **Edit**. The Edit Acceptable Value List dialog is displayed. (Same form as the Add Acceptable Value List window.)
10. Make changes as necessary.
11. Click **OK**.

### 9.3.2 Configure Artifact Stores

Determines where the files relevant to assets in Oracle Enterprise Repository are stored.

1. Select **Configure Artifact Stores** from the Actions menu. The Configure Artifact Stores dialog is displayed, as shown in [Figure 9–4](#).

**Figure 9–4 Configure Artifact Stores Dialog**



2. Click **Add**. The Create a New Artifact Store dialog is displayed, as shown in [Figure 9–5](#).

**Figure 9–5 Create a New Artifact Store Dialog**



3. Enter a name for the artifact store.
4. In the Type list, select a type for the artifact store.
  - Default Types:
    - FTP  
Accesses files on an FTP server
    - HTTP  
Accesses files on a web server
    - HTTPS  
Accesses files on a secure web server
    - UNC  
Accesses files using a Windows or Samba share
    - Raw URI  
Access files through a raw URI.
    - Raw SCM

Access files through a raw SCM. Enables the selection of SCM Types, including CVS, Perforce, and Other. The selected SCM type automatically populates the Download Path URI Suffix field, but the field can be modified however necessary.

5. Enter a hostname and path for the server.
6. (Optional) Enter a username and password. Exercising this option enables automatic login.
7. When finished, click **OK**.
8. To edit any Artifact Store, locate it in the list in the Configure Artifact Stores window.
9. Click **Edit**. The Edit Artifact Store dialog is displayed. (Same form as the Create a new Artifact Store window.)
10. Make changes as necessary.
11. Click **OK**.

### 9.3.3 Selecting a New Artifact Store

After configuring an artifact store, you must select it using the Submission Upload Artifact Store system setting on the Oracle Enterprise Repository Admin page.

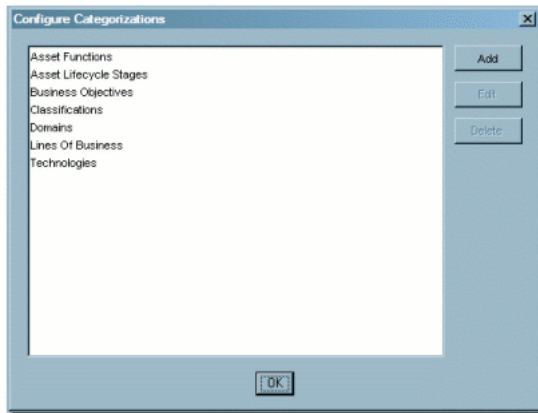
1. Open the Oracle Enterprise Repository Admin page.
2. In the left panel, click **System Settings**.
3. Locate the Upload Area section in the Server Settings group of system settings. See [Section 2.5.2, "Server Settings"](#) for more information.
4. Use the Submission Upload Artifact Store list to select the newly created artifact store.
5. Click **Save**.

### 9.3.4 Configure Categorizations

Categorizations determine how assets are organized and viewed. They can also be applied in a similar fashion to projects.

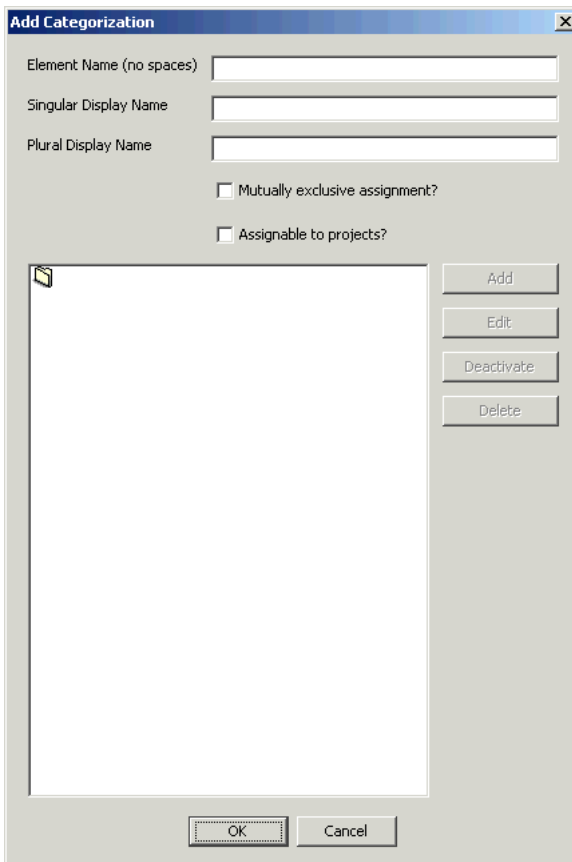
1. Select **Configure Categorizations** from the Actions menu. The Configure Categorizations dialog is displayed, as shown in [Figure 9-6](#).

**Figure 9–6 Configure Categorizations Dialog**



2. Click **Add**. The Add Categorizations dialog is displayed, as shown in [Figure 9–7](#).
3. Enter the appropriate information in each of the text boxes.

**Figure 9–7 Add Categorizations Dialog**

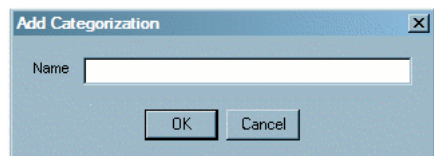


4. Select the **Mutually exclusive assignment?** option if assets can be assigned to only one of the categorizations in this categorization type.
5. Select the **Assignable to Projects?** option (if this option has been enabled by your Oracle Enterprise Repository Administrator) so categorizations can be assigned to projects, and not to assets.



6. If UDDI is enabled for your environment, then Oracle Enterprise Repository automatically creates tModelKeys for categorizations. If required, an administrator can assign specific UDDI Version 2 or Version 3 tModelKeys to any categorization.
7. Click **Add** to add sub-categorizations. The Add Categorizations dialog is displayed, as shown in [Figure 9-8](#).

**Figure 9-8 Add Categorizations Dialog**



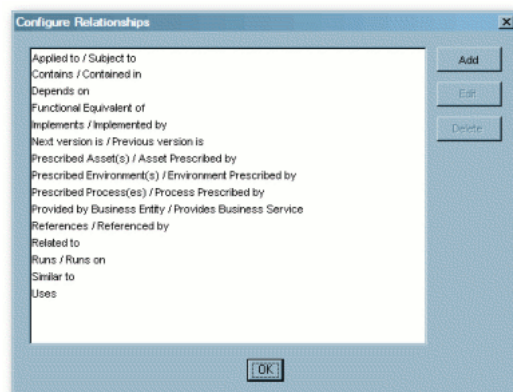
8. When finished, click **OK**. The new categorization appears in the Configure Categorizations window.
9. To edit any Categorization, locate it in the list in the Configure Categorizations window.
10. Click **Edit**. The Edit Categorization dialog is displayed. (Same form as the Add Categorizations window.)
11. Make changes as necessary.
12. Click **OK**.

### 9.3.5 Configure Relationships

Relationships define the connection, interaction, or interdependence between assets. If so configured, assets subject to a specific relationship to an asset that has been selected for use (that is, the Use - Download button has been clicked) can also be downloaded.

1. Select **Configure Relationships** from the Actions menu. The Configure Relationships dialog is displayed, as shown in [Figure 9-9](#).

**Figure 9-9 Configure Relationships Dialog**



2. Click **Add**. The Add Relationships dialog is displayed, as shown in [Figure 9-10](#).
3. Enter the appropriate information in each of the text boxes.

**Figure 9–10 Add Relationship Dialog**

4. Make the appropriate selection in the Direction list.  
The selected direction affects the information displayed in the Source Asset and Target Asset text boxes. For example, in a two-way relationship, the target and source are interdependent, whereas Two-way -- order matters can be used to describe the relationship between sequential versions of an asset, for example, the relationship between Asset v0.90 and Asset v1.0.
5. Make the appropriate selection in the **Make target assets available for download with the source asset** check box.
6. When finished, click **OK**.
7. To edit any Relationship, locate it in the list in the Configure Relationships window.
8. Click **Edit**. The Edit Relationships dialog is displayed. (Same form as the Add Relationship window.)
9. Make changes as necessary.
10. Click **OK**.

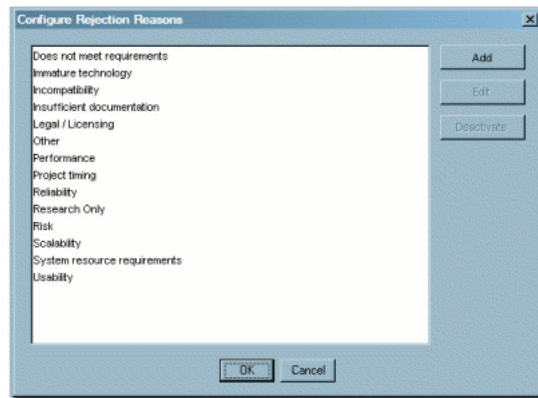
### 9.3.5.1 Configure System-Supplied Relationships

There are also system-supplied relationships in Oracle Enterprise Repository with limited read-only functionality to end users. Unlike standard asset relationships, system-supplied relationships are used in a context that requires their name and direction to be immutable. Therefore, these relationships cannot be added, deleted, or modified by users, except where otherwise noted.

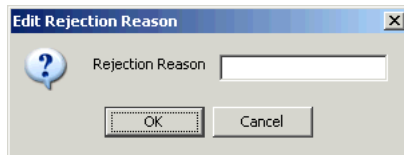
## 9.3.6 Configure Rejection Reasons

Determines the selection of Rejection Reasons available to users who reject previously used/downloaded assets. Rejection Reasons provide valuable local community feedback on assets.

1. Select **Configure Rejection Reasons** from the Actions menu. The Configure Rejection Reasons dialog is displayed, as shown in [Figure 9–11](#).

**Figure 9–11 Configure Rejection Reasons Dialog**

2. Click **Add**. The Edit Rejection Reasons dialog is displayed, as shown in [Figure 9–12](#).

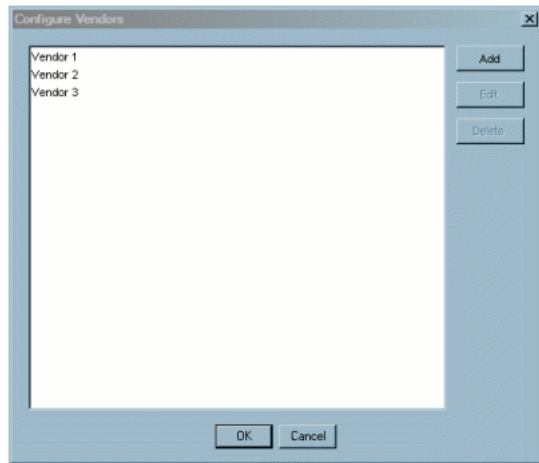
**Figure 9–12 Edit Rejection Reason Dialog**

3. Enter the appropriate text in the **Rejection Reason** text box.
4. When finished, click **OK**.
5. To edit any Rejection Reason, locate it in the list in the Configure Rejection Reasons window.
6. Click **Edit**. The Edit Rejection Reasons dialog is displayed.
7. Make changes as necessary.
8. Click **OK**.

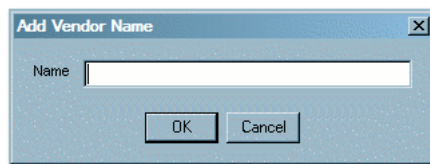
### 9.3.7 Configure Vendors

Defines/identifies vendors providing assets to the repository.

1. Select **Configure Vendors** from the Actions menu. The Configure Vendors dialog is displayed, as shown in [Figure 9–13](#).

**Figure 9–13 Configure Vendors Dialog**

2. Click **Add**. The Add Vendor Name dialog is displayed, as shown in [Figure 9–14](#).

**Figure 9–14 Add Vendor Name Dialog**

3. Enter the appropriate information in the **Name** text box.
4. When finished, click **OK**.
5. To edit any Vendor, locate it in the list in the Configure Vendors window.
6. Click **Edit**. The Edit Vendor Name window opens. (Same form as the Add Vendor Name window.)
7. Make changes as necessary.
8. Click **OK**.

## 9.4 Advanced Asset Submission

This function enables a user (with the appropriate privileges) to submit assets through the Asset Editor rather than through the Web Console. This enables the user to enter more asset details upon submission. This procedure is performed in the Asset Editor. (Access to the Asset Editor is determined by the role to which the user is assigned.)

1. Open the File menu in the Asset Editor and select **New**. The Create a New Asset dialog is displayed, as shown in [Figure 9–15](#).

**Figure 9–15 Create a New Asset Dialog**

2. Enter the appropriate information in the **Name** and **Version** text boxes.
3. Select the appropriate Type template from the Type list, as shown in [Figure 9–16](#).

**Figure 9–16 Create a New Asset Dialog - Type List**

4. Click **OK**. The new asset appears in the Asset Editor, as shown in [Figure 9–17](#).

**Figure 9–17 Asset Editor - New Asset**

The asset is submitted for review and registration.

## 9.5 Asset/Compliance Template/Policy Migration

Migration refers to the transfer of metadata and files from one asset, compliance template, or Policy to another, usually of the same type. Since it is easier to edit metadata than to recreate it, this function enables users to quickly create different versions of assets, compliance templates, or policies. For example, to create JavaComponent v1.0, one would migrate the metadata and files from JavaComponent v0.90, then make the necessary changes in the metadata for the v1.0 asset. When

migrating an asset the new asset must have a different name, version number, or both. Migration does NOT remove the original asset from Oracle Enterprise Repository.

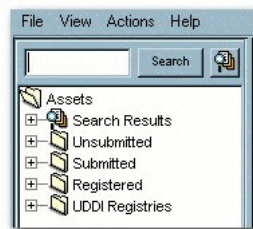
It is also possible to migrate an asset, compliance template, or policy to a different type. However, differing metadata elements does not transfer. For example, if the originating asset type uses a table on one of its tabs and is migrated to an asset type that does not have the same table element, the table and its metadata does not migrate. The table metadata element would have to be created in the new asset type.

### 9.5.1 Migrating an Asset/Compliance Template/Policy

This procedure is performed in the Asset Editor.

1. Use **Search** or browse the Asset Editor file tree to locate the source asset or compliance template to be migrated, as shown in [Figure 9–18](#).

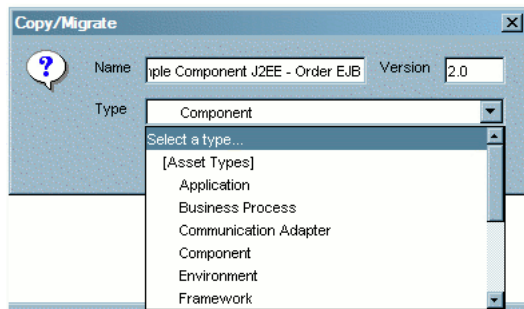
**Figure 9–18 Asset Editor - Search Button**



2. Select **Copy/Migrate** in the File menu.

The Copy/Migrate dialog is displayed, as shown in [Figure 9–19](#).

**Figure 9–19 Copy/Migrate Dialog**

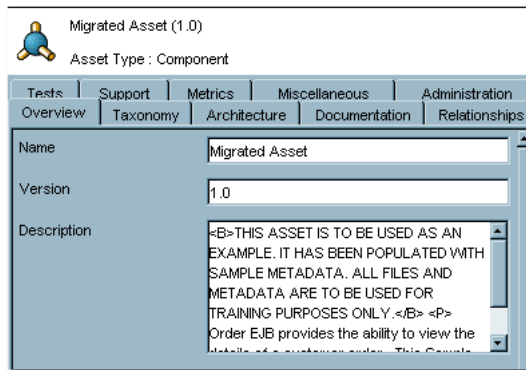


3. Enter the appropriate information in the **Name** and **Version** text boxes.
4. Select the destination Type (the type to which the source artifact is to be migrated) in the Type list.

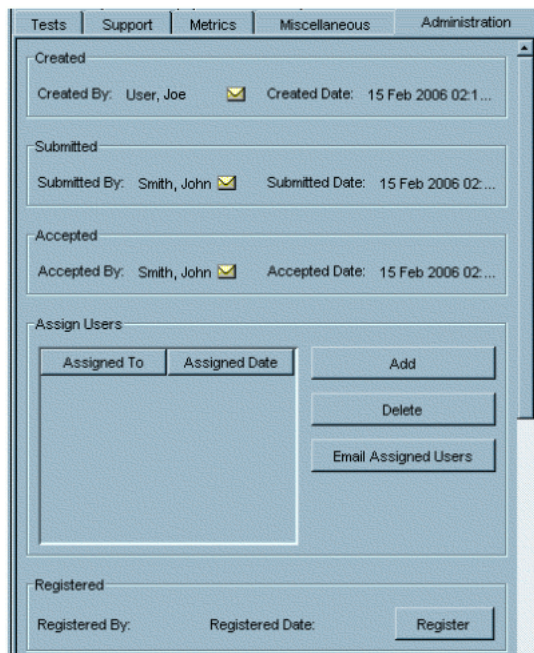
Migrating assets between dissimilar types affects the display of asset information in the asset detail and in the Asset Editor.

5. When finished, click **OK**.

The migrated asset appears in the Asset Editor, as shown in [Figure 9–20](#).

**Figure 9–20 Asset Editor - Overview Tab**

6. Click the new asset's **Administration** tab, as shown in [Figure 9–21](#).

**Figure 9–21 Administration Tab**

7. Register the new asset or assign it to another user for registration.

## 9.6 Understanding the Asset Registration Process

Assets moving through the registration process are organized and managed via several folders, as displayed in the file tree in the Asset Editor.

An asset begins the registration process in the Submitted folder. Assets can be submitted via the Oracle Enterprise Repository Web Interface, as described in "Performing a Standard Asset Submission" in the *Oracle Fusion Middleware Developer's Guide for Oracle Enterprise Repository*, or via the Asset Editor, as described in [Section 9.4, "Advanced Asset Submission"](#). Once accepted or rejected by the registrar, the asset moves to the Under Review folder, under Submitted.

Pending registrar review and approval of the data on the tabs in the Asset Editor, the asset moves from Under Review to the Registered folder. Users can track the progress



of assets by using the Search function, which accesses Submitted, Unsubmitted, and Registered assets, or by using My Stuff.

The registration process includes the following actions:

- **Submission:** An asset is submitted by a user and appears in the Pending Review folder under Submitted. An automatic email message alerts the registrar that a new asset has entered the submission queue.
- **Review:** The registrar examines the asset and its associated information and makes a decision to enter it into the work queues or reject it.
- **Rejection:** If the asset is rejected, the registrar enters a reason for the rejection.  
When an asset is rejected from the Submissions folder, it is removed and marked as rejected in the submitter's My Submissions folder in My Stuff.
- **Acceptance:** Assets accepted for registration move to the Under Review folder, and the registrar or advanced submitter begins the registration process. The required information is gathered and entered on the appropriate tabs in the Asset Editor. The registrar examines each tab and monitors the workflow. When information for a specific stage of the workflow is acceptable, the registrar approves the data on the appropriate tab. There is no prescribed order in the approval process; the registrar can approve any stage in any order. The registrar also has the option to edit any of the information for any stage of the process.
- **Approval:** The registrar grants final approval on the Administration tab, based on organizational standards regarding the information supplied on each of the various tabs. The specific configuration of Asset Editor tabs for any asset is determined by the Type template to which the asset is assigned on submission. Each tab provides various elements for metadata that is used to describe the asset and facilitate its use.

This section describes the following topics:

- [Section 9.6.1, "Accepting a Submitted Asset"](#)
- [Section 9.6.2, "Registering an Asset"](#)
- [Section 9.6.3, "Adding a Note to an Asset"](#)

## 9.6.1 Accepting a Submitted Asset

Submissions must be accepted and assigned to one or more registrars before assets can be processed and registered in Oracle Enterprise Repository.

This procedure is performed in the Asset Editor. (Requires appropriate permission.)

To accept a submitted asset:

1. Open the **Submitted** folder.
2. Open the **Pending Review** folder.
3. Select the asset to be registered.
4. To complete the registration process:
  - To accept without assigning the asset to a user, click **Accept**. The asset moves to the Under Review folder in the tree, and also appears in each of the workflow folders under the asset. The workflow folders correspond to tabs in the Asset Editor
  - To accept and assign to specific users, click **Accept and Assign**. The Assigning Users dialog is displayed. Use the arrow buttons to move users between the



**Available Users** and **Selected Users** columns, and then click **OK**. The asset moves to the Under Review folder in the tree and is assigned to the selected user/users, who will provide the information required for each of the tabs in the Asset Editor

## 9.6.2 Registering an Asset

After an asset has been accepted, a registrar must review it and register it for use in Oracle Enterprise Repository

This section describes the following topics:

- [Section 9.6.2.1, "Approving an Asset's Tabs"](#)
- [Section 9.6.2.2, "Completing the Registration Process"](#)

### 9.6.2.1 Approving an Asset's Tabs

You should review the information on an asset's tabs before registering it.

While certain tabs are common to all asset types, the specific Asset Editor tabs for any asset are determined by the configuration of the Type template to which the asset is assigned on submission. Similarly, the metadata elements that appear on any tab are also determined by the Type configuration. While the specific tabs and elements may vary from Type to Type, the approval process for each tab involves the entry and/or review of the information in each element.

To approve an asset's tabs:

#### Task 1 Approve the Overview tab

To approve the Overview Tab:

1. Click the **Overview** tab in the selected asset.
2. Enter the appropriate information in each of the fields.
3. Click **Approve**. The text in the Overview tab label changes color and the **Approve** button changes to **Unapprove**.

---



---

**Note:** Approval buttons in the Asset Editor are visible only to users with the appropriate permissions.

---



---

#### Task 2 Approve the Taxonomy Tab

To approve the Taxonomy Tab:

1. Click the **Taxonomy** tab in the selected asset. Several Categorizations are displayed.
2. Click the **Assign** button in the Classification section. The Assign Classifications dialog is displayed.
3. Use the radio buttons to select the appropriate classification. The available options are:

**Raw:** No assurance of quality or completeness.

**Educational:** For educational/informational purposes only. Not approved for use in projects.

**Approved:** Approved by the registrar for use in projects.

**Recommended:** Approved and successfully deployed in at least one project.

**Mandated:** Must be used whenever a project requires the functionality the asset provides. This is especially relevant for Web services that access customer data, process payments, etc.

4. Click **OK**. The selections are listed in the Classifications section.

---

---

**Note:** Default Categories may be customized to reflect your environment.

---

---

5. Repeat the process for each section in the Taxonomy tab.
6. Enter appropriate keyword in the text box in the **Keywords** section, and then click **Add**. The new keyword appears in the Keywords list. Add other keywords as necessary.
7. When the Taxonomy tab is completed, click **Approve**. The text in the Overview tab label changes color and the **Approve** button changes to **Unapprove**.

### **Task 3 Approve the Documentation Tab**

To approve the Documentation Tab:

1. Click the **Documentation** tab in the selected asset. A number of suggested document titles are listed in the Documentation window. Appropriate documentation may be associated with each of these titles, and new documents may be added to the list.
2. Click **Add**. The Edit dialog is displayed.
3. Enter the appropriate information in the **Name** text box.
4. Click the **Edit** button next to the URL text box. The Edit URL dialog is displayed.
5. Select **Repository File** or **External File**, as appropriate.
6. Enter all necessary information in the available text boxes.
7. Click **OK**. The new document appears in the list.
8. To edit file information for an existing document, select the document, click **Edit**, and repeat Steps 3-7.
9. When finished, click **Approve**. The text in the Documentation tab label changes color and the **Approve** button changes to **Unapprove**.

### **Task 4 Approve the Relationships Tab**

To approve the relationships tab:

1. Click the **Relationships** tab in the selected asset.
2. Click the **Add** button. The Add Relationships dialog is displayed.
3. Use the **Search** or **List All Active** buttons to display assets in the Asset section of the dialog.
4. Select an asset from the list.
5. Use the **Relationship Type** list to select the appropriate relationship between the two assets.
6. Click **OK**.

7. Add other asset relationships as necessary.
8. When finished, click **Approve**. The text in the Relationships tab label changes color and the **Approve** button changes to **Unapprove**.

### Task 5 Approve the Administration Tab

Every asset in Oracle Enterprise Repository has an Administration tab. Use the Administration tab to:

- Track the asset Created, Submitted, Accepted workflow.
- Assign users to review and approve information on the other tabs.
- Change the asset's status. See [Section 9.7, "Understanding Asset Status,"](#) for more information.
- View assets notes and reviews.

To approve the Administration Tab, click the **Register** button. This asset will be registered after you complete the registration process, as detailed in [Section 9.6.2.2, "Completing the Registration Process"](#).

---



---

**Note:** A registered asset can include unapproved tabs.

---



---

#### 9.6.2.2 Completing the Registration Process

After the tabs have been approved (specifically the Administration tab, as discussed in [Task 5, "Approve the Administration Tab"](#)), you complete the registration process by clicking **Save** from the Asset Editor's File menu. The asset is now registered. Users with appropriate permissions can now discover it via Oracle Enterprise Repository.

### 9.6.3 Adding a Note to an Asset

You can add notes to assets from the Asset Editor.

To add a note to an asset:

1. Click **Add Note** in the Asset Editor File menu. The Add a Note for... dialog is displayed.
2. Enter the appropriate information in the text box.
3. Click **OK**. The note is added to the list in the Logs section on the asset's Administration tab. (It may be necessary to click the **Refresh** button in the Logs section.)

## 9.7 Understanding Asset Status

The registrar can assign a status to the asset using the Administration tab in the Asset Editor.

- **Active**  
The status assigned to any asset that is available for use.
- **Inactive**  
The status assigned to any asset in the repository but unavailable for general use. An inactive asset can be re-activated or retired. Inactive assets show up in a search, but cannot be selected for use (the Use - Download button is not displayed).
- **Retired**

The status assigned to any asset in the repository that is retired from general use. A retired asset can be re-activated or made inactive. Retired assets cannot be selected for use (the Use - Download button is not displayed), nor do they show up in a search; however, they would be available as related assets to active or inactive assets in the repository.

- Deleted

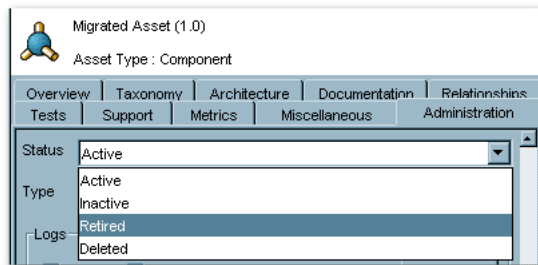
The status assigned to any asset that is removed from the repository. Deleted assets cannot be recovered or have their status reassigned.

### 9.7.1 Assigning or Changing an Asset's Status

This procedure is performed in the Asset Editor. Any changes to an asset's status are automatically saved by the Asset Editor.

1. Use **Search** or browse the Asset Editor file tree to locate the source asset or compliance template to be modified.
2. Click the **Administration** tab.
3. In the Status list, select the status to be assigned to the asset, as shown in [Figure 9-22](#).

**Figure 9-22 Status List**




---

**Note:** Depending on the assigned status, the asset may disappear from the tree (Deleted) or be unavailable for user searches (Retired).

---

## 9.8 Deleting an Asset

To delete an asset in Oracle Enterprise Repository, perform the following steps:

---

**Note:** Only users whose role allows them to launch the Asset Editor (users who have been granted the "Launch Asset Editor" permission are able to delete assets that they can view. See "User Role Descriptions" in the *Oracle Fusion Middleware Developer's Guide for Oracle Enterprise Repository* for more information about user roles and their associated privileges.

---



---

**Note:** This is a new feature in OER 12c. If you are upgrading from a previous version of OER, it is important to understand that users granted the "Launch Asset Editor" permission in previous versions will now be able to delete assets.

---

1. In the left panel, click **Assets**.
2. Click **List All**, or search for assets that you want to delete.
3. To delete an asset from the Search Results screen, select the check box next to each asset that you want to delete, and then click **Delete**. You can delete multiple assets simultaneously using this method:



# Part III

---

## Securing Oracle Enterprise Repository

This part describes security tasks for Oracle Enterprise Repository.

This part includes the following chapters:

- [Chapter 10, "Configuring Oracle Enterprise Repository to use External Authentication Tooling"](#)
- [Chapter 11, "Configuring Advanced Role-based Access Control"](#)
- [Chapter 12, "Password Encryption"](#)





---

## Configuring Oracle Enterprise Repository to use External Authentication Tooling

This chapter describes how to authenticate Oracle Enterprise Repository using the external authentication tools, such as LDAP / Active Directory.

This chapter contains the following sections:

- [Section 10.1, "Overview"](#)
- [Section 10.2, "Single Sign-On"](#)
- [Section 10.3, "Container Managed Setup"](#)

### 10.1 Overview

This section describes how to use LDAP/Active Directory to authenticate the users in Oracle Enterprise Repository.

This section contains the following topics:

- [Section 10.1.1, "LDAP/Active Directory"](#)
- [Section 10.1.2, "Enable LDAP Integration System Properties"](#)
- [Section 10.1.3, "Modify LDAP/Active Directory Properties"](#)
- [Section 10.1.4, "Security Considerations"](#)
- [Section 10.1.5, "LDAP Property Examples"](#)

#### 10.1.1 LDAP/Active Directory

When utilizing LDAP / Active Directory (AD) to authenticate users, some consideration must be given to the user's department and role configurations prior to the utilization of LDAP / Active Directory server. All users are authenticated through LDAP/AD once the integration is enabled; it is essential to have at least one admin-level user account created within the Oracle Enterprise Repository database that matches the username from LDAP/AD. This user account should be assigned the admin role so that administrative functions within Oracle Enterprise Repository can still be performed when LDAP/AD is enabled.

If role synchronization is enabled from LDAP/AD, at least one user account should be assigned an administrative-level role. LDAP roles should be created and appropriate Oracle Enterprise Repository permissions assigned within Oracle Enterprise Repository prior to utilizing the role-synch option. Role names are synchronized on name only. When the administrative-level user logs into Oracle Enterprise Repository

using LDAP/AD, that person has the ability to configure and administer the application properly. This user account should NOT be used for daily activities.

## 10.1.2 Enable LDAP Integration System Properties

This procedure is performed on the Oracle Enterprise Repository Admin screen.

1. Click **System Settings** in the left pane. The System Settings section opens in the main pane.
2. Use the System Settings Search box to easily locate each of the following settings.
3. Enter `enterprise.authentication.ldap.enabled` into the Search box. Set the value to **True** and click **Save**.

Change the settings as indicated below:

- Unapproved User Login
  - Set to **True**.
- Plug-in Login Module Class Name
  - Enter `com.flashline.enterprise.authentication.server.loginmodule.LDAPLogin` in the text box.

---

---

**Note:** This property turns LDAP on/off. Once enabled, the application uses LDAP server for user authentication.

---

---

- Plug-in Login Module Display Name
    - Enter **Enterprise LDAP Login** in the text box.
  - Plug-in Login Module
    - Set to **True**.
  - Plug-in Login Module - Internal Checking
    - Set to **False**.
4. Click **Save** when finished.

## 10.1.3 Modify LDAP/Active Directory Properties

You can modify LDAP/AD properties, as follows:

1. Click **System Settings** in the left pane.
  2. Use the System Settings Search to easily locate each of the following settings. Enter the values as indicated below:
    - LDAP Server Host Name
      - In the text box, enter the Host name, or the directory server IP address.
    - LDAP Server Port Number
      - Enter 389 in the text box.
    - LDAP Mask
      - Enter `uid=^` for LDAP
- Or

- Enter `samAccountName=^` for Active Directory
- Creation of Unapproved User Accounts
  - Set to `True`.
- Assign default roles to users
  - Set to `True`.

---

---

**Note:** This property assigns default roles on every user authentication.

---

---

- Auto create missing roles
  - Set to `True`.

---

---

**Note:** This property creates role synchronized from the LDAP/AD server, but does NOT assign any permissions to those roles.

---

---

- Auto create missing departments
  - Set to `True`.

---

---

**Note:** This property creates departments synchronized from the LDAP/AD server, but does not assign any description to those departments. However, the user is assigned to the new role.

---

---

- LDAP Version
  - Enter 3 in the text box. (Supported versions are 2 and 3)
- Administrator Account Distinguished Name

---

---

**Note:** This property is required with using Active Directory. This property must contain a DN of a user account with at least read-only directory look-up permissions.

---

---

Example: `CN=Some_User,CN=Users,DC=ad,DC=example,DC=com`

- Administrator Account Password
  - In the text box, enter the password for the administrator account identified in the Administrator Account Distinguished Name property, above.
- Use SSL Connection
  - Set to `True` to enable an SSL connection for LDAP. The default value is `false`.
- Follow referrals
  - Set to `True`.
- Retrieve data using the admin account
  - Set to `False` for LDAP (if applicable) Or
  - Set to `True` for Active Directory or restricted LDAP environments.

- Search Start Location

---

---

**Note:** This property defines where in the directory tree the search for user records begins.

---

---

Examples:

- For LDAP: OU=MemberGroupB, O=en\_us
- For Active Directory: CN=Users,DC=ad,DC=example,DC=com

- Search Scope

- Select subtree in the list.

---

---

**Note:** This property defines the depth (below the baseDN) of user record searches.

---

---

- Attribute Name that Identifies a Found Entry

---

---

**Note:** This property designates the attribute name that uniquely identifies the user account within the scope of the tree search.

---

---

- For LDAP: uid

Or

- For Active Directory: samAccountName

- Found Entry Email Attribute Name

- Enter mail

- Found Entry First Name Attribute Name

- Enter givenName

- Found Entry Middle Name Attribute Name

- Enter the middle name attribute from your LDAP or Active Directory (if applicable)

- Found Entry Last Name Attribute Name

- Enter sn

- Found Entry Telephone Number Attribute Name

- Enter telephoneNumber

- Use LDAP Departments

- Set to True

---

---

**Note:** This property defines the user's department attribute value that is synchronized within Oracle Enterprise Repository.

---

---

- Department Attribute

- Enter department

- Use LDAP Roles
  - Set to `False`
- Role Attribute
  - Enter the LDAP / Active directory attribute that contains the role information for the user.
- Second Level Lookup Attribute

---



---

**Note:** This property defines the attribute that identifies a second-level lookup to retrieve user info; the value must be a DN. If you are using a redirect for second level lookups, define the base DN for this second lookup.

---



---

3. Click **Save** when finished.
4. Restart the Oracle Enterprise Repository application.

## 10.1.4 Security Considerations

Using the Oracle Enterprise Repository LDAP/Active Directory Connector allows LDAP to act as the single source of user identification for Oracle Enterprise Repository user authentication and role assignment. However, this does not prevent respective host repositories from managing user authentication for access to files through Oracle Enterprise Repository.

When using the Oracle Enterprise Repository LDAP/Active Directory Connector, Oracle Enterprise Repository depends on LDAP or Active Directory to authenticate users. The username/password combination is delegated to the LDAP system as a bind request. The user is authenticated only if the bind request is successful.

As an option, LDAP can be configured to store/retrieve Oracle Enterprise Repository user role assignments. In this configuration, at each user login Oracle Enterprise Repository synchronizes with the user's roles as stored in LDAP. Roles are added directly through LDAP, and are not managed by Oracle Enterprise Repository; the group mappings within the LDAP are not used by OER. To use this option, the `ldap.enable-synch-roles` system settings must be set to `True`. The `ldap.rbac.roleAttrib` system setting must be set to the name of the multi-value LDAP attribute that contains the role names that the user has within Oracle Enterprise Repository. If the `ldap.enable-synch-roles` system setting is not enabled, or if the LDAP attribute referenced by the `ldap.rbac.roleAttrib` option is not configured, an administrator manually assigns user roles in Oracle Enterprise Repository.

### 10.1.4.1 Use Case Sample Scenarios

The following scenarios illustrate a selection of LDAP setups and configurations in order to clarify property settings for user management.

#### Scenario 1

Prevent user access to Oracle Enterprise Repository despite LDAP authentication. Access is provided only to pre-existing users with active Oracle Enterprise Repository accounts.

- Rationale
  - Non-enterprise license agreements where user base is predefined and number of users allowed into the application is limited.

- Property Settings
  - **ldap.allow-user-creation**
    - \* Set to False
  - **enterprise.security.unapproveduser.allowlogin**
    - \* Set to False

### Scenario 2

On LDAP authentication, create a default Oracle Enterprise Repository user account and assign the default role(s), but deny the user access to the Oracle Enterprise Repository.

In Oracle Enterprise Repository, the default user is "user".

- Rationale
  - To deny Oracle Enterprise Repository access to a new user until the security administrator is notified that the new user account was created. Once approved by the security administrator, the user's status is changed to active, allowing Oracle Enterprise Repository login.
- Property Settings
  - **ldap.allow-user-creation**
    - \* Set to True
  - **ldap.assign-default-roles**
    - \* Set to True
  - **enterprise.security.unapproveduser.allowlogin**
    - \* Set to False

### Scenario 3

On LDAP authentication, a default Oracle Enterprise Repository user account is created with the default role(s), and the user is permitted to login to the Oracle Enterprise Repository.

- Rationale
  - An enterprise license agreement in which LDAP authentication is the only restriction on new user creation. Typically, the default Oracle Enterprise Repository role would be set to User in order to limit access for new users whose roles are not predefined by an LDAP account.
- Property Settings
  - **ldap.allow-user-creation**
    - \* Set to True
  - **ldap.assign-default-roles**
    - \* Set to True
  - **enterprise.security.unapproveduser.allowlogin**
    - \* Set to True

## 10.1.5 LDAP Property Examples

Since limitations in Active Directory prevent searches below the top-level of the directory while anonymously bound (not authenticated) to the directory server, Oracle Enterprise Repository user information lookup requires the Bind DN, Bind Password, and Retrieve Data As Admin properties to be set with appropriate values. [Table 10-1](#) describes the LDAP / Active Directory properties with its examples.

**Table 10-1 Active Directory and Traditional LDAP Properties**

Active Directory		Traditional LDAP (InetOrgPerson)	
ldap.host	ad.example.com	ldap.host	ad.example.com
ldap.port	389	ldap.port	389
ldap.version	3	ldap.version	3
ldap.bindDN	CN=Some_ User,OU=Users,DC=ad,DC=example,DC=com	ldap.bindDN	(required if Anonymous lookups are disabled)
ldap.bindPassword	password	ldap.bindPassword	(required if Anonymous lookups are disabled)
ldap.retrieve-data-as-admin	true	ldap.retrieve-data-as-admin	false (TRUE if anonymous lookups are disabled)
ldap.mask	sAMAccountName=^	ldap.mask	uid=^
ldap.baseDN	CN=Users,DC=ad,DC=example,DC=com	ldap.baseDN	OU=MemberGroupB, O=en_us
ldap.scope	subtree	ldap.scope	one
ldap.uniqueIDAttrib	samAccountName	ldap.uniqueIDAttrib	uid
ldap.emailAttrib	mail	ldap.emailAttrib	mail
ldap.givennameAttrib	givenname	ldap.givennameAttrib	givenName
ldap.surnameAttrib	sn	ldap.surnameAttrib	sn
ldap.telephoneAttrib	telephonenumber	ldap.telephoneAttrib	telephoneNumber
ldap.deptAttrib	department	ldap.deptAttrib	department

[Table 10-2](#) describes the custom and common LDAP properties.

**Table 10-2 Custom and Common LDAP Properties**

Custom and Common Properties Regardless of implementation	
ldap.deptAttrib	department
ldap.rbac.roleAttrib	roles
ldap.allow-user-creation	true
ldap.enable-synch-roles	true
ldap.enable-synch-depts	true

## 10.2 Single Sign-On

This section describes how to use Single Sign-On to authenticate the users in Oracle Enterprise Repository.

This section contains the following topics:

- Section 10.2.1, "Overview"
- Section 10.2.2, "Configure Oracle Enterprise Repository For Use With Single Sign-On Authentication"
- Section 10.2.3, "Advanced Options"

## 10.2.1 Overview

The Oracle Enterprise Repository Advanced Container Authentication LoginModule is used to accept user credentials passed by HTTP Request Headers (potentially populated by an SSO system). This feature allows integration with single-sign-on systems.

## 10.2.2 Configure Oracle Enterprise Repository For Use With Single Sign-On Authentication

You can access the following configuration properties only using the administrator rights:

---

---

**Note:** This enhancement allows AdvancedContainerLogin Module to accept user information in SOAP Headers for the AuthtokenCreate REX API method. The username is passed in a SOAP Header with a name that is identified by the Oracle Enterprise Repository system setting `enterprise.container.auth.username` and has a namespaceUri of `www.oracle.com/oer`. The value of the SOAP Header is the username of the user. If the username is not passed within a SOAP Header, then the Oracle Enterprise Repository system setting `enterprise.loginmodules.fallbackauthentication` is used. If `enterprise.loginmodules.fallbackauthentication` is true, then the user is authenticated by the configured PluggableLoginModule for the specified username/password.

---

---

Plugin in Login module is a configuration set up to configure Database login Module, LDAP Login Module, and Container Login Module. Container Login module can be Container Managed Login Module or Advanced Container Login Module i.e. SSO. These can be configured in the System Settings tab.

This section contains the following topics:

- Section 10.2.2.1, "Enable Single Sign-On Integration System Properties"
- Section 10.2.2.2, "Using the Oracle Enterprise Repository SSO Integration with Basic Authentication"
- Section 10.2.2.3, "Modify Application Property Files Manually"

---

---

**Note:** The Fallback authentication works only with REX API.

---

---

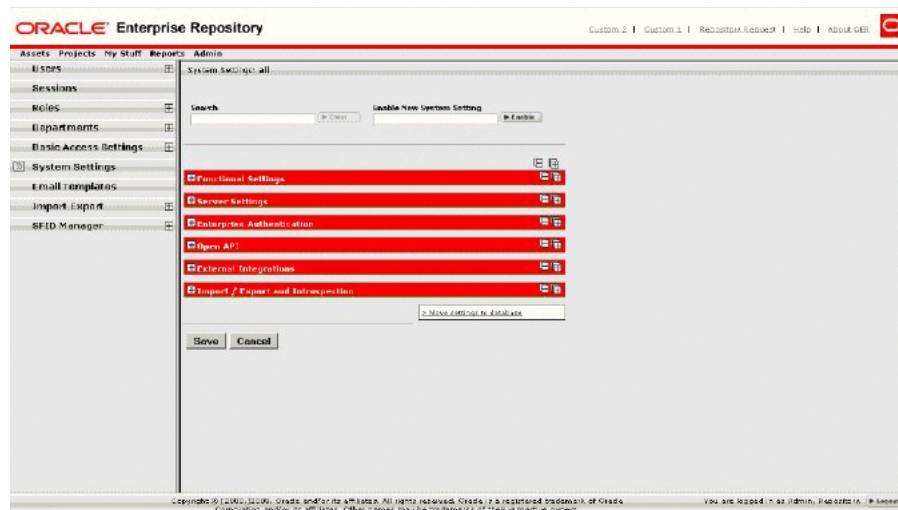
### 10.2.2.1 Enable Single Sign-On Integration System Properties

This procedure is performed on the Oracle Enterprise Repository Admin screen.

1. Click **System Settings** in the left pane. The Oracle Enterprise Repository System Settings screen is displayed, as shown in Figure 10–1.



**Figure 10-1 Oracle Enterprise Repository System Settings Screen**

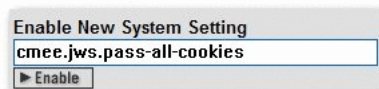


This figure illustrates the Oracle Enterprise Repository System Settings pane.

\*\*\*\*\*

2. Enter `enterprise.authentication.advancedcontainer.enabled` into the Search box. Set the value to `True` and click **Save**.
3. Enter `cmee.jws.pass-all-cookies` in the Enable New System Setting text box, as shown in Figure 10-2.

**Figure 10-2 Enable New System Settings Dialog**

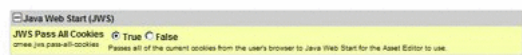


This figure illustrates the Enable New System Settings dialog.

\*\*\*\*\*

4. Click **Enable**. JWS Pass All Cookies appears in the Java Web Start (JWS) section of the Server Settings group of system settings, as shown in Figure 10-3.

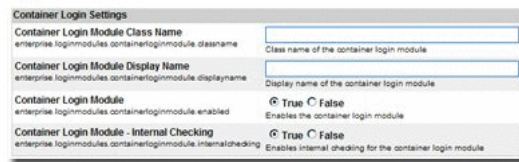
**Figure 10-3 JWS Pass All Cookies**



This figure illustrates the JWS Pass All Cookies pane.

\*\*\*\*\*

5. Make sure the property is set to `True`.
6. Click **Save**.
7. Enter `container.login.module` in the System Settings Search text box. The Container Login Module section is displayed in the Enterprise Authentication group of system settings, as shown in Figure 10-4.

**Figure 10–4 Container Login Settings Section**

This figure illustrates the Container Login Settings pane.

\*\*\*\*\*

8. Modify the following properties as indicated:
  - **Container Login Module Class Name**
    - Enter `com.oer.enterprise.authentication.server.loginmodule.AdvancedContainerLogin` in the text box.
  - **Container Login Module Display Name**
    - Enter `Advanced Container Login Module` in the text box.
  - **Container Login Module**
    - Set the property to `True`.
9. Supply SSO Header Values as indicated (these are often called Responses within the Policy Server). Data types expected, and possible values are listed below the header name. The expected value types apply to the responses supplied by the policy server:
  - **Username Header Name**
    - Set this property to the Name of the header that contains the user's UID value.  
This header should contain the user's user id (REQUIRED).
  - **Firstname Header Name**
    - Set this property to the Name of the header that contains the user's First Name value. (Alpha String)  
This header should contain the user's proper name.
  - **Middlename Header Name**
    - Set this property to the Name of the header that contains the user's Middle Name value. (Alpha String)  
This header should contain the user's middle name.
  - **Lastname Header Name**
    - Set this property to the Name of the header that contains the user's Last Name value. (Alpha String)  
This header should contain the user's surname.
  - **Status Header Name**
    - Set this property to the Name of the header that contains the user's Active Status value.

This header should contain a valid integer value specifying the user's status within OER. Refer to the following table for valid values (REQUIRED).

\* 00 - Active

\* 10 - Unapproved

\* 20 - Locked Out

\* 30 - Inactive

- Email Header Name
    - Set this property to the Name of the header that contains the user's Email value.
 

This header should contain the user's e-mail address (REQUIRED).
  - Phone Header Name
    - Set this property to the Name of the header that contains the user's Phone Number value.
 

This header should contain the user's phone number.
  - Roles Header Name
    - Set this property to the Name of the header that contains the user's Role(s) value.
 

This header should contain the user's role(s).
  - Department Header Name
    - Set this property to the Name of the header that contains the user's Department(s) value.
 

This header should contain the user's department(s).
10. Update the behavior of the SSO module with the following properties:
- Use Container passed Departments
    - Set this value to `True` if you would like to synchronize the user's department from the policy server responses.
  - Departments passed within single header
    - Set this value to `True` if more than one department name is passed as a Policy Server response.
  - Department Delimiter
    - Set the value of this property to the character that will delimit multiple departments within the single department header. This field can accept Unicode notations such as `\u0020` for a space.
  - Use Container passed Roles
    - Set this value to `True` if you would like to synchronize the user's roles from the policy server responses. (NOTE: Setting this value to true prior to verifying the correct configuration may render your Oracle Enterprise Repository application unusable).
  - Roles passed within single header
    - Set this value to `True` if more than one role name is passed as a Policy Server response.

- **Role Delimiter**
    - Set the value of this property to the character that will delimit multiple roles within the single roles header. This field can accept Unicode notations such as `\u0020` for a space.
  - **Assign default roles to users**
    - Set this value to `True` if existing and new users are assigned all roles marked as 'default' assigned to their user account within Oracle Enterprise Repository.
  - **Auto create missing roles**
    - Set this value to `True` to allow Oracle Enterprise Repository to create roles included within a user's role header that do not exist currently within Oracle Enterprise Repository. This feature will create a role and assign the user to that role, but the created role(s) will have no permissions assigned.
  - **Auto create missing departments**
    - Set this value to `True` to allow Oracle Enterprise Repository to create departments included within a user's department header that do not exist currently. This feature will create a department and assign the user to that department; the newly created department will not be assign to a project.
11. Enter `cookie login module` in the System Settings Search text box. The Cookie Login Settings section opens in the Enterprise Authentication group of system settings.
  12. Set the **Cookie Login Module property** to `False`.
  13. Enter `plug-in login` in the System Settings Search text box. The Plugin Login Settings section opens in the Enterprise Authentication group of system settings.
  14. Enter `false` in the Plug-in Login Module text box.
  15. Click **Save**.

### 10.2.2.2 Using the Oracle Enterprise Repository SSO Integration with Basic Authentication

If the Single Sign-On installation uses Basic Authentication, additional property settings are required to allow the Oracle Enterprise Repository Asset Editor to function properly.

1. Using the process described above, enable the following property:
  - `cmee.jws.suppress-authorization-header`
2. Set the property to `True`.
3. Click **Save**.

### 10.2.2.3 Modify Application Property Files Manually

**Prerequisite:** Stop the application server. Modifications to properties files may impact any applications running on the application server.

1. Edit the `containerauth.properties` file in `WEB-INF/classes`.

This file contains a list of header names that are specific to the Single Sign-On server. This information represents the Response Headers Single Sign-On uses for replies, and should be acquired from your organization's Single Sign-On Administrators/Architects.

If Single Sign-On responses do not provide the appropriate value for an email header, a blank "" can be substituted instead of a true header value. Other fields that are not supplied or populated by Single Sign-On should be left null.

(An asterisk <\*> indicates a required field.)

- Configure the Header variables that should be mapped to the appropriate Oracle Enterprise Repository user information:

---



---

**Note:** The values indicated below are examples only and must be replaced with the appropriate Single Sign-On Response Header names defined by your Single Sign-On system.

---



---

```

- enterprise.container.auth.username = <UID>
- enterprise.container.auth.firstname = <FIRST_NAME>
- enterprise.container.auth.middlename = <MIDDLE_NAME>
- enterprise.container.auth.lastname = <LAST_NAME>
- enterprise.container.auth.status = <STATUS>
- enterprise.container.auth.email = <MAIL>
- enterprise.container.auth.phone = <PHONE>
- enterprise.container.auth.roles = <ROLES>
- enterprise.container.auth.depts = <DEPARTMENTS>
- enterprise.container.auth.enable-synch-roles = true
- enterprise.container.auth.roles-single-header = true
- enterprise.container.auth.roles-delimiter = \u0020
- enterprise.container.auth.enable-synch-depts = true
- enterprise.container.auth.depts-single-header = true
- enterprise.container.auth.depts-delimiter = \u0020

```

---



---

**Note:** The last six properties listed above are utilized when role and/or department syncing is enabled, and more than one role or department is supplied in a single header. These additional properties can be disabled/ignored depending on the values supplied in the boolean parameters enable-synch-roles and enable-synch-depts. The delimiter field in this example uses the unicode space character; however, unicode is not required for any other delimiter character.

---



---

2. Restart the Oracle Enterprise Repository application.

### 10.2.3 Advanced Options

The following options add functionality for assigning default roles, new user creation/notification, syncing departments, and syncing roles:

- Section 10.2.3.1, "Creating/Assigning Default Roles for New Users"
- Section 10.2.3.2, "Create New Users/Allow Unapproved Users"

- Section 10.2.3.3, "Enable Unapproved/New User Login"
- Section 10.2.3.4, "New User Notification"
- Section 10.2.3.5, "Syncing Departments"
- Section 10.2.3.6, "Syncing Roles"
- Section 10.2.3.7, "Enable Debug Logging"

### 10.2.3.1 Creating/Assigning Default Roles for New Users

#### With Advanced RBAC:

1. Click **Admin** on the Oracle Enterprise Repository menu bar. The Oracle Enterprise Repository Admin screen is displayed.
2. Click **Roles**.
3. Click **Create New**.
4. Enter `Browse_Only` in the name field.
  - Check **Automatically assign to new users**
  - Add any existing users who fit this profile.
5. Click **Save**.
6. Click **role 1: Create/Submit**.
7. Click **Edit**.
  - Uncheck **Automatically assign to new users**.
8. Click **Save**.
9. Click the **User** role.
10. Click **Edit**.
  - Uncheck **Automatically assign to new users**. (User is the default role and automatically assigned to new users as shipped with the Oracle Enterprise Repository.)
11. Click **Save**.
12. Click **Custom Access Settings**.
13. Click **Create New**.
14. Enter `Browse_Only` in the name field.
  - Check **Automatically assign to all new assets**.
  - Locate `Browse_Only` in the list of roles.
  - Check **View**.
15. Click **Save**.
16. Click **OK** to apply to all assets.

#### With Basic Access Settings:

1. Click **Admin** on the Oracle Enterprise Repository menu bar. The Oracle Enterprise Repository Admin screen is displayed.
2. Click **Roles**.

3. Click **Create New**.
4. Enter `Browse_Only` in the name field.
  - Check **Automatically assign to new users**
  - Add any existing users who fit this profile.
5. Click the **User** role.
6. Click **Edit**.
  - Uncheck **Automatically assign to new users**. (User is the default role and automatically assigned to new users as shipped with Oracle Enterprise Repository.)
7. Click **Save**.

### 10.2.3.2 Create New Users/Allow Unapproved Users

The Oracle Enterprise Repository Single Sign-On authentication integration will automatically create new users within the Oracle Enterprise Repository database once they are successfully authenticated. The specific access and permissions granted to new users is determined by the configuration of the default New User Role(s), as described in the previous section. Upon approval by the access administrator, new users may be assigned to other roles with different access settings. However, if the Single Sign-On integration is configured with role synchronization enabled, then the user is assigned the roles provided by Single Sign-On response headers.

### 10.2.3.3 Enable Unapproved/New User Login

When enabled, this option allows unapproved/new Oracle Enterprise Repository users to access the application after Single Sign-On authentication. If disabled, new or unapproved users cannot access Oracle Enterprise Repository. This feature is particularly useful when a manual approval process is required before accessing the application.

- Enable Unapproved User Login = true (file: `enterprise.properties`)
  - `enterprise.security.unapproveduser.allowlogin=true`

### 10.2.3.4 New User Notification

When enabled, this property will notify the access administrator via E-mail when a new user account is added to Oracle Enterprise Repository via Single Sign-On.

- Enable New User Notification = true (file: `cmee.properties`)
  - `cmee.new.unapproved.users.notify=true`

### 10.2.3.5 Syncing Departments

When enabled, this property will synchronize department names from Single Sign-On response header values.

- Enable Department Syncing = true (file: `containerauth.properties`)
  - `enterprise.container.auth.enable-synch-depts` - Set to true if known departments are to be synchronized with users, set to false otherwise.
- Enable Department Creation = true (file: `containerauth.properties`)\*
  - `enterprise.container.auth.auto-create-missing-depts` - Set to true if user's departments are to be automatically created at login, set to false otherwise.

### Notes on Department Synchronization

The Single Sign-On integration will not create new departments. It will only link users to departments that already exist within Oracle Enterprise Repository and have the same name as that provided in the Single Sign-On response header value(s).

The Single Sign-On server may be configured to pass multiple headers of the same name but different values for each department a user is assigned, or one header containing all of the departments that a user is assigned.

- Configuration 1 - A multiple headers of the same name, with a different value in each:

```
enterprise.container.auth.enable-synch-depts= true
enterprise.container.auth.depts-single-header= false
enterprise.container.auth.depts-delimiter= " "
enterprise.container.auth.depts= DEPT_HEADER_NAME
DEPT_HEADER_NAME=DEPTA
DEPT_HEADER_NAME=DEPTB
DEPT_HEADER_NAME=DEPTC
```

and NOT

```
DEPT_HEADER_NAME=DEPTA DEPTB DEPTC ..
```

- Configuration 2 - One header with multiple values separated by a delimiter:

```
enterprise.container.auth.enable-synch-depts= true
enterprise.container.auth.depts-single-header= true
enterprise.container.auth.depts-delimiter= "^"
enterprise.container.auth.depts= DEPT_HEADER_NAME
DEPT_HEADER_NAME=DEPTA^DEPTB^DEPTC^ ...
```

and NOT

```
DEPT_HEADER_NAME=DEPTA
DEPT_HEADER_NAME=DEPTB
DEPT_HEADER_NAME=DEPTC
```

### 10.2.3.6 Syncing Roles

When enabled, this property will synchronize role names from Single Sign-On response header values.

- Enable Role Syncing = true (file: containerauth.properties)
  - enterprise.container.auth.auto-create-missing-roles - Set to true if unknown roles are to be auto-created, set it to false otherwise.

### Notes on Role Synchronization

The Single Sign-On integration can create new roles. The integration will link users to roles that previously exist within the Oracle Enterprise Repository and have the same name as that provided in the Single Sign-On response header value(s). In addition to linking to existing roles, the integration will also create roles found in the header values that do not already exist within the Oracle Enterprise Repository. Roles created in this way will have no rights assigned to them by default.



- Enable Missing Role Creation = true (file: containerauth.properties)
  - enterprise.container.auth.auto-create-missing-roles = true

The Single Sign-On server may be configured to pass one header value for each role a user is assigned.

- Configuration 1 - A multiple headers of the same name, with a different value in each:

```
enterprise.container.auth.enable-synch-roles= true
enterprise.container.auth.roles-single-header= false
enterprise.container.auth.roles-delimiter= " "
enterprise.container.auth.roles= ROLE_HEADER_NAME
ROLE_HEADER_NAME=ROLEA
ROLE_HEADER_NAME=ROLEB
ROLE_HEADER_NAME=ROLEC
and NOT
DEPT_HEADER_NAME=ROLEA ROLEB ROLEC ...
```

- Configuration 2 - One header with multiple values separated by a delimiter:

```
enterprise.container.auth.enable-synch-roles= true
enterprise.container.auth.roles-single-header= true
enterprise.container.auth.roles-delimiter= "^"
enterprise.container.auth.roles= ROLE_HEADER_NAME
DEPT_HEADER_NAME=ROLEA^ROLEB^ROLEC^ ...
and NOT
ROLE_HEADER_NAME=ROLEA
ROLE_HEADER_NAME=ROLEB
ROLE_HEADER_NAME=ROLEC
```

### 10.2.3.7 Enable Debug Logging

Enable debug logging by appending the following line in the log4fl.properties file:

```
log4j.category.com.oer.enterprise.authentication.client.LoginContext=debug,
cmeelog
```

## 10.3 Container Managed Setup

This section describes how to use Container Managed setup to authenticate the users in Oracle Enterprise Repository.

This section contains the following topics:

- Section 10.3.1, "Overview"
- Section 10.3.2, "Configure Oracle Enterprise Repository for Container Managed Authentication"

- Section 10.3.3, "Modify the Web Application's Web.xml File to Allow for Container Authentication"

### 10.3.1 Overview

The container is configured appropriately with a Realm or Authenticator back-end prior to enabling the values within the Oracle Enterprise Repository application.

### 10.3.2 Configure Oracle Enterprise Repository for Container Managed Authentication

You can configure the Oracle Enterprise Repository for container managed authentication with the Access Administrator rights. This procedure is performed on the Oracle Enterprise Repository Admin screen.

1. Enter `container login module` in the System Settings Search text box. The Container Login Module section is displayed in the Enterprise Authentication group of system settings, as shown in Figure 10-4.
2. Modify the following properties as indicated:
  - Container Login Module Class Name
    - Enter `com.oer.enterprise.authentication.server.loginmodule.ContainerLogin` in the text box.
  - Container Login Module Display Name
    - Enter `Container Login Module` in the text box.
  - Container Login Module
    - Set the property to `True`.
  - Enable the Container Managed Authentication Feature
    - Set the `enterprise.authentication.container.enabled` to `True`.
  - Enable Role Synchronization from the User's Security Principle
    - Set the `enterprise.authentication.container.synchroles.enabled` to `True`.

---

---

**Note:** When using Container Managed Authentication with BPM Harvester, the Exchange Utility, or any other REX operation, this property must be set to `false`. With this property set to `false`, user accounts will need to be created manually and have roles assigned to them by someone with at least `accessAdministrator`-level permissions.

---

---

3. Click **Save**.
4. Enter `cookie login module` in the System Settings Search text box. The Cookie Login Settings section opens in the Enterprise Authentication group of system settings.
5. Set the Cookie Login Module property to `False`.
6. Click **Save**.
7. Enter `plug-in login` in the System Settings Search text box. The Plugin Login Settings section opens in the Enterprise Authentication group of system settings.
8. Enter `false` in the Plug-in Login Module text box.

9. Click **Save**.

### 10.3.3 Modify the Web Application's Web.xml File to Allow for Container Authentication

1. Stop the Oracle Enterprise Repository application or the application server that it runs within.
2. Modify the Oracle Enterprise Repository web.xml file:
  - Add the following security constraint contents to the end of the file.

---

**Note:** This configuration will need to be modified to fit your authentication requirements. This example uses BASIC authentication, which may not be appropriate for your environment.

---

```

<!-- Define a security constraint on this application -->
<security-constraint>
  <web-resource-collection>
    <web-resource-name>Entire Application</web-resource-name>
    <url-pattern>/*</url-pattern>
    <http-method>GET</http-method>
    <http-method>PUT</http-method>
    <http-method>POST</http-method>
    <http-method>DELETE</http-method>
  </web-resource-collection>
  <auth-constraint>
    <description>These roles have access to the Oracle Enterprise
Repository</description>
    <role-name>user</role-name>
  </auth-constraint>
</security-constraint>
<security-constraint>
  <web-resource-collection>
    <web-resource-name>Secure Web Service</web-resource-name>
    <url-pattern>/services/OERRegistry</url-pattern>
  </web-resource-collection>
</security-constraint>
<!-- Define the login configuration for this application -->
<login-config>
  <auth-method>BASIC</auth-method>
  <realm-name>Oracle Enterprise Repository</realm-name>
</login-config>
<security-role>
  <role-name>user</role-name>
  <role-name>admin</role-name>
  <role-name>accessAdministrator</role-name>
  <role-name>advancedSubmitter</role-name>
  <role-name>businessAnalyst</role-name>
  <role-name>projectAdministrator</role-name>
  <role-name>projectArchitect</role-name>
  <role-name>registrar</role-name>
  <role-name>registrarAdministrator</role-name>
  <role-name>systemAdministrator</role-name>
</security-role>

```

3. Start / Restart the Oracle Enterprise Repository application.



---

# Configuring Advanced Role-based Access Control

This chapter provides an overview to advanced role-based access control and describes the various concepts in role-based access control.

This chapter contains the following sections:

- [Section 11.1, "Overview"](#)
- [Section 11.2, "Security Considerations"](#)
- [Section 11.3, "Basic Concepts"](#)
- [Section 11.4, "Access Definitions"](#)
- [Section 11.5, "Process Overview"](#)
- [Section 11.6, "Granting and Denying Permissions"](#)
- [Section 11.7, "Configuring Access Settings for Existing Roles"](#)
- [Section 11.8, "Role-based Access Control Use Cases"](#)

## 11.1 Overview

Advanced Role-based Access Control, if enabled, allows organizations to limit access to and visibility of Oracle Enterprise Repository content by role at the asset and file level. This is accomplished by applying custom access settings to assets and/or files in order to limit their accessibility to particular communities of interest.

This foundational capability can be applied to a wide range of organizational initiatives:

- Exposing Web Services to customers and trading partners.
- Limiting the amount of intellectual property that is available to outsourced development teams and managing export control.
- Establishing a Federated Repository that allows everyone to view and access enterprise assets, but limits domain-specific information to users in relevant domains.
- Managing actions available to users, such as submitting, accepting, and registering assets.
- Limiting visibility of assets under development and retired assets.
- Limiting access to source code files to asset production teams.
- Granting browse-only Oracle Enterprise Repository access to selected groups.

## 11.2 Security Considerations

Role-based Access Control (RBAC) allows Oracle Enterprise Repository to track asset usage and production on a by-user basis, and personalizes the presentation of assets, limiting visibility to designated content in Oracle Enterprise Repository. However, Role-based Access Control is not intended to provide security for asset metadata, nor is it intended to supplement the security of underlying asset repositories.

Oracle Enterprise Repository's access settings provide no additional security for otherwise unsecured assets or data. For maximum security, asset metadata should be managed in a manner that prevents direct access to the metadata in an unsecured database. Confidential information should be embedded in the asset downloadable payload, or attached as documentation files and hosted in a secure repository.

---

---

**Caution:** If you choose to circumvent an authentication and permission challenge from the underlying asset repository, you do so at your own risk.

---

---

The following actions also have security implications:

- Configuring SCM systems for access via a single Oracle Enterprise Repository user account
- Allowing system access by unapproved users

Any of these actions has the potential to open the contents of the repository to anyone with network access to Oracle Enterprise Repository.

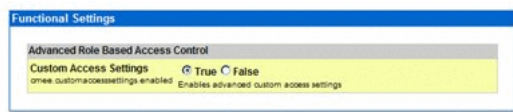
### 11.2.1 Access Settings and the Asset Editor

Depending on the specific Access Settings in use, users with access to the Asset Editor may be able to see relationships to assets that are otherwise invisible to them in Oracle Enterprise Repository. In this situation, the name of the hidden asset is visible, as is the detail of the relationship between that asset and an asset to which the users have access. No other information on the hidden asset is visible in the Asset Editor. In this situation, it is possible to delete the relationship between the visible and the hidden asset. However, given that the invisible asset is inaccessible to the users in question, the relationship cannot be restored.

### 11.2.2 Enabling Custom Access Settings

For Oracle Enterprise Repository users, Custom Access Settings are normally already enabled and configured properly upon installation. If not, three properties must be configured in order to enable Custom Access Settings.

1. Click **System Settings** in the sidebar on the Oracle Enterprise Repository Admin page. The System Settings section opens in the main pane.
2. Enter `cmee.customaccesssettings.enabled` in the Enable New System Setting text box.
3. Click the **Enable** button. Custom Access Settings is displayed in the Advanced Role Based Access Control group in the Functional Settings section, as shown in [Figure 11-1](#).

**Figure 11–1 Functional Settings**

4. Repeat the process to enable each of the following properties:

- `cmee.customaccesssettings.file`
- `cmee.customaccesssettings.asset`

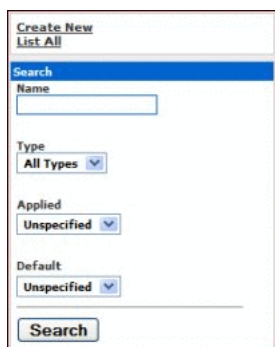
Once enabled, all three CAS system settings will appear in the Advanced Role Based Access Control group in the Functional Settings section. (Disabling any of the properties turns the feature off.)

---

**Note:** If these Access Settings are not displayed in the System Settings: Access list, contact your System Administrator or Oracle Enterprise Repository Customer Support.

---

5. Ensure each of the settings is set to True.
6. When finished, click **Save**. Custom Access Settings now is displayed in the sidebar in the Admin screen, as shown in [Figure 11–2](#).

**Figure 11–2 Custom Access Settings**

## 11.3 Basic Concepts

This section describes the basic concepts of Role-based Access Control. This section describes the following topics:

- [Section 11.3.1, "Roles"](#)
- [Section 11.3.2, "Access Settings"](#)

### 11.3.1 Roles

Within the context of Oracle Enterprise Repository, a role is defined by a specific combination of functions and responsibilities. Any given individual may have multiple roles. Access to specific assets or collections of assets, and to various Oracle Enterprise Repository features, is determined by the configuration of Access Settings for each role.

## 11.3.2 Access Settings

Access settings identify the functions and responsibilities that can be performed by a role. Oracle Enterprise Repository access settings fall into two categories:

- Basic Access Settings (BAS) determine the access to all assets in Oracle Enterprise Repository, and to specific tools, such as the Asset Editor. Basic Access Settings determine each user's rights to general functionality across the system. For example, a user with View permission can view all of the assets in Oracle Enterprise Repository. A user with Use and Download can download all files for all assets in Oracle Enterprise Repository.
- Custom Access Settings (CAS), if enabled, give users permission to access specific assets and files in Oracle Enterprise Repository. For example, Custom Access Settings can give a user the ability to view a specific group of assets, edit specific assets, view specific files within a particular set of assets, and so on.

All sites have Basic Access Settings enabled. Depending on your repository configuration, Custom Access Settings may be enabled.

## 11.4 Access Definitions

This section describes the various definitions used in Oracle Enterprise Repository. This section contains the following topics:

- ["Assets"](#)
- ["Access"](#)
- ["Policies"](#)
- ["Projects"](#)
- ["Reports"](#)
- ["System Administration"](#)

### Assets

Tools regulated: Asset tab, Asset Editor, Type Manager (if enabled in your configuration of the repository).

---



---

**Note:** Users granted the "Launch Asset Editor" permission are able to delete assets. This is a new feature in OER 12c. If you are upgrading from a previous version of OER, it is important to understand that users granted this permission in previous versions will now be able to delete assets.

---



---

Permission	Description	BAS	CAS
View	Access to the Asset Detail.	X	X
Use	Displays the Use/Download button in the Asset Detail.	X	X
Download	Access to file links after clicking an asset's Use/Download button.	X	X
Review	Access to Review function in the Asset Detail.	X	X
Notify	Notify subscribers about assets via ad-hoc email.	X	X
Edit	Access to the Asset Editor.	X	X



Permission	Description	BAS	CAS
Accept	Displays the Accept button in the Asset Editor.	X	X
Approve Tabs	Displays the Approve button at the bottom of each tab in the Asset Editor.	X	X
Register	Ability to register assets via the Asset Editor.	X	X
Edit Access Settings	(CAS only) Provides the user with the ability to change CAS permissions on all assets.	X	X
Create/Submit	Ability to submit assets via the Oracle Enterprise Repository Assets screen, create new assets via the Asset Editor.	X	
Launch Asset Editor	Access to the Asset Editor.	X	
Edit Artifact Stores	Ability to create and edit artifact stores via the Asset Editor.	X	
Edit Asset Types	Ability to configure Type metadata using the Type Manager. Requires Asset Editor permission.	X	

### Access

Features regulated: Admin screen: Users, Sessions, Roles, Departments, File Stores, Basic Access Settings, Custom Access Settings (if enabled in your configuration of the repository).

Permission	Description	BAS	CAS
View	View entries in the sections noted above.	X	
Edit	Edit entries in the sections noted above.	X	
Create	Create users, roles, departments, file stores, and custom access settings.	X	
Delete	Delete sessions, roles, and custom access settings.	X	

### Policies

Tools regulated: Policies screen (if enabled in your configuration of the repository).

Permission	Description	BAS	CAS
Apply Policy	Allows user to apply a policy to other assets.	X	

### Projects

Feature regulated: Projects screen (if enabled in your configuration of the repository).

Permission	Description	BAS	CAS
View	View projects on the Projects screen.	X	
Edit	Edit projects on the Projects screen.	X	
Create	Create projects on the Projects screen.	X	
Apply Template	Apply compliance templates to projects from the Asset Detail or the Asset Editor.	X	

### Reports

Feature regulated: Reports screen (if enabled in your configuration of the repository).

Permission	Description	BAS	CAS
View	Viewing reports.	X	

### System Administration

Features regulated: Admin screen: System Settings, Email Templates.

Permission	Description	BAS	CAS
Edit	Ability to edit general, security, authentication, and file store settings.	X	
Enable	Ability to enable new general, security, authentication, and file store settings.	X	

**Note:** When using BAS or CAS, the following global permissions must be maintained in BAS to grant global feature permissions and functions

- Create/Submit
- Launch Asset Editor
- Edit Artifact Store
- Edit Types

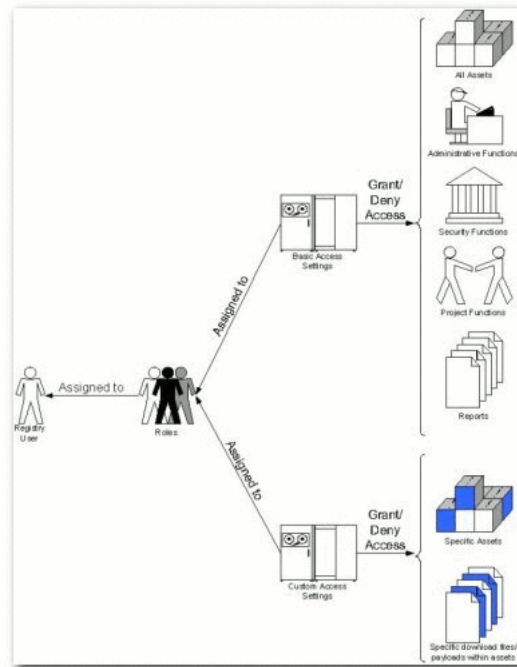
## 11.5 Process Overview

A few simple steps allow your organization to establish a variety of security models:

- Set up roles
- Assign users to roles
- Grant permissions to roles using BAS and CAS
- Assign CAS to assets or files

[Figure 11-3](#) describes the process overview.

Figure 11–3 Process Overview



## 11.6 Granting and Denying Permissions

Individual users typically hold multiple Oracle Enterprise Repository Roles. For example, an individual may have both User and Registrar roles. In this case, the Registrar has permissions that are not granted to User. Similarly, an outsourced development user may have both User and Outsourced Development roles. The User role might grant access to all assets in the repository, while the Outsourced Development role denies access to any assets that have sensitive information.

When determining how to assign specific permissions to a particular Oracle Enterprise Repository user, keep in mind that permission status is divided into three levels:

- **Granted**  
The permission is explicitly granted.
- **Not Granted**  
The permission is not explicitly granted nor denied.
- **Denied**  
The permission is explicitly denied.

In multiple role situations, Oracle Enterprise Repository recognizes the Most Restrictive Access, that is, Denied supersedes Granted in the permission status settings. Any permission specifically Denied to a role to which a user is assigned is also denied to that user in any other role.

This section contains the following topics:

- [Section 11.6.1, "Grant/Deny Access to Specific Assets"](#)
- [Section 11.6.2, "Grant/Deny Access to Specific Download Files Within an Asset"](#)

## 11.6.1 Grant/Deny Access to Specific Assets

If Advanced Role Based Access Control is enabled in your configuration, then you can use Custom Access Settings to grant or deny permission to access individual assets and files. In order to create any Asset CAS, you must be assigned both the Access Administrator role and the following (BAS) permissions:

- Launch Asset Editor
- Edit Access Settings

Access to the Asset Editor and permission to edit assets is required in order to make changes to existing access settings. Oracle Enterprise Repository safeguards eliminate the possibility of locking yourself out of an asset after editing the asset's access settings (it is impossible to save such changes to the asset). However, changes to your role settings may restrict access to certain assets. Exercise caution when editing role settings.

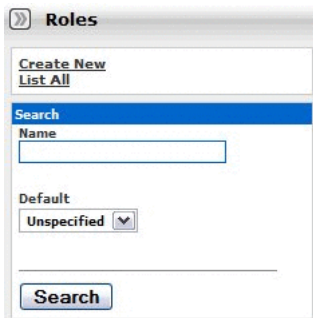
### 11.6.1.1 Prerequisites

- Verify that all targeted users exist in Oracle Enterprise Repository.
- Assign global BAS permissions to each user (i.e. Create/Submit Assets, Launch Asset Editor).

If no existing role encompasses the appropriate users, one must be created. This procedure is performed on the Oracle Enterprise Repository Admin page.

1. Click **Roles** in the left sidebar. The Roles section is displayed, as shown in [Figure 11-4](#).

**Figure 11-4 Roles Section**



The screenshot shows a web interface for managing roles. At the top, there is a header with a double arrow icon and the word 'Roles'. Below this, there are two links: 'Create New' and 'List All'. A search section is highlighted with a blue bar and contains a 'Name' input field. Below the search section, there is a 'Default' dropdown menu with 'Unspecified' selected. At the bottom of the search section, there is a 'Search' button.

2. Click **Create New**. The Create New Role dialog is displayed, as shown in [Figure 11-5](#).

**Figure 11–5 Create New Role Dialog**

3. Enter the appropriate information in the Name and Description fields.
4. Click **Save**.

### 11.6.1.2 Create the Asset CAS

This procedure is performed on the Oracle Enterprise Repository Admin page.

1. Click **Roles** in the left sidebar. A list of available Roles is displayed in the main pane, as shown in [Figure 11–6](#).

**Figure 11–6 List of Roles**

Role Name	Default	Description
accessAdministrator		Access Administrator
advancedSubmitter		Advanced Submitter
businessAnalyst		Business Analyst
projectAdministrator		Project Administrator
projectArchitect		Project Architect
registrar		Registrar
registrarAdministrator		Registrar Administrator
systemAdministrator		System Administrator
user	yes	Normal User

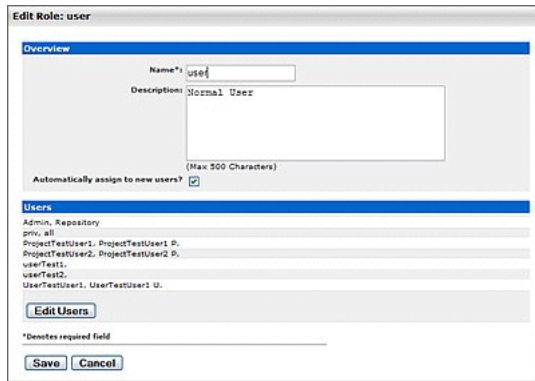
Overview
Name: accessAdministrator
Description: Access Administrator

Users
Admin, Repository
priv. all

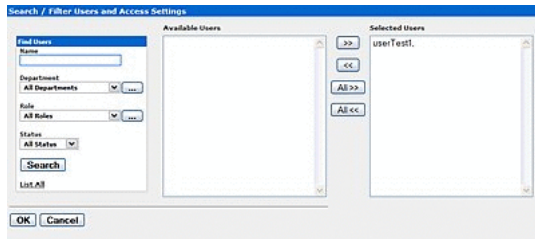
2. From the list, select the role to be edited. The role detail is displayed in the bottom frame of the main pane.
3. Click the **Edit** button in the role detail. The Edit Role dialog is displayed, as shown in [Figure 11–7](#).

**Figure 11-7 Edit Role Dialog**



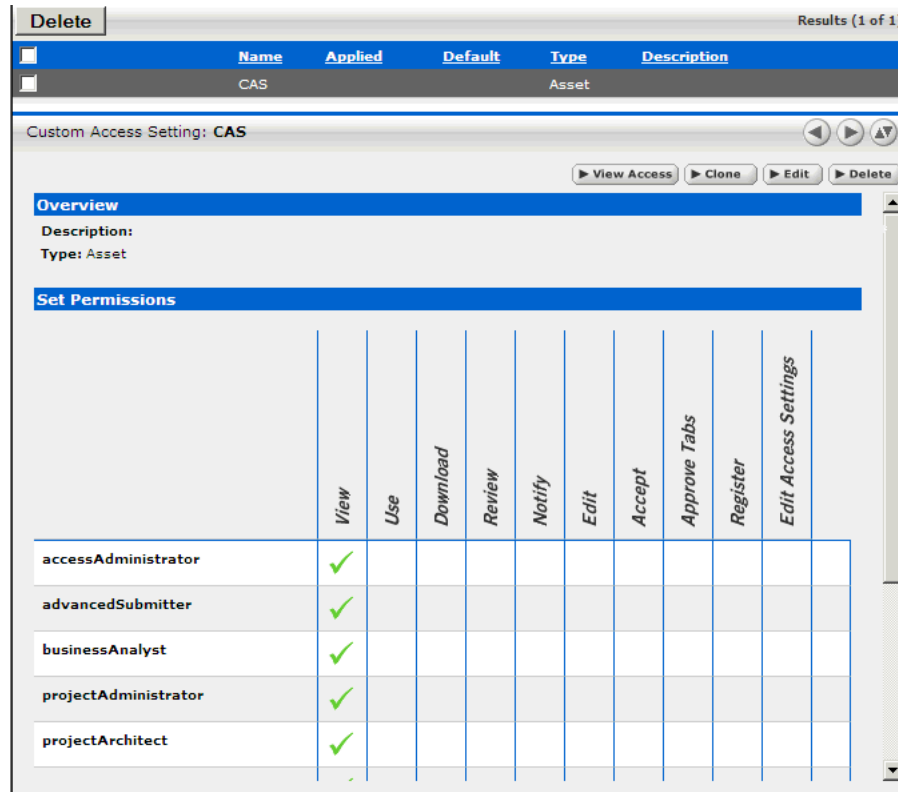
4. Click the **Edit Users** button in the Users section of the Edit Role dialog. The Add Users dialog is displayed, as shown in [Figure 11-8](#).

**Figure 11-8 Search / Filter Users and Access Settings**



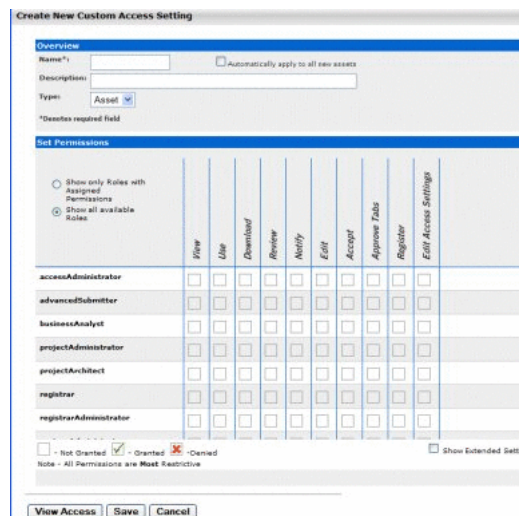
5. Use Search or click **List All** to populate the Available Users column.
6. Use the arrow buttons to move users from the Available Users to the Selected Users column.
7. When finished, click **OK**. The Add Users dialog closes, and the selected users are listed in the Users section of the Edit Role dialog.
8. Click **OK**. The Edit Role dialog closes.
9. Click **Custom Access Settings** in the sidebar on the Admin screen. The Custom Access Settings section is displayed, as shown in [Figure 11-9](#).

**Figure 11–9 Custom Access Settings Section**



10. Click **Create New**. The Create New Custom Access Setting dialog is displayed, as shown in Figure 11–10.

**Figure 11–10 Create New Custom Access Setting Dialog**



11. Enter text as appropriate in the Name and Description text boxes.
12. Select Asset in the Type list.
13. Use the options in the Set Permissions section to assign relevant permissions to the appropriate roles.

- A single click grants permission, as shown in [Figure 11–11](#).

**Figure 11–11 Right Mark**



- A second click denies permission, as shown in [Figure 11–12](#).

**Figure 11–12 Cross Mark**



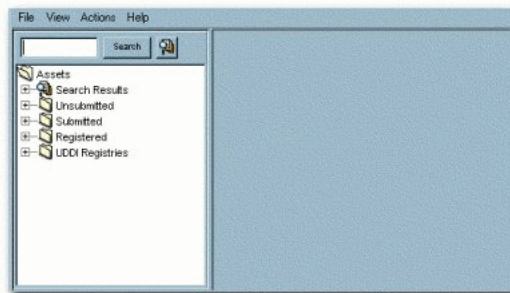
- A third click clears the option (permission Not Granted):

**Figure 11–13 Blank**



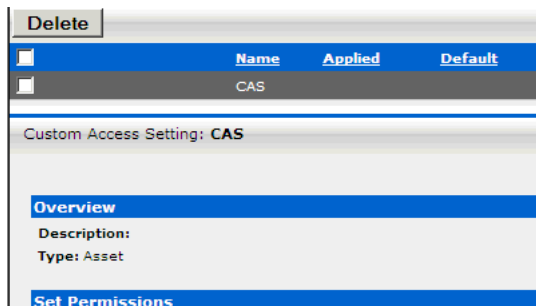
- When finished, click **Save**. The Create New Custom Access Setting dialog is displayed.
- Click **Assets** in the Oracle Enterprise Repository menu bar.
- Click **Edit / Manage Assets**. The Asset Editor is displayed, as shown in [Figure 11–14](#).

**Figure 11–14 Asset Editor Window**



- Locate and open (or create, if necessary) the asset to which the new Custom Access Settings are applied.
- Click the asset's Administration tab, as shown in [Figure 11–15](#).

**Figure 11–15 Administration Tab**





19. Use the arrow buttons as necessary to move the new CAS from the Available to the Selected column.
20. Test the CAS by checking user permissions within and without the use case. Click the **View Access** button to confirm user access.

## 11.6.2 Grant/Deny Access to Specific Download Files Within an Asset

In order to create this or any File CAS you must be assigned the Access Administrator role and the following (BAS) permissions:

- Launch Asset Editor
- Edit Access Settings

### 11.6.2.1 Prerequisites

- Verify that all targeted users exist in Oracle Enterprise Repository.
- Assign global BAS permissions to each user (i.e. Create/Submit Assets, Launch Asset Editor) to allow them to submit and edit assets.

If no existing role encompasses the appropriate users, one must be created. This procedure is performed in the Oracle Enterprise Repository Admin screen.

1. Click **Roles**.
2. Click **Create New**.
3. Supply the appropriate information in the Name and Description fields in the Create New User window.
4. Click **Save**.

### 11.6.2.2 Create the File CAS

This procedure is performed in the Oracle Enterprise Repository Admin screen.

1. Click **Roles**.
2. Using the Edit User button in the Edit Role screen, add the appropriate users to the new role.
3. When finished, click **Save**.
4. Click **Admin** in the Oracle Enterprise Repository's menu bar.
5. Click **Custom Access Settings** in the sidebar on the Admin screen. The Custom Access Settings section is displayed.
6. Click **Create New**. The Create New Custom Access Setting dialog is displayed.
7. Enter text as appropriate in the Name and Description text boxes.
8. Select **File** in the Type list.
9. In the Set Permissions section, select the appropriate role(s) and assign the Download permission for this CAS.
10. Click **Save**.
11. Click **Assets** in the Oracle Enterprise Repository menu bar.
12. Click **Edit / Manage Assets**. The Asset Editor screen is displayed.
13. Locate and open (or create, if necessary) the asset to which the new Custom Access Settings are applied.

14. Locate the asset's File Information metadata element.
15. Select the targeted file.
16. Click the **Edit** button.
17. In the Custom Access Settings section of the Edit window, carefully select **ONLY** the new File CAS. If the targeted file does not exist, add a new file, and (in the Custom Access Settings section) select the new File CAS.
18. Click the asset's Administration tab.
19. Test the CAS by checking user permissions within and without the use case. Click the **View Access** button on the asset to confirm user access.

Several of the most common security use case scenarios are outlined in the sections below. Before implementing one of the new security models, however, it is necessary to configure the existing roles in your system to support Custom Access Settings.

## 11.7 Configuring Access Settings for Existing Roles

Oracle Enterprise Repository ships with several default roles, as defined below. When configuring Oracle Enterprise Repository for Custom Access Settings, these roles must be re-established using a combination of Basic and Custom Access Settings in order to insure that Custom Access Settings do not disrupt existing user access privileges.

### 11.7.1 User Roles and Default Privileges

#### **User**

Anyone with an Oracle Enterprise Repository user name and password. This role can be assigned as the default role for any new users at the time the user account is created. All Oracle Enterprise Repository users can:

- View news on the home page about the host company's initiatives
- Locate, evaluate, and use assets
- View projects (if enabled)
- Generate reports (if enabled)
- Submit assets to the registrar

#### **Access Administrator**

The Access Administrator creates all Oracle Enterprise Repository user accounts and assigns permissions. The Access Administrator must be familiar with the functions available on the Oracle Enterprise Repository Admin screen. Typically, Access Administrators can:

- Create, view, and edit users and permissions
- Generate reports (if enabled)

#### **Advanced Submitter**

The Advanced Submitter role is typically assigned to asset builders and harvesters. Asset builders focus on building the asset content base in response to organizational asset needs and the needs of individual projects. Harvesters study post-implementation projects for asset reuse potential. Typically, advanced submitters can:

- Locate, evaluate, and use assets
- View projects that are associated with assets (if enabled)
- Generate reports (if enabled)
- Submit assets to the registrar
- Edit asset metadata prior to asset registration

### **Registrar**

The Registrar is responsible for asset acceptance or rejection and registration. There may be more than one person functioning as a repository registrar, depending on the functions addressed. Typically, registrars can:

- Locate, evaluate, and use assets
- View projects that are associated with assets
- Generate reports (if enabled)
- Submit assets to the registrar
- Edit asset metadata prior to asset registration
- Accept assets for the registration process
- Approve Asset Editor tabs
- Register assets
- Edit access settings

### **Registrar Administrator**

The Registrar Administrator establishes and manages asset, compliance template, and policy types within Oracle Enterprise Repository using the Type Manager, if enabled in your configuration of the repository. Typically, Registrar Administrators can:

- Locate, evaluate, and use assets
- View projects that are associated with assets (if enabled)
- Generate reports (if enabled)
- Submit assets to the registrar
- Edit asset metadata prior to asset registration
- Accept assets for the registration process
- Approve Asset Editor tabs
- Register assets
- Edit access settings
- Edit Artifact Stores
- Edit Types (if enabled)

### **Project Administrator**

If more than one default project is enabled in your configuration of the repository, then Oracle Enterprise Repository tracks asset use at the project level in order to maintain a history for maintenance purposes. Project Administrators create projects and assign users to projects using the Oracle Enterprise Repository Projects screen (visible only to

users with appropriate permissions). Project Administrators also close projects and indicate which assets were deployed. Typically, Project Administrators can:

- Create, edit, and view projects (if enabled)
- Generate reports (if enabled)

### **System Administrator**

The System Administrator configures Oracle Enterprise Repository for use. The System Administrator typically can:

- Enable and edit system settings
- Generate reports (if enabled)

## **11.7.2 Access Options**

Two options are available for configuring access settings for existing roles, as follows. If Advanced Role Based Access Control is enabled, then access to individual assets and files can be controlled using Custom Access Settings.

- Option 1 - Access to Assets

Grant or deny access to assets. This option is beneficial to organizations that want to use Custom Access Settings to expose a subset of assets to developers within a specific domain, to customers or trading partners, to outsourced developers, etc.

#### **Use Option 1 to:**

- Expose web services to customers and trading partners
- Limit the exposure of intellectual property to outsourced development teams; manage assets that are subject to export controls
- Establish a Federated Repository that allows everyone to view and access enterprise assets, but limits domain specific information to users representing relevant domains
- Manage the asset life cycle by providing limited access to assets under development
- Restrict certain groups to browse-only access to the repository

- Option 2 - File Access

Grant or deny access to download files/payloads within an asset. This option is beneficial to organizations that use Custom Access Settings to support black box reuse. For example, developers may have access to compiled code, while asset producers and maintainers may have access to both source code and compiled code.

#### **Use Option 2 to:**

- Limit access to source code files to asset production and maintenance teams

The setup process is different for each option. It is easiest to start by granting or denying access to assets (Option 1), and adding file access permissions at a later time, if necessary.

---

---

**Note:** Under certain access settings, asset files that are hidden from a particular user's view during the download process are visible and accessible to that same user when using the Asset Editor. In order to totally restrict that user's view of the files it is necessary to also block the user's ability to view the asset in the Asset Editor.

---

---

### 11.7.2.1 Option I: Grant or deny access to specific assets using CAS

#### Step 1 -- Re-establish default roles using Custom Access Settings.

1. Create a CAS for assets called **Basic\_Default\_Assets**.
2. Add the following roles and associated permissions to the CAS:
  - **Role:** User
  - **Permission:** View, Use, Download, Review
  - **Role:** Advanced Submitter
  - **Permission:** View, Use, Download, Review, Edit
  - **Role:** Registrar
  - **Permission:** View, Use, Download, Review, Edit, Accept, Approve Tabs, Register, Edit Access Settings

---

---

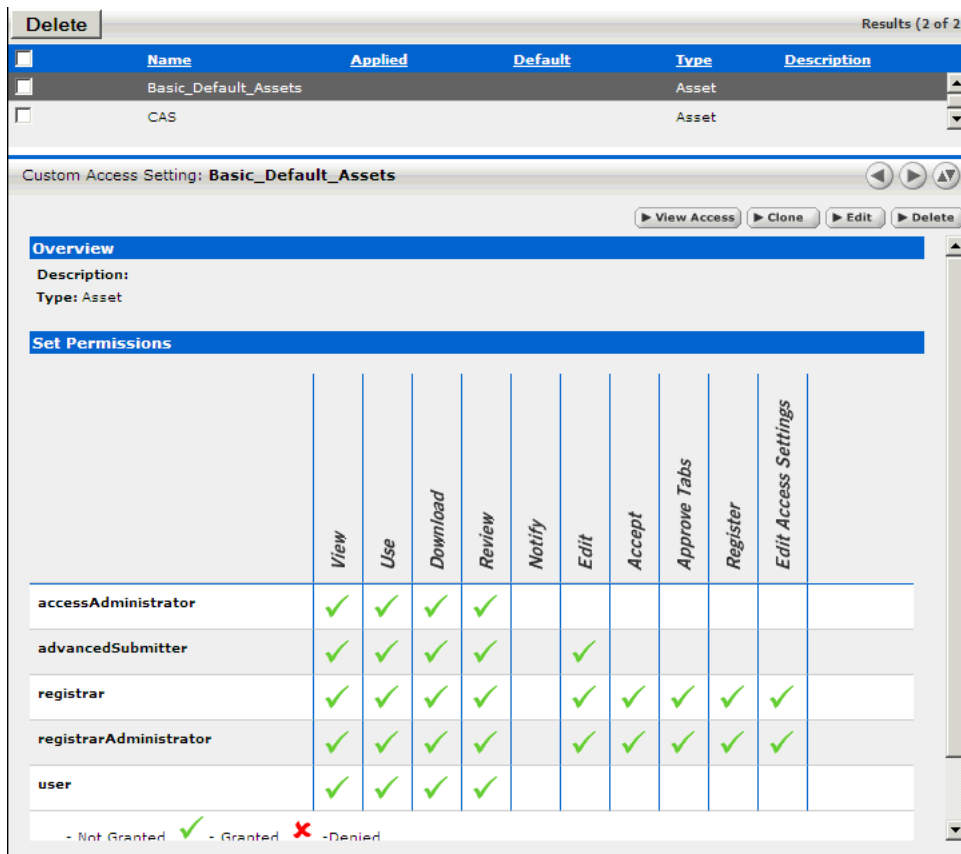
**Note:** Under this configuration, Registrars can view and change Access Settings for all assets. To restrict this privilege, leave the Edit Access Settings option blank for the Registrar role.

---

---

- **Role:** Registrar Administrator
  - **Permission:** View, Use, Download, Review, Edit, Accept, Approve Tabs, Register, Edit Access Settings
3. Select **Automatically apply to all new assets** at the top of the screen.
  4. Click **Save**.
  5. Click **Yes** when the Apply this new setting to all existing assets dialog is displayed. [Figure 11-16](#) illustrates the resulting Custom Access Setting.

**Figure 11–16 Custom Access Setting Screen**



**Step 2 -- Enable access to Oracle Enterprise Repository Tools**

1. Create the following new Roles: (include the numbers preceding the role to keep them together in the role list)
  - 1: Create/Submit
  - 2: Launch Asset Editor
  - 3: Edit Artifact Stores
  - 4: Edit Types
2. Edit current BAS settings to reflect the following:
  - Role: 1: Create/Submit
  - Permission: Create/Submit
  - Role: 2: Launch Asset Editor
  - Permission: Launch Asset Editor
  - Role: 3: Edit Artifact Stores
  - Permission: Edit Artifact Stores
  - Role: 4: Edit Types
  - Permission: Edit Types

Remove existing BAS permissions for assets so that CAS becomes the default permission set for all assets in Oracle Enterprise Repository.

---



---

**Note:** In order to retain global tool permissions through BAS, each user must be assigned to one of the new functional roles for each of the four previously designated functional or tool permissions, as illustrated below.

---



---

### Step 3 -- Tie new roles to existing users based on existing roles

1. Edit each role and place all users who should have the ability to create/submit assets, launch the asset editor, edit artifact stores, and/or edit types, respectively, into each of the four roles.
  - **Assistance Note:** The following table lists the default permissions for Oracle Enterprise Repository's default roles:

Role	Permissions
User	Create/Submit
Advanced Submitter	Create/Submit, Launch Asset Editor
Registrar	Create/Submit, Launch Asset Editor
Registrar Administrator	Create/Submit, Launch Asset Editor, Edit Artifact Stores, Edit Types

#### Example 11-1 Sample Example

Larry was previously assigned the default role User which includes the default permissions. In order to retain his existing permissions under the new settings he must be reassigned to User and also assigned to the new 1: Create/Submit role.

Similarly, in order to retain the default permissions attached to his previously assigned Registrar role, Daryl must be reassigned to Registrar and also assigned to 1: Create/Submit and 2: Launch Asset Editor. It is also possible to add users to a role, as opposed to adding roles to users. If the default permissions for the Oracle Enterprise Repository roles are unchanged, each role can be edited to add all of the applicable users at once.

### Step 4 -- Validate the changes

1. Create five users, and assign each to one of the following role combinations:
  - User, 1: Create/Submit
  - Advanced Submitter, 1: Create/Submit, 2: Launch Asset Editor
  - Registrar, 1: Create/Submit, 2: Launch Asset Editor
  - Registrar Administrator, 1: Create/Submit, 2: Launch Asset Editor, 3: Edit Artifact Stores, 4: Edit Types
  - Project Administrator.
2. Verify that each user assigned to each role listed above can see the following items:
  - User
    - Can see the Assets, Projects, My Stuff, and Reports links on the Oracle Enterprise Repository menu bar.
    - The Submit an Asset link is visible on the Assets screen. (The Edit / Manage Assets link should not be visible.)

- A list of assets is visible after clicking the **Search** button on the Asset screen. The Subscribe and Use/Download buttons should be visible in each asset. (The Edit button should not be visible.) The User should also be able to post reviews.
- After clicking the **Use/Download** button on an asset that includes downloadable files, the User should be able to download the files.
- **Advanced Submitter**
  - Can see the Assets, My Stuff, and Reports links on the Oracle Enterprise Repository menu bar.
  - Both the Submit an Asset and Edit / Manage Assets links are visible on the Assets screen.
  - A list of assets is visible after clicking the **Search** button on the Assets screen. The Subscribe, Use/Download, and Edit buttons should be visible in each asset. The Advanced Submitter should also be able to post reviews.
  - After clicking the **Use/Download** button on an asset that includes downloadable files, the Advanced Submitter should be able to download the files.
  - Can click the **Edit / Manage Assets** link on the Assets screen to launch the Asset Editor.

Verify:

- The Approve button at the bottom of each Asset Editor tab is inactive.
- The Register button on the Administration tab in the Asset Editor is inactive.
- **Registrar**
  - Can see the Assets, My Stuff, and Reports links on the Oracle Enterprise Repository menu bar.
  - Both the Submit an Asset and Edit / Manage Assets links are visible on the Assets screen.
  - A list of assets is visible after clicking the **Search** button on the Assets screen. The Subscribe, Use/Download, and Edit buttons should be visible in each asset. The Registrar should also be able to post reviews.
  - After clicking the **Use/Download** button on an asset that includes downloadable files, the Registrar should be able to download the files.
  - Can click the **Edit / Manage Assets** link on the Assets screen to launch the Asset Editor.

Verify:

- The Approve button at the bottom of each Asset Editor tab is inactive.
- The Register button on the Administration tab in the Asset Editor is inactive.
- In the Asset Editor, the Accept button is visible on assets submitted via the Submit an Asset link on the Oracle Enterprise Repository Assets screen.
- The Registrar can change an asset's access setting (if enabled in CAS).
- **Registrar Administrator**



- Can see the Assets, My Stuff, and Reports links on the Oracle Enterprise Repository menu bar.
- Both the Submit an Asset and Edit / Manage Assets links are visible on the Assets screen.
- A list of assets is visible after clicking the **Search** button on the Assets screen. The Subscribe, Use/Download, and Edit buttons should be visible in each asset. The Registrar Administrator should also be able to post reviews.
- After clicking the **Use/Download** button on an asset that includes downloadable files, the Registrar should be able to download the files.
- Can click the **Edit / Manage Assets** link on the Assets screen to launch the Asset Editor.

Verify:

- The Approve button at the bottom of each Asset Editor tab is inactive.
- The Register button on the Administration tab in the Asset Editor is inactive.
- In the Asset Editor, the Accept button is visible on assets submitted via the Submit an Asset link on the Oracle Enterprise Repository Assets screen.
- The Registrar Administrator can change an asset's access setting (if enabled in CAS).
- The Actions menu of the Asset Editor should include Configure Artifact Stores and Manage Asset Types.

- Project Administrator

- Can see the Assets, My Stuff, and Reports links on the Oracle Enterprise Repository menu bar.

Verify:

- The Submit an Asset and Edit / Manage Assets links are NOT visible on the Assets screen.
- No search results appear after clicking the **Search** button.

### 11.7.2.2 Option II: Grant or deny access to specific files and assets using CAS

Create custom access settings that determine user access to assets, as well as to specific files within an asset.

#### Step 1 -- Enable all three CAS settings under System Settings

You must enable all the three CAS settings under the System Settings pane.

#### Step 2 -- Allow all roles that currently access the download files for all assets to get the same functionality through a File CAS

1. Create a CAS for files called Basic\_Default\_Files.
2. Add the following roles, each with download privileges:
  - User
  - Advanced Submitter
  - Registrar

- Registrar Administrator
3. Select **Automatically apply to all new files**.
  4. Click **Yes** when the Apply this new setting to all existing files? dialog is displayed.

**Step 3 -- Allow all roles that currently access all assets to get the same functionality through an Asset CAS**

1. Create a CAS for assets called Basic\_Default\_Assets. Add the following roles and associated permissions:
  - **Role:** User
  - **Permissions:** View, Use, Review
  - **Role:** Advanced Submitter
  - **Permissions:** View, Use, Review, Edit
  - **Role:** Registrar
  - **Permissions:** View, Use, Review, Edit, Accept, Approve Tabs, Register, Edit Access Settings
  - **Role:** Registrar Administrator
  - **Permissions:** View, Use, Review, Edit, Accept, Approve Tabs, Register, Edit Access Settings
2. Select **Automatically apply to all new assets** at the top of the screen.
3. Click **Yes** when the Apply this new setting to all existing assets? dialog is displayed.

**Step 4 -- Remove existing BAS permissions for assets**

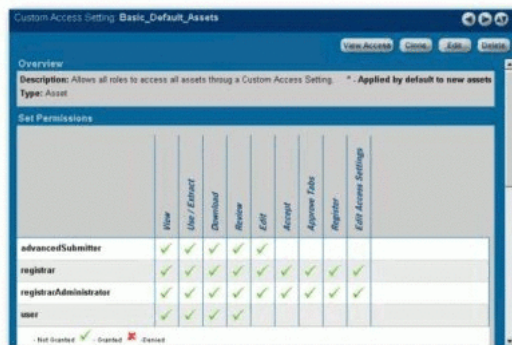
CAS becomes the default set of permissions for all assets in Oracle Enterprise Repository.

---

**Note:** In order to retain global tool permissions through BAS, each user must be assigned to one of the new functional roles for each of the four previously designated functional or tool permissions, as illustrated in [Figure 11-17](#).

---

**Figure 11-17 Custom Access Setting**



1. Create four new Roles with the following names: (include the numbers preceding the role to keep them together in the role list)

- 1: Create/Submit
  - 2: Launch Asset Editor
  - 3: Edit Artifact Stores
  - 4: Edit Types
2. Edit current BAS settings to reflect the following:
- Role: 1: Create/Submit
  - Permission: Create/Submit
  - Role: 2: Launch Asset Editor
  - Permission: Launch Asset Editor
  - Role: 3: Edit Artifact Stores
  - Permission: Edit Artifact Stores
  - Role: 4: Edit Types
  - Permission: Edit Types

### Step 5 -- Tie new roles to existing users based on existing roles

1. Edit each role and place all users who should have the ability to create/submit assets, launch the asset editor, edit artifact stores, and/or Edit Types, respectively, into each of the four roles.

The following table lists the default permissions for Oracle Enterprise Repository's default roles:

Role	Permissions
User	Create/Submit
Advanced Submitter	Create/Submit, Launch Asset Editor
Registrar	Create/Submit, Launch Asset Editor
Registrar Administrator	Create/Submit, Launch Asset Editor, Edit Artifact Stores, Edit Types

#### **Example 11–2 Sample Example**

Daryl was previously assigned the default role User, which includes the default permissions. In order to retain his existing permissions under the new settings he must be reassigned to User and also assigned to the new 1: Create/Submit role.

Similarly, in order to retain the default permissions attached to his previously assigned Registrar role, Larry must be reassigned to Registrar and also assigned to 1: Create/Submit and 2: Launch Asset Editor.

It is also possible to add users to a role, as opposed to adding roles to users. If the default permissions for the Oracle Enterprise Repository roles are unchanged, each role can be edited to add all of the applicable users at once.

### Step 6 -- Validate the changes

1. Create five users, and assign each to one of the following role combinations:
  - User, 1: Create/Submit
  - Advanced Submitter, 1: Create/Submit, 2: Launch Asset Editor

- Registrar, 1: Create/Submit, 2: Launch Asset Editor
  - Registrar Administrator, 1: Create/Submit, 2: Launch Asset Editor, 3: Edit Artifact Stores, 4: Edit Types
  - Project Administrator
2. Verify that each user assigned to each role listed above can see the following items:
- User
    - Can see the Assets, Projects, My Stuff, and Reports links on the Oracle Enterprise Repository menu bar.
    - The Submit an Asset link is visible on the Assets screen. (The Edit / Manage Assets link should not be visible.)
    - A list of assets is visible after clicking the Search button on the Asset screen. The Subscribe and Use/Download buttons should be visible in each asset. (The Edit button should not be visible.) The User should also be able to post reviews.
    - After clicking the Use/Download button on an asset that includes downloadable files, the User should be able to download the files.
  - Advanced Submitter
    - Can see the Assets, My Stuff, and Reports links on the Oracle Enterprise Repository menu bar.
    - Both the Submit an Asset and Edit / Manage Assets links are visible on the Assets screen.
    - A list of assets is visible after clicking the Search button on the Assets screen. The Subscribe, Use/Download, and Edit buttons should be visible in each asset. The Advanced Submitter should also be able to post reviews.
    - After clicking the Use/Download button on an asset that includes downloadable files, the Advanced Submitter should be able to download the files.
    - Can click the Edit / Manage Assets link on the Assets screen to launch the Asset Editor.

Verify

- The Approve button at the bottom of each Asset Editor tab is inactive.
- The Register button on the Administration tab in the Asset Editor is inactive.
- Registrar
  - Can see the Assets, My Stuff, and Reports links on the Oracle Enterprise Repository menu bar.
  - Both the Submit an Asset and Edit / Manage Assets links are visible on the Assets screen.
  - A list of assets is visible after clicking the Search button on the Assets screen. The Subscribe, Use/Download, and Edit buttons should be visible in each asset. The Registrar should also be able to post reviews.
  - After clicking the Use/Download button on an asset that includes downloadable files, the Registrar should be able to download the files.

- Can click the Edit / Manage Assets link on the Assets screen to launch the Asset Editor.

#### Verify

- The Approve button at the bottom of each Asset Editor tab is active.
- The Register button on the Administration tab in the Asset Editor is active.
- In the Asset Editor, the Accept button is visible on assets submitted via the Submit an Asset link on the Oracle Enterprise Repository Assets screen.
- The Registrar can change an asset's access setting (if enabled in CAS).

- Registrar Administrator

- Can see the Assets, My Stuff, and Reports links on the Oracle Enterprise Repository menu bar.
- Both the Submit an Asset and Edit / Manage Assets links are visible on the Assets screen.
- A list of assets is visible after clicking the Search button on the Assets screen. The Subscribe, Use/Download, and Edit buttons should be visible in each asset. The Registrar Administrator should also be able to post reviews.
- After clicking the Use/Download button on an asset that includes downloadable files, the Registrar should be able to download the files.
- Can click the Edit / Manage Assets link on the Assets screen to launch the Asset Editor.

#### Verify

- The Approve button at the bottom of each Asset Editor tab is active.
- The Register button on the Administration tab in the Asset Editor is active.
- In the Asset Editor, the Accept button is visible on assets submitted via the Submit an Asset link on the Oracle Enterprise Repository Assets screen.
- The Registrar Administrator can change an asset's access setting.
- The Actions menu of the Asset Editor should include Configure Artifact Stores and Manage Asset Types.

- Project Administrator

- Can see the Assets, Projects, My Stuff, and Reports links on the Oracle Enterprise Repository menu bar.

#### Verify

- The Submit an Asset and Edit / Manage Assets links are NOT visible on the Assets screen.
- No search results appear after clicking the Search button.

## 11.8 Role-based Access Control Use Cases

This section describes the role-based access control use cases.

This section contains the following topics:

- [Section 11.8.1, "Use Case 1: Expose Web Services to Customers and Trading Partners"](#)

- [Section 11.8.2, "Use Case 2: Manage Intellectual Property in an Outsourced Environment"](#)
- [Section 11.8.3, "Use Case 3: Establish a Federated Repository"](#)
- [Section 11.8.4, "Use Case 4: Manage the Asset Lifecycle"](#)
- [Section 11.8.5, "Use Case 5: Limit Access to Source Code Files to Asset Production Teams"](#)
- [Section 11.8.6, "Use Case 6: Grant Browse-only Repository Access to Specific Groups"](#)

## 11.8.1 Use Case 1: Expose Web Services to Customers and Trading Partners

This section describes a use case to demonstrate how to expose Web services to customers and trading partners. This section contains the following topics:

- [Section 11.8.1.1, "Benefit"](#)
- [Section 11.8.1.2, "Overview"](#)
- [Section 11.8.1.3, "Solution"](#)

### 11.8.1.1 Benefit

This scenario allows an organization to leverage Web Services to improve business operations between trading partners and allow its customers to connect to systems and gather information, but restricts access to information that might otherwise expose sensitive knowledge of backend systems.

### 11.8.1.2 Overview

In this scenario, the organization's trading partners and customers have access only to the appropriate Web Services within Oracle Enterprise Repository. Access to other assets is denied. In order to facilitate these measures, the metadata for Web service assets is organized into two general groupings:

- **Unrestricted**  
Metadata, associated files, and downloadable files that the organization wants to expose to all internal and external parties.
- **Restricted**  
Metadata, design documents, or files that reveal the inner workings of the organization's backend systems. Only internal developers will have access to this information.

The Custom Access Settings (CAS) described in this scenario provide all internal developers with access to both the restricted and unrestricted parts of Web services assets, as well as the other assets in the repository. The organization's trading partners and customers have access only to the unrestricted Web Service assets. This scenario necessitates the creation of two different Web Services asset types -- one for unrestricted data and another one for restricted data.

---

---

**Note:** This scenario describes a simple way to expose Web Services to customers and trading partners. However, other considerations not addressed by this use case may require additional or alternative measures. For example, it may be more appropriate to establish a separate instance of Oracle Enterprise Repository for external use only. The external instance may be branded differently than the internal instance, and would include only the metadata, associated files, and downloadable files suitable for exposure to external parties. In this case, usage histories and reviews would reflect only the experience of external users. If your situation requires a more complex solution, please contact your Oracle Enterprise Repository Implementation Manager or Solutions Consultant for assistance and guidance.

---

---

### 11.8.1.3 Solution

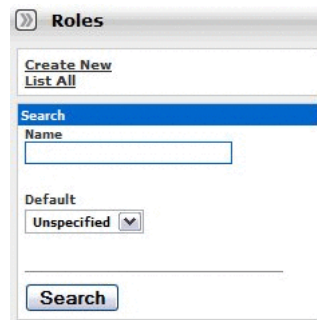
This section describes the solution to this use case.

**11.8.1.3.1 Prerequisite** Verify that Customer and Trading Partner users exist in Oracle Enterprise Repository.

**11.8.1.3.2 Create and Assign the Roles** This procedure is performed on the Oracle Enterprise Repository Admin screen.

1. Click **Roles**. The Roles section is displayed, as shown in [Figure 11–18](#).

**Figure 11–18 Roles Section**



2. Click **Create New**. The Create New Role dialog is displayed, as shown in [Figure 11–19](#).

**Figure 11–19 Create New Role Dialog**

3. Enter `User - Customer` in the Name text box.
4. Enter `Represents Web Services customers` or some other descriptive text in the Description text box.
5. When finished, click **Save**.
6. Repeat the above process to create the role `User - Trading Partner`.
7. Assign the new roles to the appropriate users (i.e. user accounts representing the appropriate customers and trading partners).
  - On the Admin screen:
    - Click **Roles** to assign user to roles, or...
    - Click **Users** to assign roles to users.

**11.8.1.3.3 Organize the Metadata** The metadata for Web Service assets must be organized into two general groupings according to the roles that will have Oracle Enterprise Repository access to the information. The first grouping consists of the metadata that can be viewed by everyone (Customers, Trading Partners, and Internal Developers). The second grouping consists of metadata that can be viewed only by Internal Developers.

1. Launch the Asset Editor.
2. Click **Manage Types** in the Actions menu. The Type Manager is displayed.
3. Use the **Service** asset type as a template to create a new asset type called `Service - Internal Only`.
4. When finished, click **Save** in the file menu in the Type Manager.
5. Click **Copy/Migrate** in the File menu in the Asset Editor.
6. Copy/migrate all existing Web Service assets to the new `Service - Internal Only` asset type.
7. Edit all original assets of the `Service` asset type to delete all metadata fields that reflect or represent information or metadata that is to be restricted from public view.

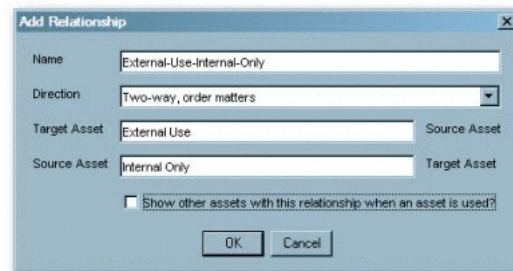


8. Delete all non-restricted fields (the first metadata grouping, as mentioned above) from the new Service - Internal Only asset type.

**11.8.1.3.4 Create the Relationships** This procedure is performed in the Asset Editor.

1. Click **Configure Relationships** in the Actions menu. The Configure Relationships dialog is displayed.
2. Click **Add**.
3. Configure a two-way relationship called External-Use-Internal-Only, as shown in [Figure 11-20](#).

**Figure 11-20 Add Relationship Dialog**

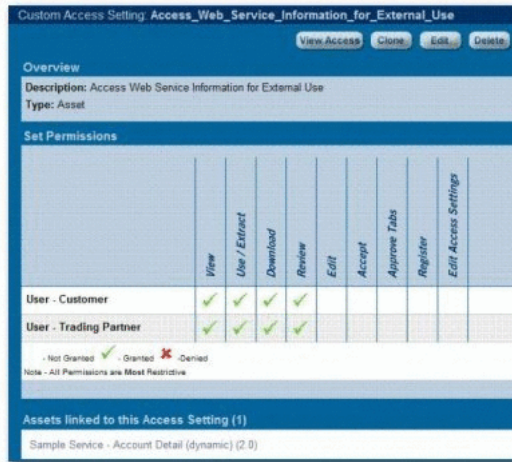


**11.8.1.3.5 Apply the Relationship** Add the External-Use-Internal-Only relationship to the Service assets.

The procedure is performed in the Asset Editor.

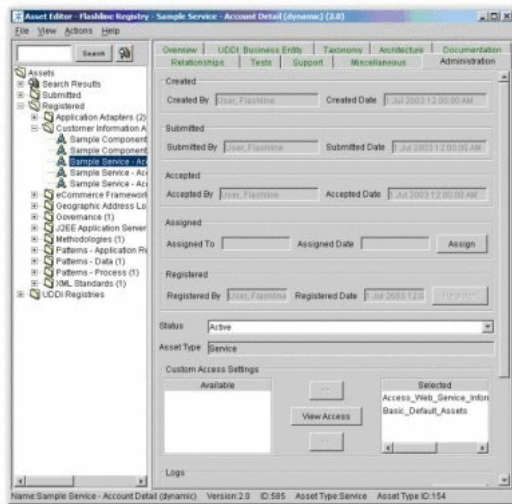
1. Open one of the Service assets.
2. Click the **Relationship** tab.
3. Apply the External-Use-Internal-Only relationship to each Service asset. This will allow Internal Users to access the restricted fields.
4. Create a CAS.
  - Click **Custom Access Settings** in the sidebar on the Admin screen.
  - Click **Create New**.
  - Name the new CAS `Access_Web_Service_Information_for_External_Use`, as shown in [Figure 11-21](#).

**Figure 11–21 Custom Access Setting: CAS Access\_Web\_Service\_Information\_for\_External\_Use**



- Set the following permissions:
    - **Role:** User - Customer
    - **Permissions:** View, Use, Download, Review
    - **Role:** User - Trading User
    - **Permissions:** View, Use, Download, Review
5. Launch the Asset Editor.
  6. Open one of the Web Service assets.
  7. Select the **Administration** tab.
  8. Apply the new `Access_Web_Service_Information_for_External_Use` CAS to the Web Service. This CAS will work in tandem with the `Basic_Default_Assets` CAS already assigned to each asset, as shown in [Figure 11–22](#).

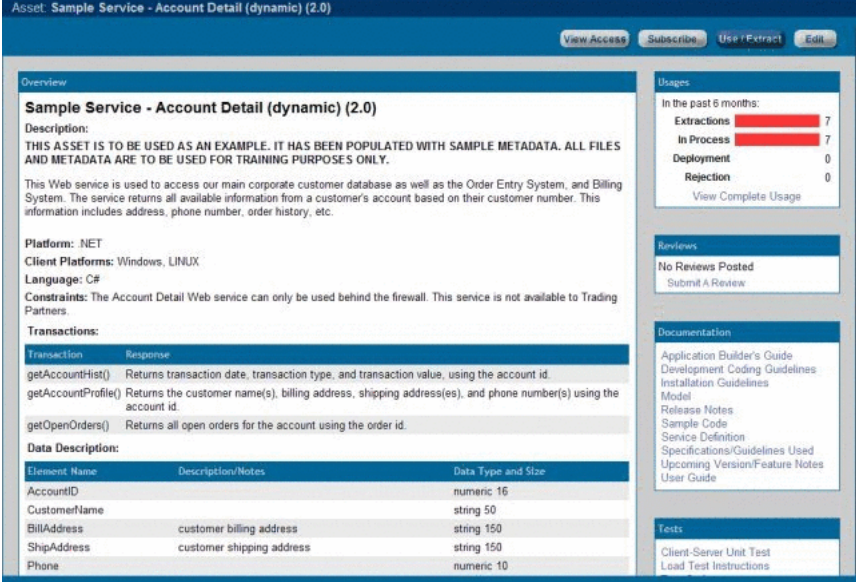
**Figure 11–22 Asset Editor Screen**



9. Repeat the process with each of the Web Service assets.

### 11.8.1.3.6 Results ■ Service Asset Type

Figure 11–23 Service Asset Type



Asset: Sample Service - Account Detail (dynamic) (2.0)

View Access | Subscribe | Use / Extract | Edit

**Overview**

**Sample Service - Account Detail (dynamic) (2.0)**

Description:  
THIS ASSET IS TO BE USED AS AN EXAMPLE. IT HAS BEEN POPULATED WITH SAMPLE METADATA. ALL FILES AND METADATA ARE TO BE USED FOR TRAINING PURPOSES ONLY.

This Web service is used to access our main corporate customer database as well as the Order Entry System, and Billing System. The service returns all available information from a customer's account based on their customer number. This information includes address, phone number, order history, etc.

Platform: NET  
Client Platforms: Windows, LINUX  
Language: C#  
Constraints: The Account Detail Web service can only be used behind the firewall. This service is not available to Trading Partners.

Transactions:

Transaction	Response
getAccountHist()	Returns transaction date, transaction type, and transaction value, using the account id.
getAccountProfile()	Returns the customer name(s), billing address, shipping address(es), and phone number(s) using the account id.
getOpenOrders()	Returns all open orders for the account using the order id.

Data Description:

Element Name	Description/Notes	Data Type and Size
AccountID		numeric 16
CustomerName		string 50
BillAddress	customer billing address	string 150
ShipAddress	customer shipping address	string 150
Phone		numeric 10

Usages

In the past 6 months:

Extractions	7
In Process	7
Deployment	0
Rejection	0

View Complete Usage

Reviews

No Reviews Posted  
Submit A Review

Documentation

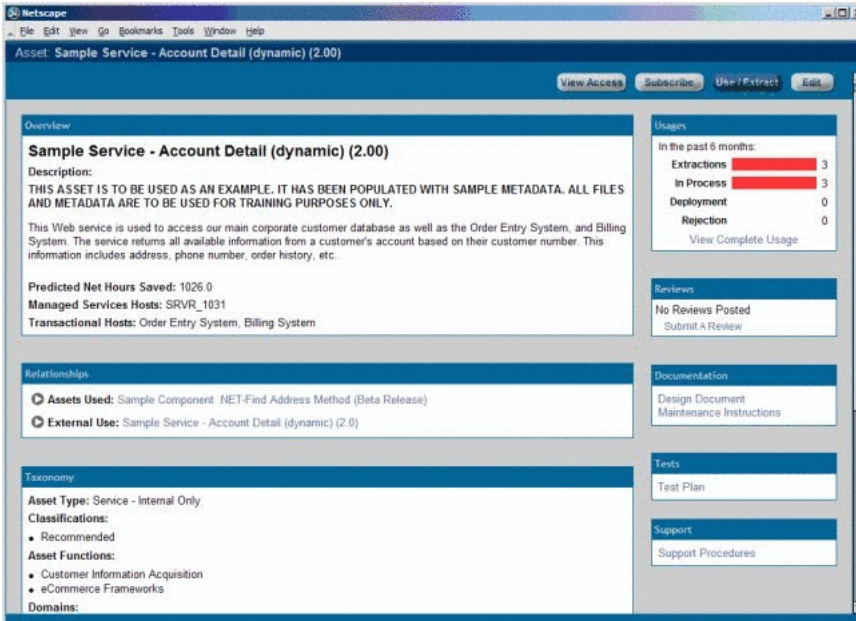
Application Builder's Guide  
Development Coding Guidelines  
Installation Guidelines  
Model  
Release Notes  
Sample Code  
Service Definition  
Specifications/Guidelines Used  
Upcoming Version/Feature Notes  
User Guide

Tests

Client-Server Unit Test  
Load Test Instructions

### ■ Service - Internal Only Asset Type

Figure 11–24 Service - Internal Only Asset Type



Netscape

Asset: Sample Service - Account Detail (dynamic) (2.00)

View Access | Subscribe | Use / Extract | Edit

**Overview**

**Sample Service - Account Detail (dynamic) (2.00)**

Description:  
THIS ASSET IS TO BE USED AS AN EXAMPLE. IT HAS BEEN POPULATED WITH SAMPLE METADATA. ALL FILES AND METADATA ARE TO BE USED FOR TRAINING PURPOSES ONLY.

This Web service is used to access our main corporate customer database as well as the Order Entry System, and Billing System. The service returns all available information from a customer's account based on their customer number. This information includes address, phone number, order history, etc.

Predicted Net Hours Saved: 1026.0  
Managed Services Hosts: SRVR\_1031  
Transactional Hosts: Order Entry System, Billing System

Relationships

Assets Used: Sample Component, NET-Find Address Method (Beta Release)  
External Use: Sample Service - Account Detail (dynamic) (2.0)

Taxonomy

Asset Type: Service - Internal Only  
Classifications:  
• Recommended  
Asset Functions:  
• Customer Information Acquisition  
• eCommerce Frameworks  
Domains:

Usages

In the past 6 months:

Extractions	3
In Process	3
Deployment	0
Rejection	0

View Complete Usage

Reviews

No Reviews Posted  
Submit A Review

Documentation

Design Document  
Maintenance Instructions

Tests

Test Plan

Support

Support Procedures

**Note:** Note that considerably less information is provided in this view.

**11.8.1.3.7 Validation Test** Select a user in each of the three designated roles and verify access as described below:

- **Roles**
  - User - Customer
  - User - Trading Partner
- **Access**
  - Only the Assets and My Stuff links are displayed in the Oracle Enterprise Repository menu bar.
  - Can see only the Web Service asset metadata intended for external view (Service Asset Type).
  - Relationships for these assets are not visible.
  - When using Web Service assets, only the primary downloadable files are available for download. No related assets are available.
  - On the Oracle Enterprise Repository Assets screen, the Submit Assets and Edit/Manage Assets buttons are not displayed.
  - The Edit button is not displayed when viewing the asset detail screen for Web Service assets.
- **Role**
  - User (internal employees):
- **Access**
  - Can see Web Service metadata intended for internal and external view (Service assets and Service - Internal Only assets).
  - Can see all relationships on Service assets.
  - If the Use/Download button is clicked when viewing a Service -Internal Only asset, files intended for external view are displayed as related assets in the download window. Both internal and external files are available for simultaneous download.

## 11.8.2 Use Case 2: Manage Intellectual Property in an Outsourced Environment

This section describes a use case that demonstrates how to manage intellectual property in an outsourced environment. This section contains the following topics:

- [Section 11.8.2.1, "Benefits"](#)
- [Section 11.8.2.2, "Overview"](#)
- [Section 11.8.2.3, "Solution"](#)

### 11.8.2.1 Benefits

The settings described in this scenario allow an organization that is operating in a highly distributed outsourced environment to manage and protect intellectual property by limiting the exposure of intellectual property to outsourced development teams. These settings also ensure that assets adhere to the limitations specified under export control regulations.

## 11.8.2.2 Overview

### Outsourced Development Teams

In this scenario, External Service Providers (ESPs) can view, use, and provide reviews ONLY for the assets specified in their Project Profile. During the course of the project, ESPs can also submit assets produced within the project to a technical lead or Project Architect for review, approval, and asset registration. For example, after the design phase of the project, an ESP can submit the design documents for approval by the technical lead or project architect. This scenario presents a project lifecycle governance model and facilitates project milestone reviews.

### Export Controls

Export control regulations typically prohibit the export of software and other types of intellectual property to certain countries. In this scenario, assets that are subject to export control are assigned a Custom Access Setting indicating that they are restricted under a specific Export Control Classification Number (ECCN). Restricted individuals are assigned a corresponding Export Control Classification Number (ECCN) role, in addition to the roles granted by virtue of their regular responsibilities as employees. For example, encryption software may not be exported to Iraq, Iran, Libya, North Korea, Sudan, Syria, or Cuba under ECCN 5D002. All encryption software assets in Oracle Enterprise Repository will receive a custom access setting `Access_Restricted_by_ECCN_5D002`, identifying that they are subject to export control. Individuals from Iraq, Iran, Libya, North Korea, Sudan, Syria, or Cuba will also be assigned the role `User - Restricted by ECCN 5D002`, indicating that they are restricted from accessing encryption software. As export control limitations change, it is easy to identify the individuals and assets affected in order to modify access settings.

## 11.8.2.3 Solution

This section describes the solution to this use case. This section contains the following topics:

- [Section 11.8.2.3.1, "Solution for Outsourced Development Teams"](#)
- [Section 11.8.2.3.2, "Solution for Export Controls"](#)

**11.8.2.3.1 Solution for Outsourced Development Teams** The following procedures are performed on the Oracle Enterprise Repository Admin screen.

### Prerequisite

Verify that External Service Providers have Oracle Enterprise Repository user accounts.

### Create the Roles

1. Click **Roles**.
2. Click **Create New**.
3. Create a new role named `User - ESP Project X`.

This role represents External Service Providers working on a particular project. Users in this role can view and download ONLY the assets specified in their Project Profile. They can also submit their assets to a technical lead for review, approval, and asset registration during the course of a project.

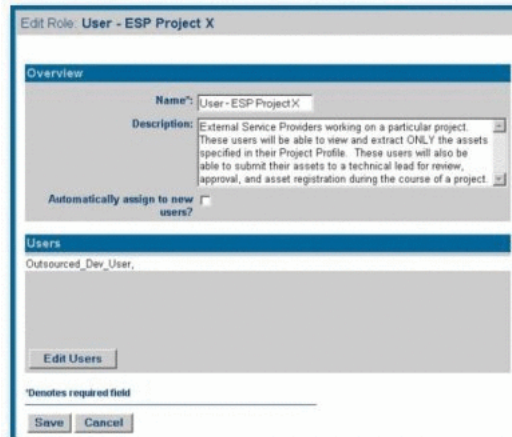
### Assign the Roles

Assign the User - ESP Project X role and the 1: Create/Submit roles to the appropriate users.

The 1: Create/Submit role allows ESPs to submit assets they produce on the project to a technical lead or Project Architect for review, approval, and asset registration during the course of the project.

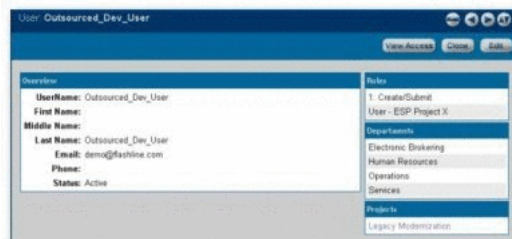
Click **Roles** to assign users to roles, as shown in [Figure 11–25](#).

**Figure 11–25 Edit Role Dialog**



Click **Users** to assign roles to users, as shown in [Figure 11–26](#).

**Figure 11–26 Users**

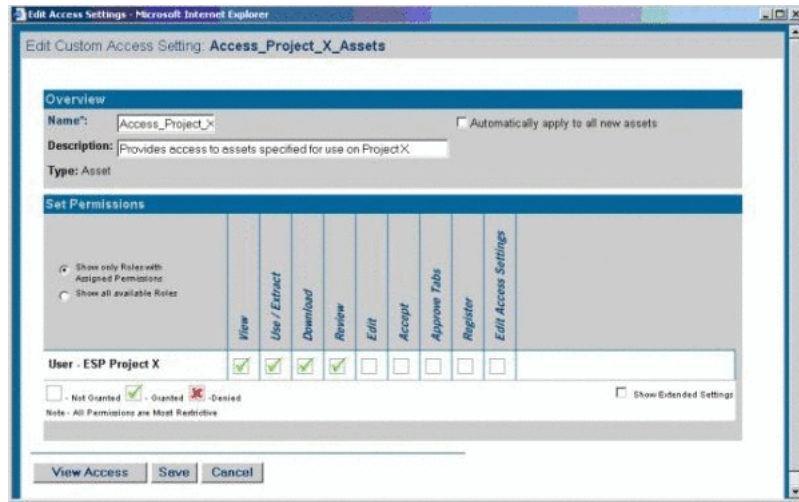


### Create the Custom Access Setting

1. Click **Custom Access Settings**.
2. Click **Create New**.
3. Create a new CAS named `Access_Project_X_Assets`. This CAS provides access to assets specified for use on Project X.
4. Set the following permissions, as shown in [Figure 11–27](#).
  - **Role:** User - ESP Project X
  - **Permissions:** View, Use, Download, Review

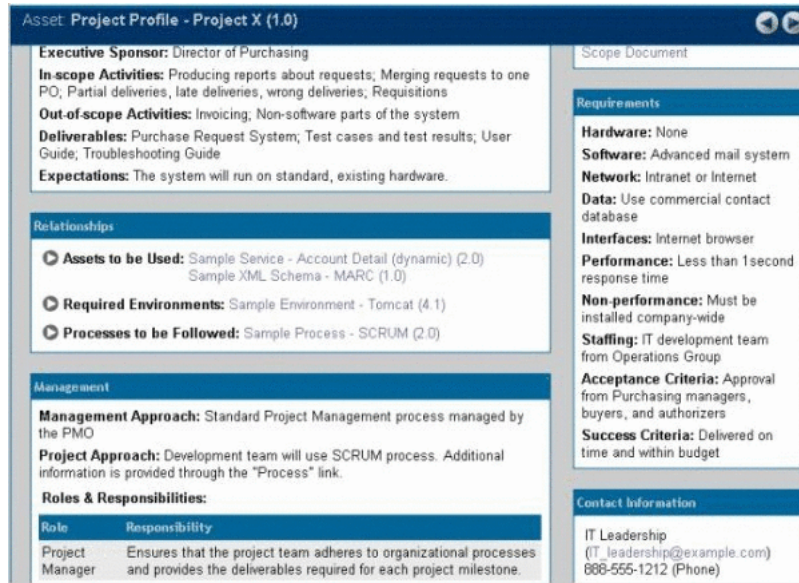


**Figure 11–27 Edit Custom Access Settings Dialog**



5. Identify the assets specified for use on Project X through the Project Profile, as shown in Figure 11–28.

**Figure 11–28 Asset Project Profile - Project X (1.0)**



6. Add Access\_Project\_X\_Assets CAS to the Project Profile for Project X, and to each asset specified in the Project Profile for Project X, as shown in Figure 11–29

**Figure 11–29 Custom Access Settings**



## Validation Test

Verify that the following conditions for External Service Providers assigned to Project X:

- **Role**
  - Outsourced\_Dev\_User:
- **Access**
  - Can see the Project Profile for Project X.
  - Can see only those assets that have been specified in the Project Profile for Project X.
  - Can download the assets specified in the Project Profile for Project X.
  - The Submit Assets link is visible on the Assets screen.
  - On the asset detail display, the Subscribe and Use/Download buttons are visible; the Edit button is hidden.
  - Only the Assets and My Stuff tabs are visible in Oracle Enterprise Repository.

**11.8.2.3.2 Solution for Export Controls** The following procedures are performed on the Oracle Enterprise Repository Admin screen.

### Prerequisites

Verify that individuals restricted under export controls have Oracle Enterprise Repository user accounts.

### Create the Role

1. Click **Roles**.
2. Click **Create New**.
3. Create a role named User - Restricted by ECCN 5D002. This role is assigned to individuals from Iraq, Iran, Libya, North Korea, Sudan, Syria, or Cuba -- any user who is to be restricted from accessing encryption software or software limited for export under ECCN 5D002.
4. Assign the new User - Restricted by ECCN 5D002 role to the appropriate users, as shown in [Figure 11-30](#).



**Figure 11–30 Edit Role**
**Create the Custom Access Setting**

1. Click **Custom Access Settings**.
2. Click **Create New**.
3. Create a new CAS named `Access_Restricted_by_ECCN_5D002`. This CAS restricts access to the assets specified as limited for export under ECCN 5D002.
4. Set the following permissions, as shown in [Figure 11–31](#).
  - **Role:** User - Restricted by ECCN 5D002
  - **Permissions:** DENY access to: View, Use, Download, Review, Accept, Approve Tabs, Register, Edit Access Settings

**Figure 11–31 Edit Custom Access Setting Dialog**

	View	Use / Extract	Download	Review	Edit	Accept	Approve Tabs	Register	Edit Access Settings
User - Restricted by ECCN 5D002	Denied	Denied	Denied	Denied	Denied	Denied	Denied	Denied	Denied

5. Add `Access_Restricted_by_ECCN_5D002` CAS to all encryption software assets or assets restricted by export control under ECCN 5D002.

**Figure 11–32 Custom Access Settings****Validation Test**

Verify that the individuals restricted by export control limitations:

- Cannot see the assets restricted by export control limitations.
- Can see all assets not restricted under export control limitations.

**11.8.3 Use Case 3: Establish a Federated Repository**

This section describes the use case that demonstrates how to establish a federated repository. This section contains the following topics:

- [Section 11.8.3.1, "Benefit"](#)
- [Section 11.8.3.2, "Overview"](#)
- [Section 11.8.3.3, "Solution"](#)

**11.8.3.1 Benefit**

The settings described in this scenario allow Oracle Enterprise Repository users to view and access enterprise assets, but limit access to domain specific information to those assigned to the relevant domains. This allows a large organization to target specific asset consumer groups in order to provide them with the most relevant assets for their needs.

**11.8.3.2 Overview**

This scenario involves enterprise-wide assets that may be viewed and downloaded by anyone within the organization, and domain-specific assets that are relevant only to development teams within specific business domains. Producers of enterprise-wide assets can create, edit, and register their assets, and view and download all domain assets. Producers of domain-specific assets can create, edit, and register assets within their domain. All users within the organization may view and download enterprise-wide assets, in addition to assets that are specific to their domain.

**11.8.3.3 Solution**

This section describes the solution to this use case.

**11.8.3.3.1 Prerequisite** Verify that enterprise and domain producers and all consumers have Oracle Enterprise Repository user accounts.

**11.8.3.3.2 Create the Role**

1. Click the **Admin** link in the Oracle Enterprise Repository menu bar.
2. On the Admin screen, click **Roles**
3. Click **Create New**.
4. Create the following roles:
  - **Role:** Enterprise Producer

- Access
  - Can view, download, edit, and register enterprise assets.
  - Can view and download Domain assets.
- **Role:** Domain X Producer
  - Access
    - Can view, download, edit, and register Domain X assets.
    - Can view and download Enterprise assets.
- **Role:** Domain X Consumer
  - Access
    - Can view and download Domain X assets and Enterprise assets.
- **Role:** Domain Y Consumer
  - Access
    - Can view, download, edit, and register Domain Y assets
    - Can view and download Enterprise assets.

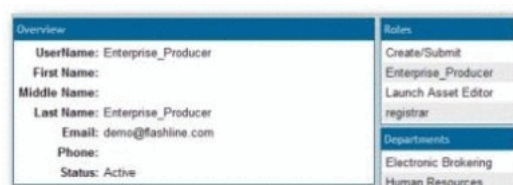
#### 11.8.3.3.3 Create the Custom Access Settings

1. Click the **Admin** link in the Oracle Enterprise Repository menu bar.
2. On the Admin screen, click **Custom Access Settings**.
3. Click **Create New**.
4. Create the following Custom Access Settings:
  - **CAS:** Access\_Enterprise\_Assets
    - Allows any Oracle Enterprise Repository user to view enterprise-wide assets.
  - **CAS:** Access\_Domain\_X\_Assets
    - Restricts access to Domain X assets to those assigned to the Enterprise Producer and/or Domain X Producer roles.
  - **CAS:** Access\_Domain\_Y\_Assets
    - Restricts access to Domain Y assets to those assigned to the Enterprise Producer and/or Domain Y Producer roles.

For each Domain or Enterprise asset, remove the Basic Default Assets access setting and replace it with the appropriate Custom Access Setting, as noted in Step 4, above.

- Roles assigned to Enterprise\_Producer

**Figure 11–33 Overview Section**



- Corresponding CAS

**Figure 11–34 Set Permissions**

	View	Use / Extract	Download	Review	Edit	Accept	Approve Tabs	Register	Edit Access Settings
Domain X Consumer	✓	✓	✓						
Domain X Producer	✓	✓	✓						
Domain Y Producer	✓	✓	✓						
Enterprise_Producer	✓	✓	✓	✓	✓	✓	✓	✓	✓

- Role assigned to Domain\_X\_Producers

**Figure 11–35 Overview Section**

Overview	Roles
UserName: Domain_X_Producer First Name: Middle Name: Last Name: Domain_X_Producer Email: demo@flashline.com Phone: Status: Active	Create/Submit Domain X Producer Launch Asset Editor registrar
	Departments
	Electronic Brokering Human Resources

- Role assigned to Domain\_X\_Consumers

**Figure 11–36 Overview Section**

Overview	Roles
UserName: Domain_X_Consumer First Name: Middle Name: Last Name: Domain_X_Consumer Email: demo@flashline.com Phone: Status: Active	Create/Submit Domain X Consumer user
	Departments
	Electronic Brokering Human Resources Operations

- Corresponding CAS

**Figure 11–37 Set Permissions**

	View	Use / Extract	Download	Review	Edit	Accept	Approve Tabs	Register	Edit Access Settings
Domain X Consumer	✓	✓	✓						
Domain X Producer	✓	✓	✓	✓	✓	✓	✓	✓	✓
Enterprise_Producer	✓	✓	✓						

- Roles assigned to Domain\_Y\_Producers

**Figure 11–38 Overview Sections**

Overview	Roles
UserName: Domain_Y_Producer	Create/Submit
First Name:	Domain Y Producer
Middle Name:	Launch Asset Editor
Last Name: Domain_Y_Producer	registrar
Email: demo@fashline.com	Departments
Phone:	Electronic Brokering
Status: Active	Human Resources

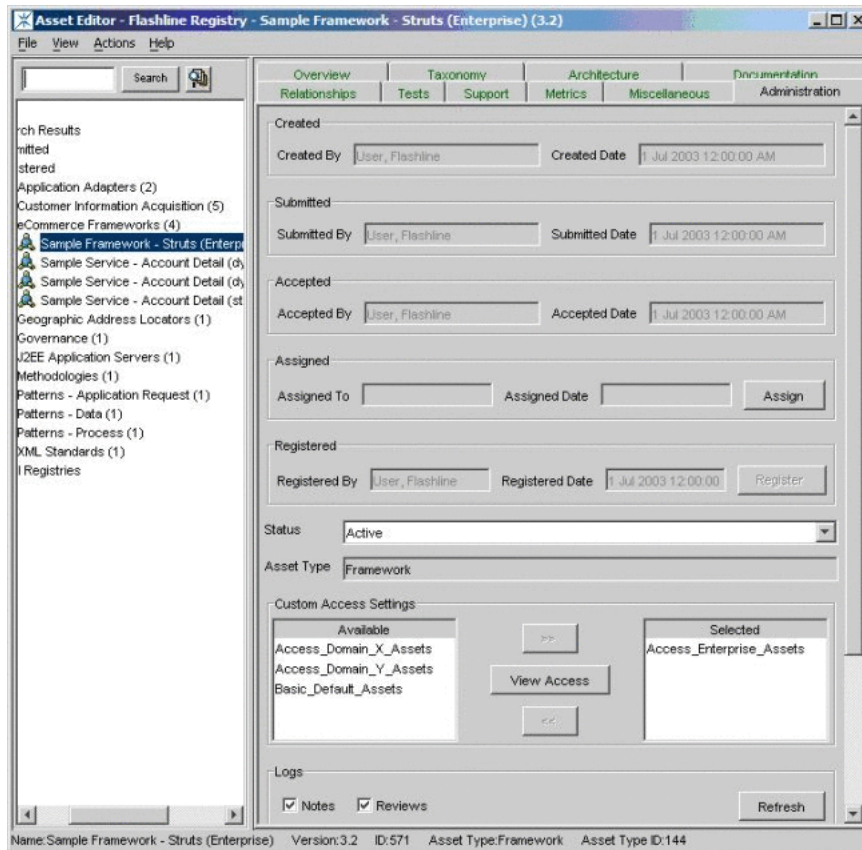
- Corresponding CAS

**Figure 11–39 Edit Access Settings**

	View	Use / Extract	Download	Review	Edit	Accept	Approve Tabs	Register	Edit Access Settings
Domain Y Producer	✓	✓	✓	✓	✓	✓	✓	✓	✓
Enterprise_Producer	✓	✓	✓						

5. For each Enterprise or Domain asset, remove the Basic\_Default\_Assets setting and replace it with the appropriate Access\_Enterprise\_Assets, Access\_Domain\_X\_Assets, or Access\_Domain\_Y\_Assets CAS, as shown in [Figure 11–40](#).



**Figure 11–40 Asset Editor**

#### 11.8.3.3.4 Validation Test Verify the following conditions for each user/role:

- Enterprise Producer:
  - Metadata for individual Enterprise, Domain X, and Domain Y assets is visible on Oracle Enterprise Repository's Assets screen.
  - Only Enterprise assets are visible in the Asset Editor (no Asset Editor access to Domain X or Domain Y assets).
  - The Subscribe, Use/Download, and Edit buttons are visible when viewing the metadata for Enterprise assets.
  - Can edit and register Enterprise assets in the Asset Editor.
  - The Use/Download buttons are visible in the metadata display for both Domain X and Domain Y assets. However, the Edit button is NOT visible.
- Domain X Producer:
  - Metadata for individual Enterprise and Domain X assets is visible on Oracle Enterprise Repository's Assets screen; Domain Y assets are not visible.
  - Domain X assets are visible in the Asset Editor (no Asset Editor access to Domain Y or Enterprise assets).
  - The Subscribe and Use/Download buttons are available when viewing the metadata for Enterprise assets. However, the Edit button is not available.
  - Can download files from Enterprise assets.

- The Subscribe, Use/Download, and Edit buttons are visible in the metadata display for Domain X assets.
- Can edit and register Domain X assets in the Asset Editor.
- Domain X Consumer:
  - Metadata for individual Enterprise and Domain X assets is visible on Oracle Enterprise Repository Assets screen. Domain Y assets are not visible.
  - The Subscribe and Use/Download buttons are available when viewing the metadata for Enterprise and Domain X assets. However, in both cases the Edit button is not available.
  - Can download files from Enterprise and Domain X assets.
- Domain Y Producer:
  - Metadata for individual Enterprise and Domain Y assets is visible on Oracle Enterprise Repository's Assets screen; Domain X assets are not visible.
  - Domain Y assets are visible in the Asset Editor. However, there is no Asset Editor access to Domain X and Enterprise assets.
  - The Subscribe and Use/Download buttons are available when viewing the metadata for Enterprise and Domain Y assets. However, in both cases the Edit button is not available.
  - Can download files from Enterprise and Domain Y assets.

#### 11.8.4 Use Case 4: Manage the Asset Lifecycle

This section describes the use case that demonstrates how to manage the asset life cycle in Oracle Enterprise Repository. This section contains the following topics:

- [Section 11.8.4.1, "Benefit"](#)
- [Section 11.8.4.2, "Overview"](#)
- [Section 11.8.4.3, "Configuring the Use Case Solutions"](#)
- [Section 11.8.4.4, "Asset Lifecycle Management Solution Validation Tests"](#)

##### 11.8.4.1 Benefit

The settings described in this scenario allow an organization to manage assets throughout their lifecycle, from initial conception through retirement. These settings provide limited access to assets under development (assets in progress) and to retired assets. This helps to eliminate redundant development efforts by exposing assets still in development to all development teams. Development teams in need of such an asset can plan to include it in their projects, and can collaborate with the asset production team during asset development. Other settings limit the distribution of retired assets, which helps to maintain control over deployed assets.

## 11.8.4.2 Overview

---

---

**Note:** This use case takes place against a backdrop of policies and procedures for Asset Release Management, which is focused on managing the lifecycle of assets and the artifacts that make up the assets. As part of Asset Release Management, the artifacts that make up an asset are produced by projects whose end result can be anything from the asset itself, to an entire product or system. These artifacts are harvested, along with supporting artifacts and other information, and the aggregate is packaged as an asset. Once verified as complete and correct these assets enter the repository where they are available to the entire organization. As registered assets are downloaded from the repository, defects are identified and enhancements are requested. New versions of the asset are created to address these change requests. Eventually, either through disuse or replacement by another version, an asset is retired. Retired assets are unavailable for use.

---

---

The Oracle Enterprise Repository can be used to manage access to an asset throughout its entire lifecycle, from inception through retirement. This use case focuses primarily on using Advanced RBAC to expose and to provide access to assets to different stakeholder groups throughout the asset lifecycle.

### 11.8.4.2.1 Asset Lifecycle Stages ■ Requirements Gathering

The organization has identified the need by several upcoming projects for a particular asset. The Production Team creates a proposed asset in progress. The metadata for this asset includes a brief description of the asset and its purpose, and is open for review and comment by a set of Subject Matter Experts associated with the relevant projects. The repository display of the asset includes a link to a dedicated virtual forum, which is used to gather functional requirements from members of the upcoming projects. The asset is visible only to those producing the asset as well as to the Subject Matter Experts on the relevant projects -- at least until the proposed asset is approved and moves to the next stage in the lifecycle, when it is visible to a wider audience.

---

---

**Note:** Requires the Assets in Progress option for Oracle Enterprise Repository.

---

---

#### ■ Design and Development

Once the requirements are gathered and funding for the development of the asset is approved, the producers begin generating the supporting artifacts (such as design documents and code) and associating them with the asset in the repository. At this point, only the asset's producers can add artifacts and metadata to the asset and download the asset's code. However, the asset is visible to a broader community of developers in order to allow interested parties to track its progress.

#### ■ Beta Release

After the code has been tested it is made available as a beta release to the Subject Matter Experts representing the upcoming projects that expect to use the asset.

#### ■ Release

Once the asset has been thoroughly tested and documented, it is made available to the entire user community.



- **Scheduled for Retirement**

Once the asset has outlived its usefulness, it is scheduled for retirement. Its repository status is changed to inactive. Inactive assets remain in the repository, but access to inactive assets is limited to metadata - the asset cannot be used, and none of its files are available for download. These restrictions provide an incentive for projects to migrate to the latest version of the asset. Those responsible for managing the asset retain access to view and edit the asset until it is retired.

- **Retired**

Once all projects have migrated to the new version of the asset, the repository status of the original asset is changed to retired. The asset is visible only to repository users (regardless of role) if it is related to another asset in the repository. Users may view the asset metadata through the relationship link.

This scenario describes only one of several strategies for managing access to assets in progress and retired assets. For more information about these strategies, contact Oracle Support.

**11.8.4.2.2 General Configuration** The various Asset Lifecycle Management solutions described in this section require the configuration of the following roles and access settings.

**(Prerequisite:** Verify that the asset production teams and subject matter experts are users within the Oracle Enterprise Repository)

**11.8.4.2.3 Create the Roles**

1. Click the **Admin** link in the Oracle Enterprise Repository menu bar.
2. On the Admin screen, click **Roles**.
3. Click **Create New**.
4. Create the following roles:
  - User - Production Team Project X
  - User - Subject Matter Experts Project X

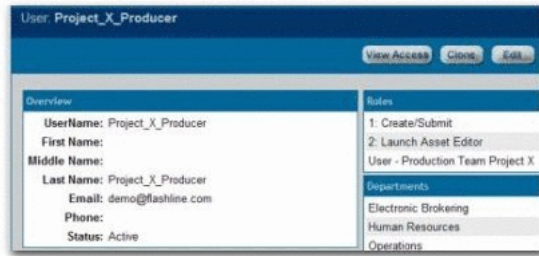
**11.8.4.2.4 Create the Custom Access Settings**

1. Click the **Admin** link in the Oracle Enterprise Repository menu bar.
2. On the Admin screen, click **Custom Access Settings**.
3. Click **Create New**.
4. Create the following Custom Access Settings:
  - CAS: Access\_Project\_X\_Assets\_Propose
  - CAS: Access\_Project\_X\_Assets\_Plan
  - CAS: Access\_Project\_X\_Assets\_Build
  - CAS: Access\_Project\_X\_Assets\_Release

The images below illustrate the detail for each role, along with the corresponding Custom Access Settings.

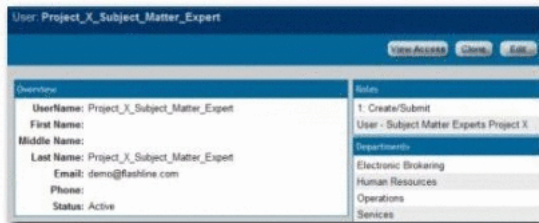
- Project\_X\_Producer: Roles

**Figure 11-41 User: Project\_X\_Producer**



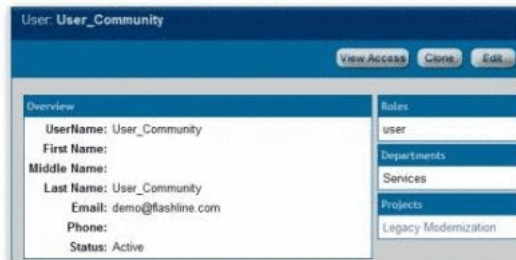
- Project\_X\_Subject\_Matter\_Expert: Roles

**Figure 11-42 User: Project\_X\_Subject\_Matter\_Expert**



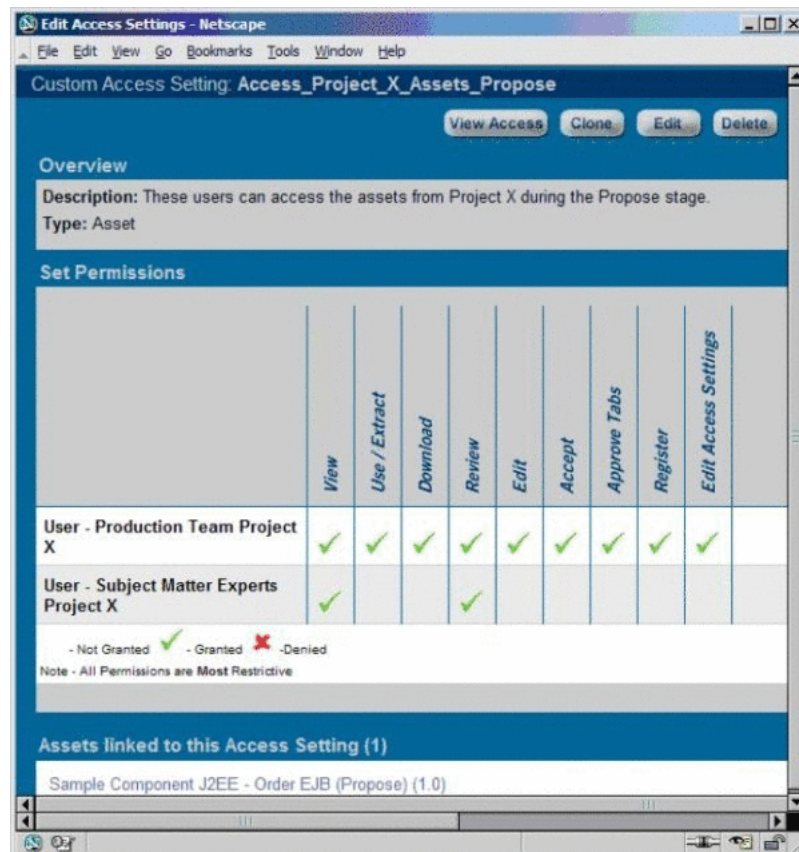
- User\_Community: Roles

**Figure 11-43 User: User\_Community**



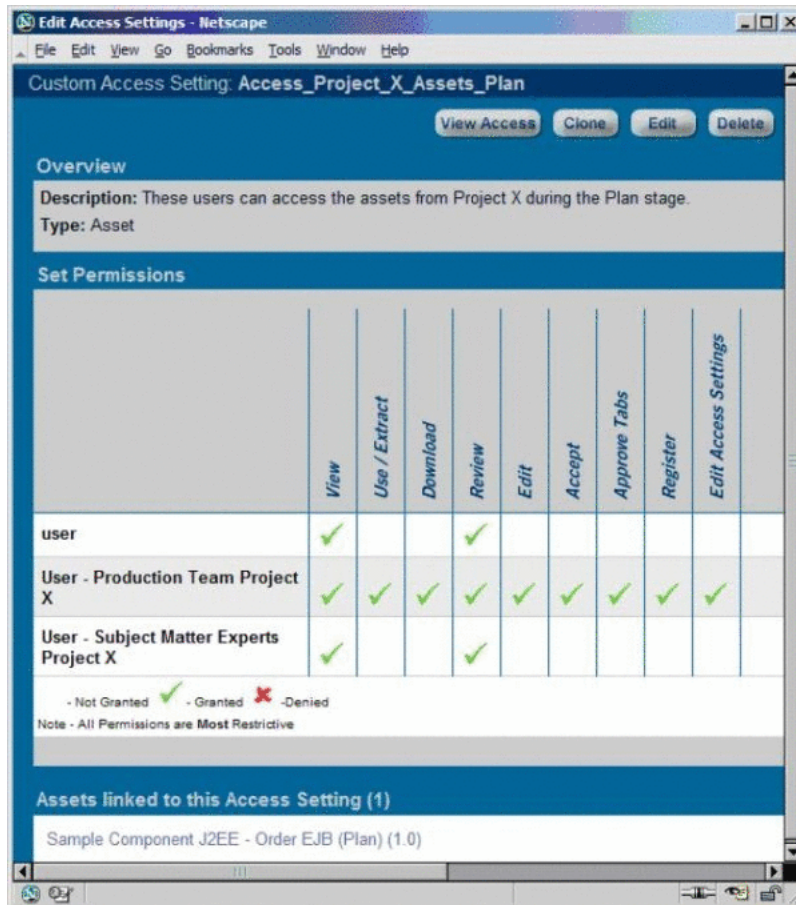
- Asset Lifecycle Stage - Propose
  - CAS: Access\_Project\_X\_Assets\_Propose

Figure 11-44 Edit Access Settings



- Asset Lifecycle Stage - Plan
  - CAS: Access\_Project\_X\_Assets\_Plan

Figure 11-45 Edit Access Settings



- Asset Lifecycle Stage - Build
  - CAS: Access\_Project\_X\_Assets\_Build

Figure 11-46 Edit Access Settings

Custom Access Setting: Access\_Project\_X\_Assets\_Build

View Access Clone Edit Delete

**Overview**

**Description:** These users can access the assets from Project X during the Build stage.  
**Type:** Asset

**Set Permissions**

	View	Use / Extract	Download	Review	Edit	Accept	Approve Tabs	Register	Edit Access Settings
user	✓			✓					
User - Production Team Project X	✓	✓	✓	✓	✓	✓	✓	✓	✓
User - Subject Matter Experts Project X	✓	✓	✓	✓					

- Not Granted ✓ - Granted ✗ -Denied

Note - All Permissions are Most Restrictive

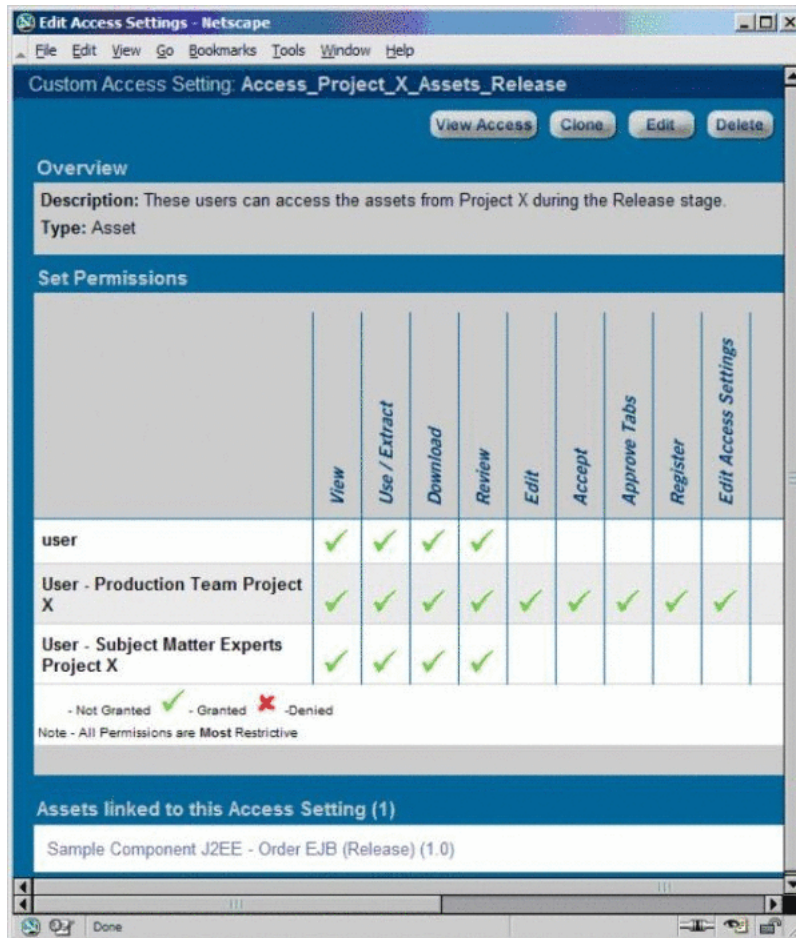
**Assets linked to this Access Setting (1)**

Sample Component J2EE - Order EJB (Build) (1.0)

- Asset Lifecycle Stage - Release
  - CAS: Access\_Project\_X\_Assets\_Release



**Figure 11–47 Edit Access Settings**



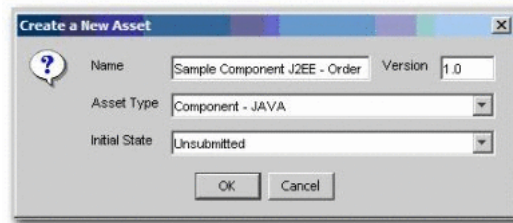
### 11.8.4.3 Configuring the Use Case Solutions

The following section outlines the necessary steps for each phase of the Asset Lifecycle Management use case.

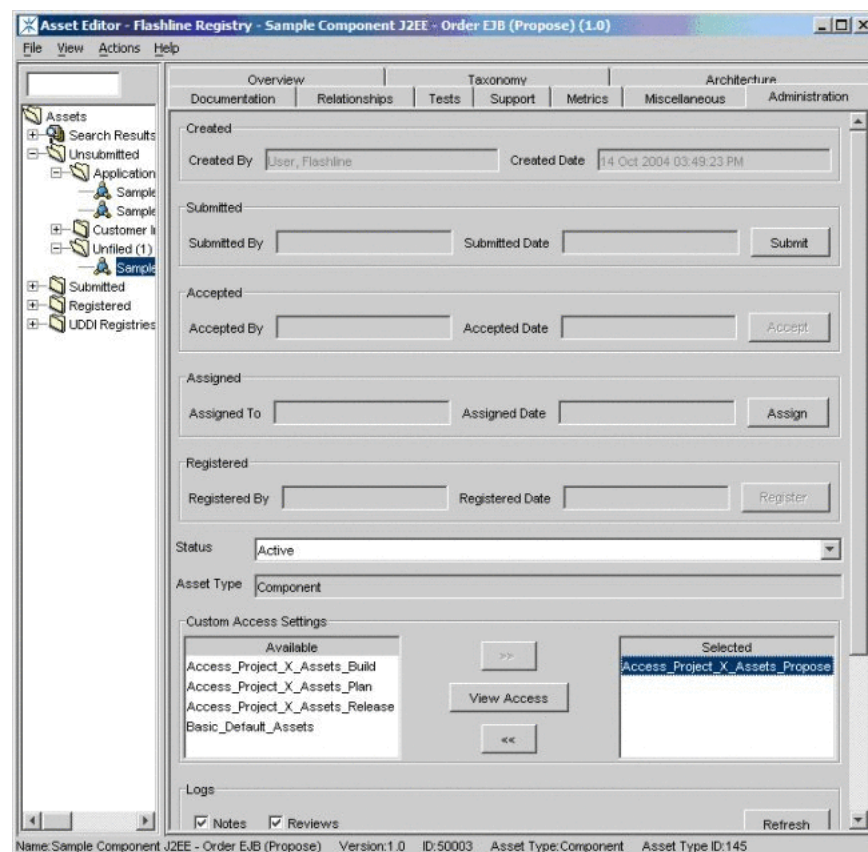
#### 11.8.4.3.1 Requirements Gathering Stage Solution

(Prerequisite: Verify that the Asset Lifecycle Categorization taxonomy is included as part of your selected asset type.)

1. Click the **Assets** link in the Oracle Enterprise Repository menu bar.
2. Click **Edit/Manage Assets** to launch the Asset Editor.
3. Open the File menu in the Asset Editor.
4. Click **New**.
5. Create an asset with the initial state of Unsubmitted, as shown in [Figure 11–48](#).

**Figure 11–48 Create a New Asset Dialog**

6. On the new asset's Taxonomy tab in the Asset Editor, find the Asset Lifecycle Stages categorization.
7. Select **Stage 1 - Propose**.
8. On the asset's Administration tab, add the new `Access_Project_X_Assets_Propose` asset CAS, as shown in [Figure 11–49](#).

**Figure 11–49 Asset Editor**

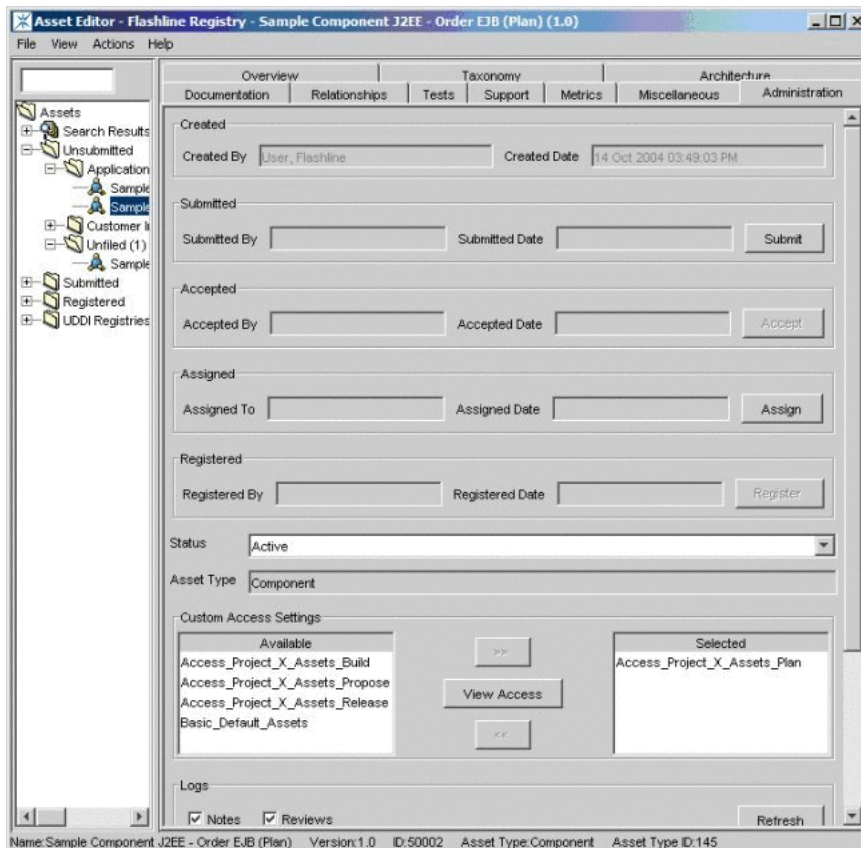
9. Briefly describe the asset and its purpose in the appropriate text boxes on the asset's General tab in the Asset Editor, and add a forum for use in gathering the asset's functional requirements.
10. View the asset in the Asset Lifecycle Stages categorization in Oracle Enterprise Repository, in the Stage 1 - Propose folder, as shown in [Figure 11–50](#).

**Figure 11–50 Browse Asset Tree**



- 11.8.4.3.2 Design and Development Stage Solution 1.** In the Asset Editor, this asset remains in the Unsubmitted state.
2. Choose **Stage 2 - Plan** under the Asset Lifecycle Stages categorization.
  3. Remove the **Access\_Project\_X\_Assets\_Propose** asset CAS and add the **Access\_Project\_X\_Assets\_Plan** asset CAS.

**Figure 11–51 Asset Editor**



- 11.8.4.3.3 Beta Release Stage Solution 1.** In the Asset Editor, this asset remains in the Unsubmitted state.
2. Choose **Stage 3 - Build** under the Asset Lifecycle Stages categorization.



3. Remove the **Access\_Project\_X\_Assets\_Plan** asset CAS and add the **Access\_Project\_X\_Assets\_Build** asset CAS to the asset.

**11.8.4.3.4 Release Stage Solution 1.** In the Asset Editor, submit the asset. When accepted and registered by the Registrar, the asset's status changes from Unsubmitted to Registered.

2. Choose **Stage 4 - Release** under the Asset Lifecycle Stages categorization.

3. Remove the **Access\_Project\_X\_Assets\_Build** asset CAS and add the **Access\_Project\_X\_Assets\_Release** asset CAS to the asset.

**11.8.4.3.5 Scheduled for Retirement Stage Solution 1.** In the Asset Editor, change the status of the asset from Active to Inactive.

**11.8.4.3.6 Retired Stage Solution 1.** In the Asset Editor, change the status of the asset from Inactive to Retired.

#### 11.8.4.4 Asset Lifecycle Management Solution Validation Tests

Confirm the following conditions for each role, as indicated:

- **Role**

- Project\_X\_Producer

- **Access**

- Has Registrar permissions throughout the lifecycle of the asset
- Can view the asset in Oracle Enterprise Repository
- The Use/Download and Edit buttons are visible
- Can accept, approve tabs, register, and edit access settings for the asset in the Asset Editor.
- In the Retired phase, the asset does not appear in the repository list of assets or searches. Repository access to the asset is possible for this role only when the asset is related to another asset, via the Relationship link. However, the asset remains accessible in the Asset Editor.

- **Role**

- Project\_X\_Subject\_Matter\_Expert

- **Access**

- In the Requirements Gathering phase:
  - \* Can view the asset, submit reviews, and subscribe to the asset (via the Subscribe button)
  - \* Has access to the asset's forum and may contribute comments
- In the Design and Development phase:
  - \* Can view the asset, submit reviews, and subscribe to the asset (via the Subscribe button)
  - \* Has access to the asset's forum and may contribute comments
- In the Beta Release phase:
  - \* Can view the asset, submit reviews, subscribe to the asset (via the Subscribe button)

- \* Can download the asset (via the Use/Download button)
- \* Has access to the asset's forum and may contribute comments
- In the Release phase:
  - \* Can view the asset, submit reviews, subscribe to the asset (via the Subscribe button)
  - \* Can download the asset (via the Use/Download button)
  - \* Has access to the asset's forum and may contribute comments
- In the Scheduled for Retirement phase:
  - \* Can view the asset, submit reviews, subscribe to the asset (via the Subscribe button)
  - \* Has access to the asset's forum and may contribute comments
- In the Retired phase, the asset does not appear in the repository list of assets or searches. Repository access to the asset is possible only when the asset is related to another asset, via the Relationship link. The asset will display the same functions/permissions as are listed in the Scheduled for Retirement phase.
- **Role**
  - User\_Community
- **Access**
  - In the Requirements Gathering phase:
    - \* Has no access to the asset.
  - In the Design and Development phase:
    - \* Can view the asset, submit reviews, and subscribe to the asset (via the Subscribe button)
    - \* Has access to the asset's forum and may contribute comments
  - In the Beta Release phase:
    - \* Can view the asset, submit reviews, subscribe to the asset (via the Subscribe button)
    - \* Has access to the asset's forum and may contribute comments
  - In the Release phase:
    - \* Can view the asset, submit reviews, subscribe to the asset (via the Subscribe button)
    - \* Can download the asset (via the Use/Download button)
    - \* Has access to the asset's forum and may contribute comments
  - In the Scheduled for Retirement phase:
    - \* Can view the asset, submit reviews, subscribe to the asset (via the Subscribe button)
    - \* Can download the asset (via the Use/Download button)
    - \* Has access to the asset's forum and may contribute comments
  - In the Retired phase, the asset does not appear in the repository list of assets or searches. Repository access to the asset is possible only when the asset is

related to another asset, via the Relationship link. The asset will display the same functions/permissions as are listed in the Scheduled for Retirement phase.

## 11.8.5 Use Case 5: Limit Access to Source Code Files to Asset Production Teams

This use case is somewhat similar to Use Case #1 (Web Services). That use case used two different asset types for each Web service asset in order to manage the presentation and availability of internally-exposed and externally-exposed asset metadata. Use Case # 5 also involves the restriction of access to asset metadata and files, but relies on only one asset type.

This section contains the following topics:

- [Section 11.8.5.1, "Benefit"](#)
- [Section 11.8.5.2, "Overview"](#)
- [Section 11.8.5.3, "Solution"](#)

### 11.8.5.1 Benefit

The settings described in this scenario facilitate black-box reuse by limiting the access of certain users/roles to assets that include compiled code. Such black-box assets are used as-is, without modification to the source code. Black-box assets are high-value assets in that their use in projects generally results in significant maintenance savings.

### 11.8.5.2 Overview

In this scenario, developers in specified asset consumer roles are limited to access to compiled code only, while those responsible for the production and maintenance of assets retain access to both source and compiled code.

### 11.8.5.3 Solution

This section describes the following use cases:

#### 11.8.5.3.1 Create the Roles

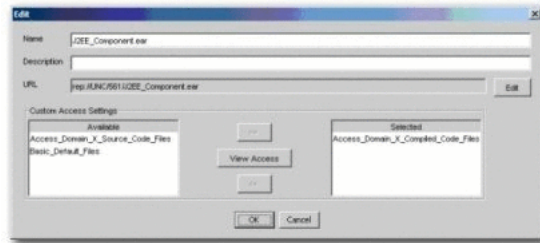
1. Click the **Admin** link in the Oracle Enterprise Repository menu bar.
2. On the Admin screen, click **Roles**.
3. Click **Create New**.
4. Create the following roles:
  - Production Team
  - Maintenance Team
  - Consumer

#### 11.8.5.3.2 Create the Customer Access Settings

1. Click the **Admin** link in the Oracle Enterprise Repository menu bar.
2. On the Admin screen, click **Custom Access Settings**.
3. Click **Create New**.
4. Create the following Custom Access Settings:
  - **Access\_Domain\_X\_Compiled\_Code\_Files**

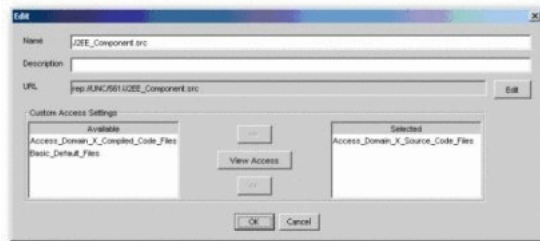
- For users who can see compiled code files.
  - **Access\_Domain\_X\_Source\_Code\_Files**
    - For users who can see sourced code files.
5. Remove the **Basic\_Default\_Files CAS** from the targeted assets. (This assumes that the Default File CAS has been set up.), as shown in [Figure 11-52](#).

**Figure 11-52 Edit Dialog**



6. Assign the **Access\_Domain\_X\_Compiled\_Code\_Files CAS** to all compiled code files, as shown in [Figure 11-53](#).

**Figure 11-53 Edit Dialog**

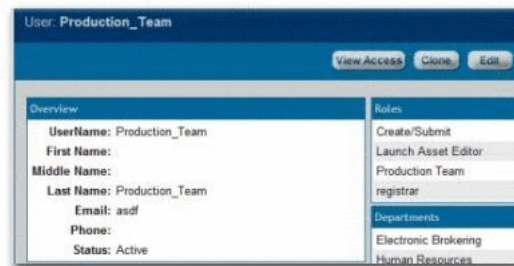


7. Assign the **Access\_Domain\_X\_Source\_Code\_Files CAS** to all source code files.

[Figure 11-54](#), [Figure 11-55](#), [Figure 11-56](#), [Figure 11-57](#), [Figure 11-58](#), and [Figure 11-59](#) illustrate the Production Team, Maintenance Team, and Consumer roles, and the relevant CAS settings respectively.

- **Production\_Team** (all production users): Roles

**Figure 11-54 User: Production\_Team**



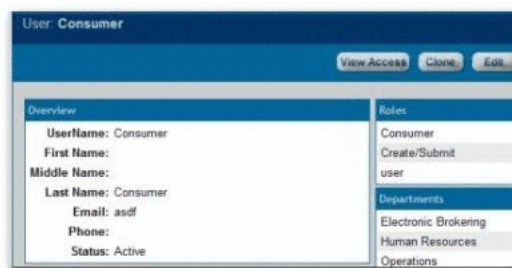
- **Maintenance\_Team** (all maintenance users): Roles

**Figure 11-55 User: Maintenance\_Team**



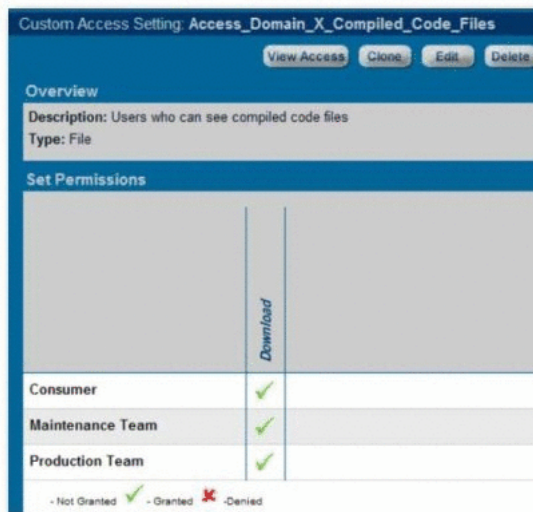
- Consumers (all consumers): Roles

**Figure 11-56 User: Consumer**



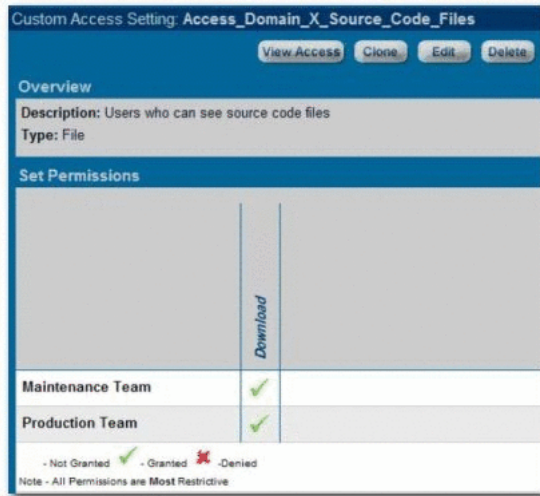
- Access\_Domain\_X\_Compiled\_Code\_Files

**Figure 11-57 Custom Access Setting**



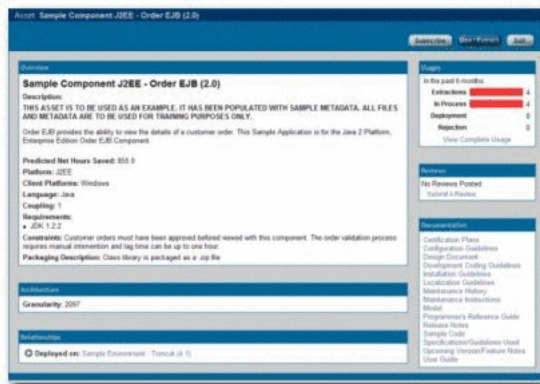
- Access\_Domain\_X\_Source\_Code\_Files

**Figure 11–58 Custom Access Setting**

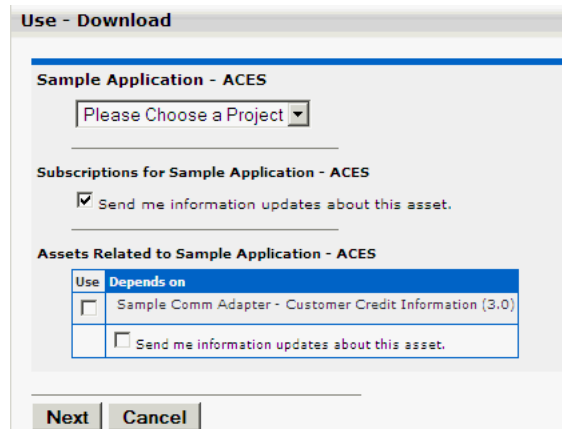


**11.8.5.3.3 Results** Production Team, Maintenance Team, and Consumer users see the same metadata information when viewing the asset in Oracle Enterprise Repository, as shown in [Figure 11–59](#).

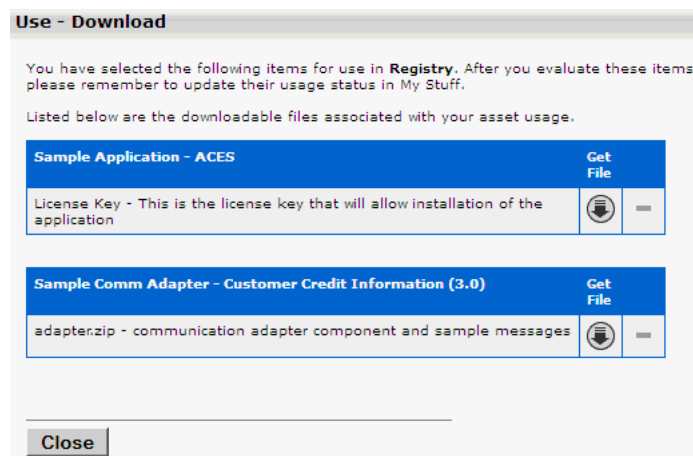
**Figure 11–59 Asset: Sample Component**



Production Team and Maintenance Team users see both the compiled and source code files during asset download, as shown in [Figure 11–60](#).

**Figure 11–60 Use / Extract Dialog**

Consumers see only the compiled code file during asset download, as shown in Figure 11–61.

**Figure 11–61 Use / Extract Dialog**

#### 11.8.5.3.4 Validation Test

Verify the following conditions for each role.

- Production\_Team and Maintenance\_Team:
  - Both source and compiled code files should be visible after clicking the Use/Download button.
  - The Edit button should be visible when viewing the asset in Oracle Enterprise Repository.
- Consumer
  - Only compiled code files are visible after clicking the Use/Download button.
  - The Edit button does not appear on assets for which the Consumer has access to compiled code only. If the Consumer has edit permission for assets containing source code, the source code files are visible to the Consumer in the Asset Editor.

## 11.8.6 Use Case 6: Grant Browse-only Repository Access to Specific Groups

This section describes the use case that demonstrates how to grant browse-only repository access to specific groups. This section contains the following topics:

- [Section 11.8.6.1, "Benefit"](#)
- [Section 11.8.6.2, "Overview"](#)
- [Section 11.8.6.3, "Solution"](#)
- [Section 11.8.6.4, "Validation Test"](#)

### 11.8.6.1 Benefit

The settings described in this scenario allow control over the rollout of the Repository within an organization by providing everyone with browse-only Repository access. This is helpful when promoting Repository use to a large group of development teams. Additional access can be assigned as necessary to the appropriate roles by the Access Administrator, as determined by organizational policy.

### 11.8.6.2 Overview

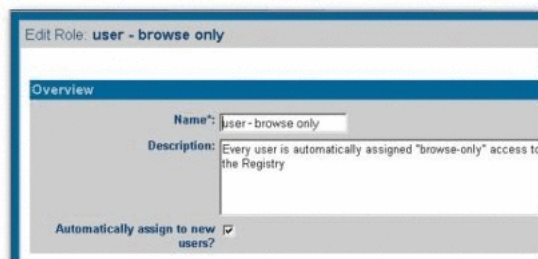
In this scenario, all Repository users are automatically assigned browse-only access. It is likely that users are authenticated against LDAP, and will gain access to the Repository using their standard organization-assigned login user name and password. As these users determine their need for additional functionality, they can be assigned additional roles by Repository administrators.

### 11.8.6.3 Solution

This section describes the solution for this use case.

1. Create the following role:
  - user - browse only
2. Ensure that this role is automatically assigned to new users, as shown in [Figure 11–62](#).

**Figure 11–62** *Edit Role: user-browse only*



3. Edit the User role to ensure that it is not automatically assigned to new users, as shown in [Figure 11–63](#).



**Figure 11–63 Edit Role: user**

Edit Role: **user**  
**Overview**  
 Name\*: user  
 Description: Normal User  
 Automatically assign to new users?

#### 4. Edit CAS: Basic\_Default\_Assets.

**Figure 11–64 Edit Custom Access Setting: Basic\_Default\_Assets**

Edit Custom Access Setting: **Basic\_Default\_Assets**  
**Overview**  
 Name\*: Basic\_Default\_As  Automatic  
 Description: Allows all default roles to access all assets through a Custom Ac  
 Type: Asset  
**Set Permissions**  
 Show only Roles with Assigned Permissions  
 Show all available Roles

	View	Use / Extract	Download	Review	Edit	Accept	Approve Tabs	Register	Edit Access Settings
advancedSubmitter	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
registrar	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
registrarAdministrator	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
user	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
user - browse only	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Not Granted  Granted  Denied

- Click the **Show all available roles** option to see the full list of available roles.
- Assign View access to user - browse only by checking the appropriate box.
- Save the CAS.

#### 11.8.6.4 Validation Test

Add a new user to the system:

- Verify that the new user is assigned browse-only privileges:
- Can view and subscribe to all assets
- Can access My Stuff
- Cannot submit assets
- Cannot download assets
- Cannot review assets



---

---

## Password Encryption

This chapter provides information about how you can encrypt passwords.

Password encryption is enabled by default within the Oracle Enterprise Repository, however, you may use the JVM startup parameter `cme.passwordencryption=false` to disable password encryption.

This chapter contains the following sections:

- [Section 12.1, "Generating Encrypted Passwords"](#)
- [Section 12.2, "Encrypting the Configuration File Passwords"](#)

### 12.1 Generating Encrypted Passwords

You can encrypt passwords using two different methods, one using the Oracle Enterprise Repository diagnostic page and other using the password encryption tool. To generate encrypted passwords, you need to perform the following steps:

1. Access the Oracle Enterprise Repository Diagnostics page.

Navigate to `http://host_name:port/application_name/diag/index.jsp` (replace `host_name` with the appropriate location).

---

---

**Note:** In Oracle Enterprise Repository 12c release, Oracle Enterprise Repository Diagnostics page is disabled by default.

To see if the diagnostics page is currently disabled, navigate to `http://host_name:port/application_name/diag/index.jsp` (replace `host_name` with the appropriate location). If the diagnostics page is disabled, the following message is displayed:

*Diag pages are currently disabled. Please contact your Oracle Enterprise Repository.*

See [Section 1.9, "Accessing Oracle Enterprise Repository Diagnostics Page"](#) for information about enabling the diagnostics page.

---

---

2. Scroll down to the Tools section and click the **Encrypt Strings for passwords** link to launch the Password encryption page.
3. Enter the clear text password into the String to Encrypt text box.
4. Click the **Submit Query** button.
5. Copy the resulting encrypted password string and paste it into the appropriate context or properties file(s).

The password encryption tool can be found at `<FMW_HOME>/oer/modules/tools/solutions/12.1.3.0.0-OER-PasswordTools.zip`. The `12.1.3.0.0-OER-PasswordTools.zip` file has two scripts (for Windows and Unix):

- `encrypt.bat/encrypt.sh` - encrypt an xml config file's password elements
- `encryptpassword.bat/encryptpassword.sh` - encrypt a single password from the command line

This section also contains the following topics:

- [Section 12.1.1, "Encrypted Passwords Used in Oracle Enterprise Repository Deployment"](#)
- [Section 12.1.2, "Other Passwords"](#)

### 12.1.1 Encrypted Passwords Used in Oracle Enterprise Repository Deployment

The suggested usage of encrypted passwords are as below:

- In the `database.properties` file  
The connection password for the database.
- The Ant task property file or build script  
The password the Oracle Enterprise Repository user will use at login.
- In the Harvester `HarvesterSettings.xml` configuration file  
The password stored in the `HarvesterSettings.xml` file.
- [In the Oracle Registry Repository Exchange Utility configuration \(`orrxu.xml`\) file](#)  
The password stored in the `orrxu.xml` file.
- [In the Oracle Enterprise Repository Workflow configuration \(`workflow.xml`\) file](#)  
The password stored in the `workflow.xml` file.
- [In the Export to API Catalog configuration file](#)  
The passwords stored in the `oac.xml` file.

### 12.1.2 Other Passwords

Other passwords in the system are encrypted automatically. This operation is invisible to the user. A number of fields stored in the properties files are encrypted by default, including:

- `ldap.bindPassword`
- `enterprise.guest.password`

This encryption occurs when the properties are edited and saved. Automatic encryption of passwords during an upgrade is unavailable at this time.

Passwords stored with the artifact stores are stored in the database in an encrypted format.

## 12.2 Encrypting the Configuration File Passwords

To ensure security, the passwords in the configuration files must be encrypted. You need to encrypt the configuration file passwords for the following:

- [Section 12.2.1, "Harvester Configuration File"](#)

- Section 12.2.2, "Exchange Utility Configuration File"
- Section 12.2.3, "Workflow Configuration File"
- Section 12.2.4, "Export to API Catalog Configuration File"

## 12.2.1 Harvester Configuration File

To ensure security, the passwords in the harvester configuration must be encrypted. The password encryption tool, (`encrypt.bat/encrypt.sh`), which is found at `oer/modules/tools/solutions/12.1.3.0.0-OER-PasswordTools.zip`, allows you to encrypt the passwords that are stored in the Harvester configuration (`HarvesterSettings.xml`) file.

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

```
> encrypt.bat HarvesterSettings.xml HarvesterSettings.xml
```

where

`HarvesterSettings.xml` = the Harvester configuration file.

## 12.2.2 Exchange Utility Configuration File

For enhanced security, the password encryption tool (`encrypt.bat/encrypt.sh`), which is packaged with the Exchange Utility kit, resides in the installation directory, and allows you to encrypt the passwords that are stored in the Oracle Registry Repository Exchange Utility configuration (`orrxu.xml`) file.

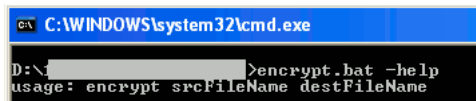
1. Navigate to the `<ExchangeUtility Tool Home>` directory.
2. From a command prompt, as shown in Figure 12–1, run the password encryption tool as follows:

```
> encrypt.bat orrxu.xml orrxu.xml
```

where:

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

**Figure 12–1 Encrypt Password Tool**



This figure illustrates the Encrypt Password tool.

\*\*\*\*\*

Figure 12–2 describes a sample image of how the password field appears before the encryption.

**Figure 12–2 Example Image of Password Before Encryption**

```

<!--Connection info to OER-->
<repository>
  <uri>http://localhost:8080/oer</uri>
  <credentials>
    <user>admin</user>
    <password>admin</password><!--run encrypt.bat to encrypt this-->
  </credentials>
</repository>

```

This figure illustrates the example image of password before encryption.

\*\*\*\*\*

Figure 12–3 describes a sample image of how the password field appears after you run the password encryption tool.

**Figure 12–3 Example Image of Password After Encryption**

```

<!--Connection info to OER-->
<repository>
  <uri>http://localhost:8080/oer</uri>
  <credentials>
    <user>admin</user>
    <password>v2_1.qRhDT11LdPo=</password><!--run encrypt.bat to encrypt this-->
  </credentials>
</repository>

```

This figure illustrates the example image of password after encryption.

\*\*\*\*\*

### 12.2.3 Workflow Configuration File

For enhanced security, the password encryption tool (`encrypt.bat/encrypt.sh`), which resides in the `12.1.3.0.0-OER-PasswordTools.zip` file, allows you to encrypt the passwords that are stored in the Workflow configuration (`workflow.xml`) file.

1. Navigate to the `ORACLE_HOME/oer/modules/tools/solutions` directory.
2. Extract the `12.1.3.0.0-OER-PasswordTools.zip` file to a directory and open a command prompt at this directory location.
3. From the command prompt, run the password encryption tool as follows:

```
> encrypt.bat workflow.xml workflow.xml
```

where

`workflow.xml` = the Workflow configuration file

### 12.2.4 Export to API Catalog Configuration File

To ensure security, the passwords in the Export to API Catalog configuration file (`oac.xml`) must be encrypted. The password encryption tool, (`encrypt.bat/encrypt.sh`), which is found at `oer/modules/tools/solutions/12.1.3.0.0-OER-PasswordTools.zip`, allows you to encrypt the passwords that are stored in the Export to API Catalog configuration file (`oac.xml`).

1. Navigate to the `ORACLE_HOME/oeer/modules/tools/solutions` directory.
2. Extract the `12.1.3.0.0-OER-PasswordTools.zip` file to a directory and open a command prompt at this directory location.
3. From a command prompt, run the password encryption tool as follows:

```
> encrypt.bat <Middleware_Home>/user_projects/domains/<oeer_
domain>/config_oeer/oac.xml <Middleware_Home>/user_
projects/domains/<oeer_domain>/config_oeer/oac.xml
```

where

`oac.xml` = the Export to API Catalog configuration file

`<Middleware_Home>` = the Middleware Home directory into which OER is installed

`<oeer_domain>` = the OER domain home directory.





# Part IV

---

## Advanced Lifecycle Administration

This part describes advanced administration tasks for the Oracle Enterprise Repository lifecycle.

This part includes the following chapters:

- [Chapter 13, "Configuring Oracle Enterprise Repository Express Workflows"](#)
- [Chapter 14, "Configuring Prescriptive Reuse"](#)
- [Chapter 15, "Integration with Code Compliance Inspector"](#)
- [Chapter 16, "Configuring Policy Management"](#)
- [Chapter 17, "Enterprise Manager Integration Utility"](#)
- [Chapter 18, "Configuring Metrics"](#)
- [Chapter 19, "Configuring Assets in Progress"](#)
- [Chapter 20, "Configuring Contract Management"](#)
- [Chapter 21, "Configuring Oracle Enterprise Repository to Exchange Metadata with the Oracle Service Registry"](#)



---

---

# Configuring Oracle Enterprise Repository Express Workflows

This chapter describes how to set up and configure Oracle Enterprise Repository (OER) Express Workflows.

This chapter assumes that you are familiar with the following concepts:

- Assets and Asset Types, Asset submission, Asset Registration Status, Asset Categorizations. See *Oracle Fusion Middleware Concepts Guide for Oracle Enterprise Repository* for more information.
- Projects and Users. See [Chapter 1, "Basic Configuration"](#) for more information.
- Oracle SOA Suite application deployment. See *Oracle Fusion Middleware Developer's Guide for Oracle SOA Suite* for more information.

---

---

**Note:** A restricted-use license for Oracle BPM Suite is included with OER. This license allows you to modify existing workflows that are supplied with OER. You can also create new OER-centric workflows using this license. If you want to create workflows that do not include OER, you must acquire a full license for the Oracle BPM Suite.

---

---

This chapter contains the following sections:

- [Section 13.1, "Introduction to Oracle Enterprise Repository Express Workflow"](#)
- [Section 13.2, "Configuring Express Workflows"](#)
- [Section 13.3, "Running the Sample Workflow"](#)
- [Section 13.4, "Modifying the Sample Workflow"](#)

## 13.1 Introduction to Oracle Enterprise Repository Express Workflow

Express Workflows, as described in this section, require Oracle SOA Suite and Oracle BPM Suite, both version 11.1.1.7 or higher.

---

---

**Note:** It is recommended to have your SOA Suite, on which the Express Workflows SCA will be deployed, in its own managed server. Express Workflows might not function correctly on a Developer edition of SOA Suite, which deploys SOA Suite in the Admin Server.

---

---

Express Workflows in Oracle Enterprise Repository is a mechanism for automating the common asset lifecycle approval processes. OER provides a set of features and options for moving an asset from an initial to a final state. Express Workflows requires an asset approval process to include a combination of actions provided by the OER Asset Editor:

- Asset Submission
- Asset Acceptance
- Asset Tab Approvals
- Asset Registration

Value changes to the `AssetLifeCycleStage` and `Classification` categorizations are also necessary.

Organizations that need more complex workflow automation can also develop external workflows with other tools, such as Oracle Business Process Management Suite, and use the REX API interface described in the "Using the Repository Extensibility Framework" chapter of the *Oracle Fusion Middleware Developer's Guide for Oracle Enterprise Repository* to access OER functions.

Express Workflows includes the following components:

- Assets
- Projects
- Users
- Approval Flows
- Events and Actions

These components are related by an Express Workflow as follows:

- An asset is submitted into the repository with a producing project. Note that Express Workflows require assets to have one and only one producing project.
- The asset will automatically enter the flow upon submission or acceptance, if the indicated producing project has an approval flow assigned, and if the asset's type is identified for the approval flow. A project can have only one approval flow, and that gets kicked off only when relevant assets in that project are submitted or accepted.
- An approval flow consists of groups, tiers, of asset tabs. Each tier has an assigned approver. The asset moves to the next tier, when the assigned approver has approved all the tabs in the tier.

An approval flow has settings determining if:

- the flow begins on asset submission or acceptance
- the asset is automatically registered when specified tabs are approved.

An OER administrator can create any number of approval flows and an approval flow can be assigned to any number of projects. Assets are manipulated by mappings of specific actions to discrete events, such as applying a custom access setting to an asset when it is registered or notifying subscribers when an asset's Life Cycle Stage categorization value changes.

## 13.2 Configuring Express Workflows

Express Workflows depends on these components:

- Express Workflows SOA composite application (SCA)
- OER Event Manager
- Express Workflows configuration file

To configure Express Workflows:

1. Install the OER Express Workflow SOA Composite Application (SCA)
2. Configure OER Event Manager to deliver events to the deployed SCA
3. Define and configure your Express Workflow(s)

This section describes how to install the Oracle Enterprise Repository Workflow into Oracle BPM 11g and 12c. It contains the following topics:

- [Section 13.2.1, "Step 1: Install the OER Express Workflows SOA Composite Application"](#)
- [Section 13.2.2, "Step 2: Configure OER Event Manager to deliver events to the deployed SCA"](#)
- [Section 13.2.3, "Step 3: Define and Configure your Express Workflows"](#)

### 13.2.1 Step 1: Install the OER Express Workflows SOA Composite Application

The Express Workflows SOA composite application listens for events delivered by the OER Event Manager to its web service endpoint. Installing this application requires an installed instance of Oracle SOA Suite and Oracle BPM Suite, both version 11.1.1.7 or higher. No additional configuration is required in your SOA Suite and BPM Suite servers.

---

**Note:** Ensure that payload validation under soa-infra-->SOA Administration-->Common Properties is set to *false* before you proceed.

---

#### Deploying the Express Workflows SOA composite application on WebLogic Server

1. Log into the SOA Suite Enterprise Manager console, typically on, `http://<soaserver_hostname>:<soaserver_port>/em`.
2. On the left-side menu, expand **SOA**.
3. Click **soa-infra**.
4. Select **Deployed Composites** tab.
5. Click **Deploy....**
6. The Select Archive window appears.
7. Select the appropriate option in the **Archive** or **Exploded Directory** region.
8. Upload the composite delivered with the ExpressWorkflows zip file:
  - a. Navigate to the `<FMW_Home>/oer/modules/tools/solutions/` directory.
  - b. Expand the `12.1.3.0.0-ExpressWorkflows.zip` file.
  - c. Select the `sca_OERWorkflows_rev1.0.jar` file.
9. Click **Next**.
10. Select the default SOA partition. Click **Next**.

## 11. Click **Deploy**.

### 13.2.2 Step 2: Configure OER Event Manager to deliver events to the deployed SCA

The Event Manager is a component embedded within Oracle Enterprise Repository that manages event subscriptions, event persistence, event filtering, and event delivery.

External applications subscribe to the events generated as assets move through a lifecycle. For Express Workflows, a SOA composite application, deployed to an Oracle SOA Suite server instance, accesses these events via a web service endpoint defined in the OER configuration file.

Express Workflows supports the following events:

- Asset submitted
- Asset unsubmitted
- Asset accepted
- Asset unaccepted
- Asset tab approved: value=<tab name>
- Asset all tabs approved
- Asset registered
- Asset unregistered
- Asset status changed: value="DELETED/INACTIVE/RETIRED"
- Policy Assertion changed: value="pass/fail"
- Categorization changed: AssetLifecycleStage (act on any change)
- Categorization changed: AssetLifecycleStage: value="stage name"
- Categorization changed: Classification (act on any change)
- Categorization changed: Classification: value="stage name"

#### 13.2.2.1 Configuring Web Service Endpoints

The Event Manager uses a configuration file to load information about the Web Service endpoints, where events are delivered. This file is delivered in the `ExpressWorkflows.zip` file at `<FMW_HOME>oer/modules/tools/solutions`. The `EndPointEventSubscription.xml` file should be extracted from `ExpressWorkflows.zip`, modified appropriately, and pasted into the `<FMW_HOME>/oer/applications/oer/oer-app/WEB-INF/classes` directory.

In this file, to add your SCA endpoint to receive events, create a new instance of the `<sub:eventSubscription>` element as in the example below:

```
<sub:eventSubscription>
<!--The name should be unique within this file since this name is passed as the
Durable subscriber name to the JMS Server-->
<sub:endPoint name="SOACompositeEndpoint">

<!--The host of the SOA Server hosting the composite -->
<sub:host>localhost</sub:host>
<!--The SOA Server listener port -->
<sub:port>9000</sub:port>
<!--The URI of the deployed composite -->
```

```

<sub:uri>soa-infra/services/default/OERWorkflows/HandleStatusChange.service</sub:uri>
<!--Unless a custom WSDL Contract is used, the namespace should not be changed -->
<sub:targetNamespace>http://www.bea.com/infra/events</sub:targetNamespace>

<!--The Webservice operation that is invoked. Please refer to the WSDL in
WEB-INF\lib\eventNotifier.jar for all the available operations-->
<!--Unless a Custom Webservice is implemented, the operation should not be changed
if it's ALBPM -->
<sub:operationName>newEvent</sub:operationName>
<!--Protocol for the SOA Suite server. Use HTTPS if SSL-enabled -->
<sub:protocol>HTTP</sub:protocol>
<!--The user and password to authenticate the composite deployed on the SOA Suite
server. -->
<sub:authenticationData>
<sub:basicAuthentication>
<sub:username>oer_workflow_user</sub:username>
<sub:password></sub:password>
</sub:basicAuthentication>
</sub:authenticationData>
</sub:endPoint>

<!--The Java class that serializes the Event Data. Unless a custom class is
written, this value should not be changed.-->

<sub:notifierClass>com.bea.infra.event.notifier.plugin.http.DefaultHTTPEventNotifi
er</sub:notifierClass>

<!--This expression filters the Event data and only the matched events are
delivered to the Endpoint. The default is all the events are delivered. -->
<!--Example:- asset_id BETWEEN 50000 AND 50100 -->
<sub:expression></sub:expression>
</sub:eventSubscription>

```

In this file, the host, port, username, and password of the Express Workflows SOA composite endpoint must be configured manually, as shown in the example above. The above example is available from `EndPointEventSubscription.xml`, delivered in the `12.1.3.0-ExpressWorkflows.zip` file located in the `<FMW_HOME>oer/modules/tools/solutions` directory.

---

**Note:** Passwords must be encrypted, using either of the two encryption tools provided with OER. See [Chapter 12, "Password Encryption"](#).

---

Changes to this file will not take effect until the OER managed server is stopped and restarted, which you will be doing after enabling the Event Manager next. If the Event Manager is already enabled, you can restart the OER server now.

### 13.2.2.2 Enabling the Event Manager

With the Event Manager configuration completed, you must enable the Event Manager in Oracle Enterprise Repository to allow it to send events to external Web Service endpoints.

1. Click **System Settings** in the sidebar of the Oracle Enterprise Repository Admin screen.

2. Enter event in the System Settings Search to view all the Event Manager related settings.
3. Click **True** next to the Enable Event Manager property, `cmee.eventframework.enabled`.
4. Click **Save**.
5. Restart Oracle Enterprise Repository for the configuration changes to take effect.

### 13.2.3 Step 3: Define and Configure your Express Workflows

The Express Workflows components and their relationships are defined and configured in an XML file, which is read by the Express Workflows SOA composite application whenever an OER event is delivered to the application's web service endpoint.

A sample version of this file (and its XSD) is provided with a sample workflow preconfigured to run when the configuration steps are completed. This sample file is at `<FMW_HOME>/oer/modules/tools/solutions/12.1.3.0.0-ExpressWorkflows.zip`

A more detailed explanation of the different elements of this XML is provided in [Section 13.4, "Modifying the Sample Workflow"](#).

#### Setting up the configuration file and enabling the sample workflow

1. Create a new OER user with the required permissions to launch the Asset Editor, edit assets, and approve tabs. See [Chapter 1, "Basic Configuration"](#) for information on users, roles and Basic Access Settings.
2. Save the original sample `ExpressWorkflow.xml` with a new filename.
3. Edit `ExpressWorkflow.xml` by entering the host and port of your OER server and the clear text password of your admin user in the `<oerConnection>` tag:

```
<oerConnection>
  <uri>http://[OERHOST:PORT]/oer/services/FlashlineRegistry</uri>
  <registrar>
    <username>admin</username>
    <password>[your admin password]</password>
  </registrar>
```

4. Enter the username of your new OER user as the approver of tier2:

```
<approvalFlow name="Flow1" acceptOnAssetSubmission="true"
registerOnFlowCompletion="true">
  <tiers>
    <tier name="Tier1">
      <approvers>
        <oeruser>
          <username>admin</username>
        </oeruser>
      </approvers>
    </tier>
    <tier name="Tier2">
      <approvers>
        <oeruser>
          <username>[Your New Username]</username>
        </oeruser>
      </approvers>
    </tier>
  </tiers>
  <tabs>
    <tab name="Overview"/>
    <tab name="Technical"/>
    <tab name="Documentation"/>
  </tabs>
</approvalFlow>
```



```

        </approvers>
        <tabs>
            <tab name="Taxonomy" />
        </tabs>
    </tier>

```

5. Encrypt the passwords in the file as per [Chapter 12, "Password Encryption"](#).
6. Save the file.
7. Verify the OER server is running, then run the Express Workflows validation script, delivered in `<FMW_Home>/oer/modules/tools/solutions/12.1.3.0.0-ExpressWorkflows.zip`, to check your `expressWorkflow.xml` file:
 

```
validateWorkflow.sh -settings [your expressWorkflow.xml file]
```
8. Create a new folder in your Oracle SOA Suite domain called `oerconfig` as follows:
 

```
<FMW_Home>\user_projects\domains\<SOADomain>\oerconfig
```
9. Copy the validated file to the `oerconfig` folder with the name `ExpressWorkflow.xml`.

Your Express Workflows are now set up and configured with an enabled sample. You are now ready to move on to section [Section 13.3](#) and run the sample.

## 13.3 Running the Sample Workflow

With your OER and SOA servers running, create a new asset of type **Service** in the repository.

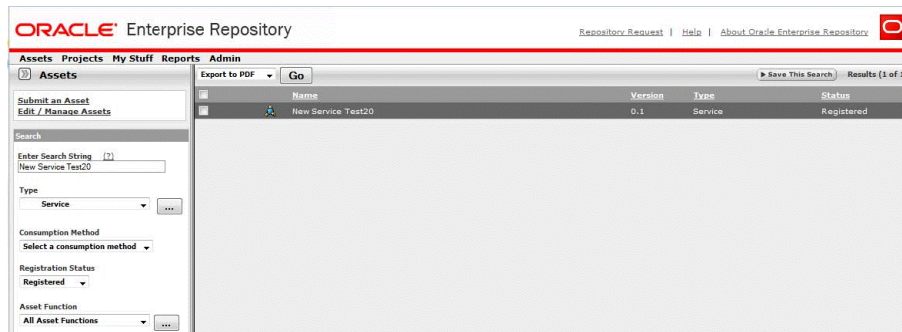
### Create a New Asset

1. Login to OER as `admin` or any user with asset creation rights.
2. From the Asset Editor, select **New** from the File menu.
3. In the Create a New Asset field, enter the asset name.
4. Select type of **Service** and initial state of **Unsubmitted**.
5. Click **OK**. The new asset tabs display.
6. On the Overview tab, select **Common Project** for Producing Project. Click **Save**.

### Run the Sample

1. Login to OER as an admin user.
2. Go to **My Stuff**, where you see the newly created and submitted asset in your queue.
3. Open the asset editor and approve the **Overview**, **Technical** and **Documentation** tabs. Exit the Asset Editor.
4. Log out and log back into OER.
5. Go to **My Stuff**, find the Service asset and edit it to approve the **Taxonomy** tab. Exit the Asset Editor.
6. In the OER console, search for the Service Asset by name, with Registration Status set to **Registered**.

You will see the asset in the search results.



## 13.4 Modifying the Sample Workflow

Once you have successfully run the sample workflow, you can modify it and/or create new workflows as follows:

1. Copy the installed `ExpressWorkflow.xml` to a new location and edit it to add new projects, users, asset types, event/action mappings and approval flows.
2. Run the validation script to ensure your changes are compatible with your running OER instance.
3. Replace the installed `ExpressWorkflow.xml` file with your new version.

---

**Note:** Note the Express Workflows SOA composite reads the installed file every time it receives an event from the OER event manager, so any changes in a newly installed `ExpressWorkflow.xml` version will take effect immediately without a need to restart.

---

### 13.4.1 Modifying the Express Workflows Configuration File

Previously, you modified the `ExpressWorkflow.xml` file to:

- identify your OER Server instance,
- supply an encrypted password for the OER admin user,
- modify the approver name for the second tier of asset tabs in the approval flow `Flow1`

#### 13.4.1.1 Producing Projects

Here is an excerpt from the default `ExpressWorkflow.xml` file for producing project Common Project:

```

        <producingProject name="Common Project" approvalFlow="Flow1"/>
<assetTypes>
  <assetType name="Business Process"/>
  <assetType name="Composite"/>
  <assetType name="Service"/>
</assetTypes>
</producingProject>

```

- `name` refers to an existing OER project. The validation script will generate an error if the specified project does not exist in the OER instance.
- `approvalFlow` is an optional attribute which specifies a tiered tab approval, or governance flow, for assets created within this project. A producing project can have one `approvalFlow` or none.

- `assetType` refers to existing types. All assets of the specified types, submitted within the producing project, will follow the project's specified `approvalFlow` and any event/action mappings specified for that asset type

---

**Note:** Assets can be governed by Express Workflows only if they have one and only one producing project specified in the `ExpressWorkflow.xml` file.

---

### 13.4.1.2 Approval Flows

Express Workflows approval flows determine the behavior of specified assets as the assets tabs are approved during the governance process. Here is the `ExpressWorkflow.xml` excerpt, which you modified previously:

```
<approvalFlow name="Flow1" acceptOnAssetSubmission="true"
registerOnFlowCompletion="true">
  <tiers>
    <tier name="Tier1">
      <approvers>
        <oeruser>
          <username>admin</username>
        </oeruser>
      </approvers>
      <tabs>
        <tab name="Overview"/>
      </tabs>
    </tier>
    <tier name="Tier2">
      <approvers>
        <oeruser>
          <username>[Your New Username]</username>
        </oeruser>
      </approvers>
      <tabs>
        <tab name="Taxonomy"/>
      </tabs>
    </tier>
  </tiers>
</approvalFlow>
```

- `approvalFlow` name identifies the flow and is used in the `approvalFlow` attribute of `producingProject`. This attribute is not used outside the `expressWorkflow.xml` file.
- `acceptOnAssetSubmission` is an optional attribute which determines whether a specified asset will be accepted automatically when it is submitted (`value="true"`) or whether the asset will be manually accepted by an OER user with sufficient access rights (`value="false"`). If this attribute is not specified, the default value is `"false"`.
- `registerOnFlowCompletion` is an optional attribute which determines whether a specified asset will be registered automatically when all of the tabs specified in the approval flow have been approved (`value="true"`). If this attribute is not specified, the default value is `"false"`.
- In the `<tier>` element, `tier` name identifies a combination of asset tabs.

- In the <tier> element, *approver* indicates an existing OER user to whom the asset will be assigned, either upon asset acceptance for the first specified tier, or upon approval of all tabs of a previous tier.
- In the <tier> element, *tab name* indicates specific asset tabs that must exist for the specified asset types, which will be approved within a particular tier.

### 13.4.1.3 Event-to-Action Mappings

Event-to-Action mappings offer an alternative method to specifying an approval flow for automatically accepting and registering an asset. These mappings apply only to the specified asset types. Different asset types within a project can have different (or the same) mappings.

Express Workflows recognizes a number of asset lifecycle events generated by the Event Manager and provides a mechanism for associating these events with specific actions on specific assets. The full list of supported events and actions are included further below in the syntax required for the `ExpressWorkflow.xml` file.

Examine the following sample `producingProject` element:

```
<producingProject name="Project1" />
  <assetTypes>
    <assetType name="Service">
      <events>
        <event name="urn:com:bea:aler:events:type:AssetSubmission">
          <actions>
            <action name="AcceptAsset" />
          </actions>
        </event>
        <event name="urn:com:bea:aler:events:type:AssetTabApproved">
          <actions>
            <action name="RegisterAsset" />
          </actions>
        </event>
      </events>
    </assetType>
  </assetTypes>
  <producingProject>
```

In the above example, `Project1` does not have an approval flow specified. It does, however, have event-to-action mappings specified for assets of type `Service`. The first mapping will automatically accept any asset of type `Service` when it is submitted. The second will register the asset when the "Management Review" tab is approved.

In the following example, we automatically apply a Custom Access Setting to assets of type `Service` when they are registered by the approval flow, `Flow1` from the default `expressWorkflow.xml` file:

```
<producingProject name="Project2" approvalFlow="Flow1" />
  <assetType name="Service">
    <event name="urn:com:bea:aler:events:type:AssetRegistered">
      <action>ChangeCAS</action>
      <customAccessSettings>
        <customAccessSetting>[YourSetting]</customAccessSetting/>
      </customAccessSettings>
    </assetLifecycle/>
```

```

    </event>
  </assetType>
</producingProject>

```

Custom Access Settings can be used to control the visibility of an asset within a governance community, so the setting in this example would most likely be used to expose an asset to potential consumers after the approval process was complete.

In general, producing projects can either specify approval flows, or specify event-to-action mappings for specific asset types, or both. When both are used, however, you must take care to avoid overlapping or confusing rules, for instance specifying tabs for approval in an approval flow tier and specifying an Approve Tab action within the an event-to-action mapping.

Following is a list of events and actions supported by Express Workflows.

### Supported Events

- urn:com:bea:aler:events:type:AssetSubmission
- urn:com:bea:aler:events:type:AssetUnSubmission
- urn:com:bea:aler:events:type:AssetAccepted
- urn:com:bea:aler:events:type:AssetUnAccept
- urn:com:bea:aler:events:type:AssetTabApproved
- urn:com:bea:aler:events:type:AssetAllTabApproved
- urn:com:bea:aler:events:type:AssetRegister
- urn:com:bea:aler:events:type:AssetUnregister
- urn:com:bea:aler:events:type:PolicyAssertionChanged
- urn:com:bea:aler:events:type:CategorizationChanged:assetLifecycleStage
- urn:com:bea:aler:events:type:CategorizationChanged:classification

### Supported Actions

- ChangeCAS: applies one or more Custom Access settings to an asset
- ChangeAssetLifecycle: sets the Asset Lifecycle Stage of an asset
- ChangeClassification: sets the classification of an asset
- ReAssignAssetToRegistrar: assigns the asset to Registrar
- NotifySubscriber: notifies the Subscribers about the Metadata Change
- ApproveTab: approves one or more tab
- UnapproveTab: unapproves one or more tab
- AcceptAsset: accepts the asset
- UnacceptAsset: unaccepts the asset
- RegisterAsset: Unregisters the Asset if the asset is in registered state.
- UnRegisterAsset: Unregisters the Asset if the asset is in registered state.
- PublishAssetToUddi: Moves individual services and their metadata to Oracle Service Registry by wiring the events that get triggered when these services are registered or a lifecycle of a service is changed.



---

---

## Configuring Prescriptive Reuse

This chapter provides an overview of compliance templates, and their role in prescriptive reuse.

This chapter contains the following sections:

- [Section 14.1, "Overview"](#)
- [Section 14.2, "Creating a Compliance Template Type"](#)
- [Section 14.3, "Creating a Compliance Template"](#)
- [Section 14.4, "Apply a Compliance Template to a Project"](#)
- [Section 14.5, "Using Assets Associated with Compliance Templates \(Prescribed Assets\)"](#)
- [Section 14.6, "Running a Project Compliance Report"](#)

### 14.1 Overview

Oracle Enterprise Repository serves as a communication mechanism between stakeholders involved in project evaluation and developers who are involved in a project's implementation. Business Analysts or architects are responsible for reviewing a project's requirements. During this process, they might identify existing services or capabilities that will fulfill the functional and non-functional requirements of the project.

Oracle Enterprise Repository allows these stakeholders to construct a **Compliance Template** that includes a list of existing assets that might be relevant to the project. This process is called **prescriptive reuse**, and it is extremely valuable in contracting organizations. A contracting organization may specify or bid a project several months prior to the bid being accepted by the customer. Compliance templates become a way for these organizations to capture the assumptions that were made when the project bid was created. When the bid is accepted, the team executing the work has visibility into the assets that were assumed to be reusable.

The term Compliance Template describes a particular family of Oracle Enterprise Repository artifacts. Examples of specific compliance template types include:

- Project Profiles
- Architecture Blueprints

Compliance Templates are used to communicate asset requirements or asset solution sets to internal or outsourced project teams. Project Profiles are usually created for individual projects, whereas Architecture Blueprints are reusable solution sets that can be leveraged by multiple projects.

Compliance Templates are typically generated by individuals or teams responsible for asset and project portfolios:

- Project Planners might generate a Project Profile for each project in the portfolio, outlining the planning and estimating assumptions.
- Project Architects might generate a Project Profile to identify assets that fulfill a project's business requirements.
- Enterprise Architects might generate an Architecture Blueprint that specifies the standard frameworks and servers that are to be used by new projects.
- Those responsible for Service Oriented Architectures might generate an Architecture Blueprint to identify the services that orchestrate a particular business function.
- Product Line Architects might generate an Architecture Blueprint that specifies the assets that are to be used on a specific product line.

If upgrading from earlier product versions, compliance template examples can be obtained by installing the examples in the Policy Management datapack using the Import/Export tool. For more information about the Import/Export tool, see [Chapter 5, "Import Export Tool"](#).

## 14.2 Creating a Compliance Template Type

This procedure is performed in the Type Manager. For more information about running the Type Manager, see [Chapter 7, "Using the Type Manager"](#). This example illustrates the creation of a Project Profile compliance template type.

---



---

**Caution:** Adding a compliance template constitutes a permanent change to the Oracle Enterprise Repository system.

---



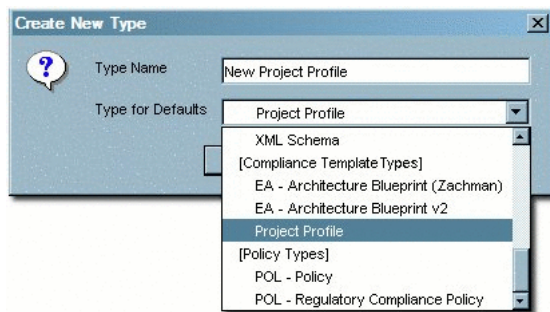
---

In order to preserve database integrity, only Compliance Templates to which no assets are currently associated may be deleted.

### If a Compliance Template Type is available in the Type Manager:

1. Select **New** in the File menu. The Create New Type dialog is displayed, as shown in [Figure 14-1](#).

**Figure 14-1** Create New Type Dialog



2. Enter the appropriate information in the Type Name text box.
3. In the Type for Defaults list, select a Compliance Template Type from the list.



4. Click **OK**. The new Type is displayed in the tree in the Type Manager sidebar, as shown in [Figure 14-2](#).

**Figure 14-2 Type Manager**



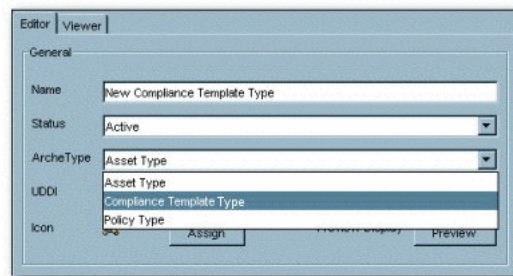
**If no Compliance Template Type is available in the Type Manager:**

Compliance templates and examples may be obtained from the Oracle Enterprise Repository Base datapack using the Import/Export Tool.

**If an alternative method is preferred, a new Compliance Template Type may be manually generated:**

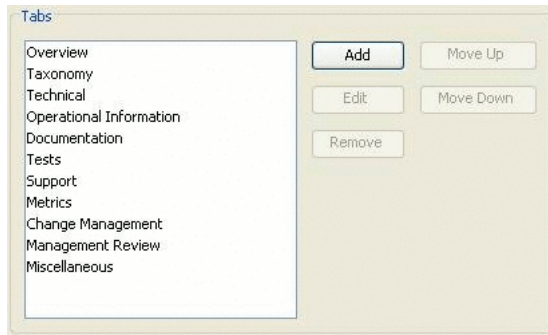
1. Select **New** in the File menu. The Create New Type dialog is displayed.
2. Enter the appropriate information in the Type Name text box.
3. In the Type for Defaults list, select any Type from the list.
4. Click **OK**. The new Type is displayed in the tree in the Type Manager sidebar.
5. Open the new compliance template type in the Type Manager.
6. Click the **Editor** tab of new compliance template type.
7. Select **Compliance Template Type** in the Archetype list in the General section of the Editor tab, as shown in [Figure 14-3](#).

**Figure 14-3 Editor Tab**



8. Click **Save** in the File menu.
9. Select one of the of the tabs listed in the Tabs section, as show in [Figure 14-4](#).

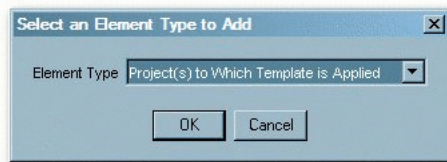
**Figure 14–4 Tabs Section**



The elements assigned to that tab appear in the list in the Elements section.

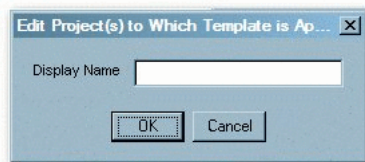
10. Click the **Add** button in the Elements section. The Select an Element Type to Add dialog is displayed, as shown in [Figure 14–5](#).

**Figure 14–5 Select an Element Type to Add Dialog**



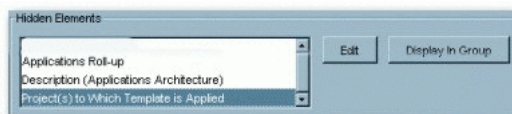
11. Select **Project(s) to Which Template is Applied** from the list in the Select an Element Type to Add dialog.
12. Click **OK**. The Select an Element Type to Add closes.
13. Follow the prompt to enter a label for the new element, as shown in [Figure 14–6](#).

**Figure 14–6 Edit Project(s) to Which Template is Applied Dialog**



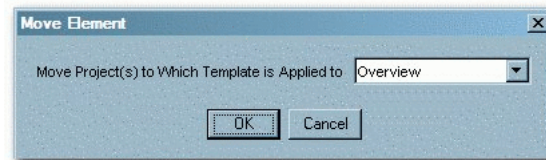
14. Click **OK**. The new element is displayed in the list in the Elements section.
15. Click **Save** in the File menu.
16. Click the **Viewer** tab of new compliance template type.
17. Locate and select the new element in the list of Hidden Elements, as shown in [Figure 14–7](#).

**Figure 14–7 Hidden Elements Section**



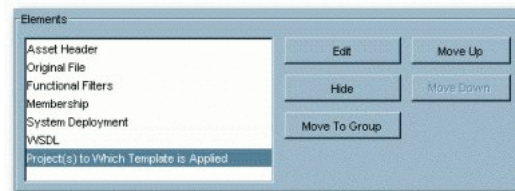
18. Click the **Display in Group** button. The Move Element dialog is displayed, as shown in [Figure 14–8](#).

**Figure 14–8** Move Element Dialog



19. Use the list in the Move Element dialog to select a location for the new element.
20. Click **OK**. The new element is displayed in the list in the Elements section of the Viewer tab.

**Figure 14–9** Elements Section



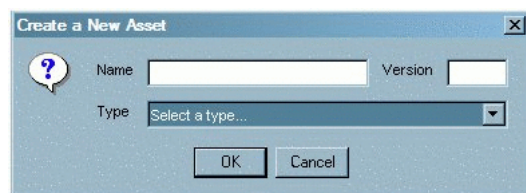
21. Click **Save** in the File menu.

## 14.3 Creating a Compliance Template

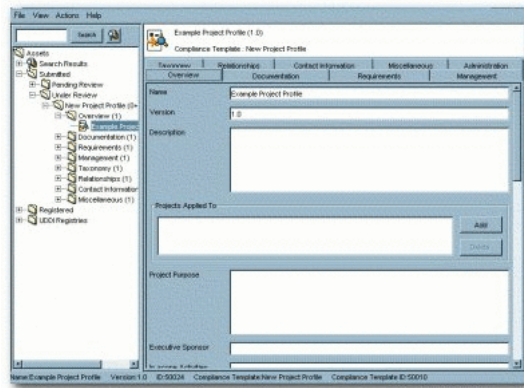
This procedure is performed in the Asset Editor, and is very similar to Advanced Asset Submission, as described in [Section 9.4, "Advanced Asset Submission"](#).

1. Select **New** in the File menu. The Create a New Asset dialog is displayed, as shown in [Figure 14–10](#).

**Figure 14–10** Create a New Asset Dialog



2. Enter the appropriate information in the Name and Version text boxes.
3. Select the appropriate Type template from the list.
4. Click **OK**. The new compliance template is displayed in the Asset Editor, as shown in [Figure 14–11](#).

**Figure 14–11 Asset Editor**

5. Click the **Taxonomy** tab, and under Relationships section, click **Add** to add relationships.
6. Select the assets that are required to be used by the project.
7. Assign the appropriate relationships: Prescribed Assets, Prescribed Processes, and Prescribed Environments, as appropriate.
8. Click **Save** in the File menu in the Asset Editor.

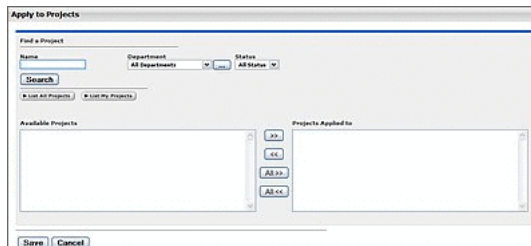
The asset is submitted for review and registration.

Follow this procedure to create Project Profiles, Architecture Blueprints, or other variations of the Compliance Template.

## 14.4 Apply a Compliance Template to a Project

This procedure begins on the Oracle Enterprise Repository Assets screen.

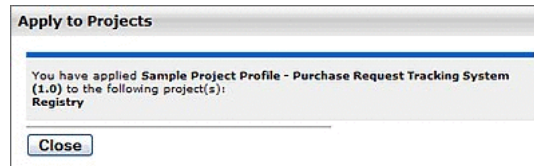
1. Use Search or other means to locate the compliance template to be applied to the project. (Select the desired compliance template in the Type list in the Search section to filter search results.)
2. Click **Apply to Projects**. The Apply to Projects dialog is displayed, as shown in [Figure 14–12](#).

**Figure 14–12 Apply to Projects Dialog**

3. Click **List All Projects** or use Search to locate the project to which the profile is applied. A list is displayed in the Available Projects column.
4. Select the project to which the profile is applied.

5. Click the right arrow button to move the project from the Available Projects column to the Projects Applied to column. The project moves from Available Projects to Projects Applied to.
6. Click **Save**. A confirmation dialog is displayed, as shown in [Figure 14–13](#).

**Figure 14–13 Apply to Projects Confirmation Dialog**



7. Click **Close**.
8. Click the **Projects** link in the Oracle Enterprise Repository menu bar. The Projects screen is displayed.
9. Use Search or other means to locate the project to which the project profile was applied.
10. Click the **Compliance Templates** tab in the Project Detail. The applied compliance template is listed in the Compliance Templates tab.

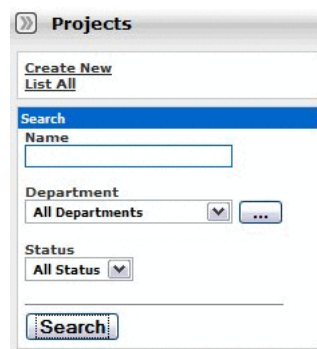
## 14.5 Using Assets Associated with Compliance Templates (Prescribed Assets)

A Compliance Template may be used to prescribe specific assets for use in any projects to which the compliance template is applied. This section outlines the process for consuming prescribed assets.

This procedure begins on the Oracle Enterprise Repository Projects screen.

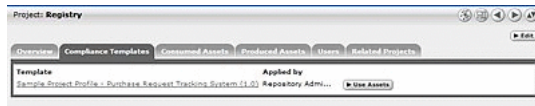
1. Use Search or other means, as shown in [Figure 14–14](#), to locate the project.

**Figure 14–14 Projects Screen**



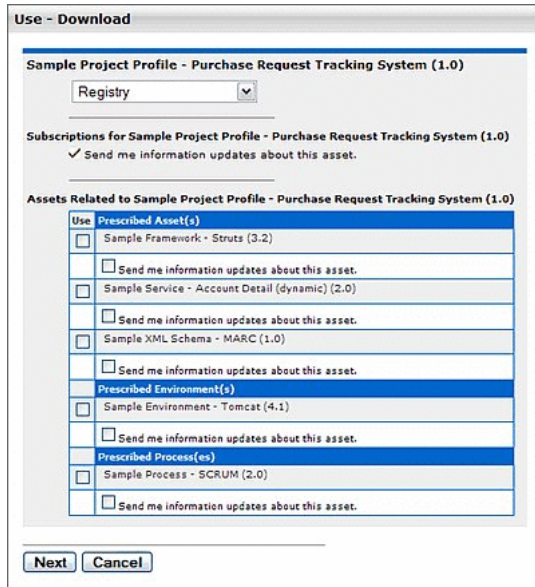
2. Any compliance templates applied to the project is listed in the Compliance Templates section of the project detail, as shown in [Figure 14–15](#).

**Figure 14–15 Compliance Templates**



3. Click **Use Assets** to the right of the name of the listed compliance template. The Use / Download dialog is displayed, as shown in [Figure 14–16](#).

**Figure 14–16 Use - Download Dialog**




---

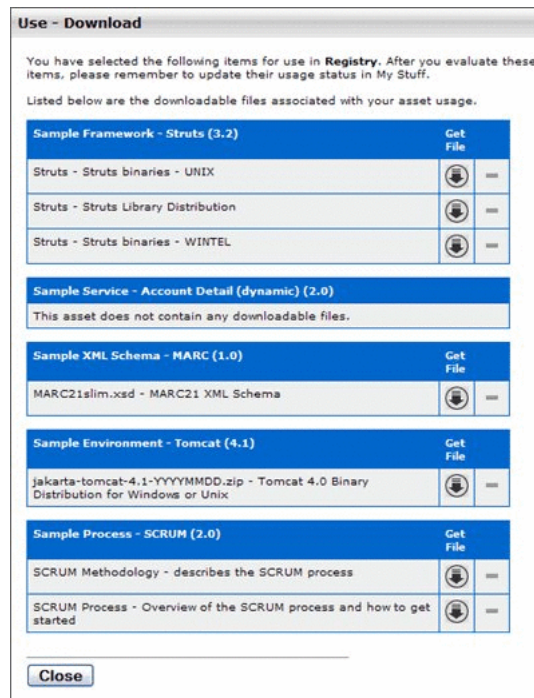
**Note:** The Use Assets button will appear only to users who are assigned to the project.

---

4. Use the options in the Use column to select the assets to be downloaded.
5. Click the **Next** button. The Use / Download dialog confirms the asset selections, as shown in [Figure 14–17](#).

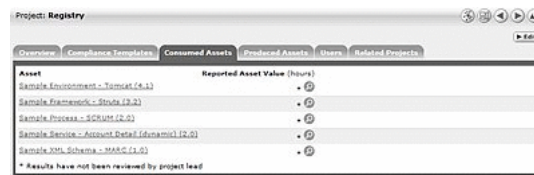


Figure 14–17 Use - Download Dialog



6. Click the down arrow button to download the payloads for the selected assets.
7. Click **Close** when finished. The downloaded assets are listed on the Consumed Assets section of the project detail, as shown in [Figure 14–18](#).

Figure 14–18 Consumed Assets Tab



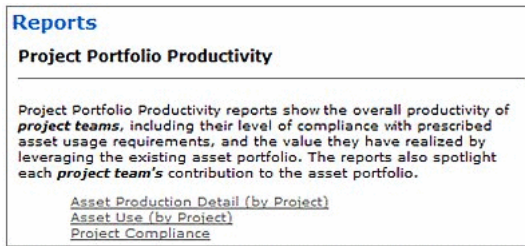
## 14.6 Running a Project Compliance Report

Project Compliance reports provide the means for verifying project adherence to the assets prescribed for use in compliance templates.

This procedure is performed on the Oracle Enterprise Repository Reports screen.

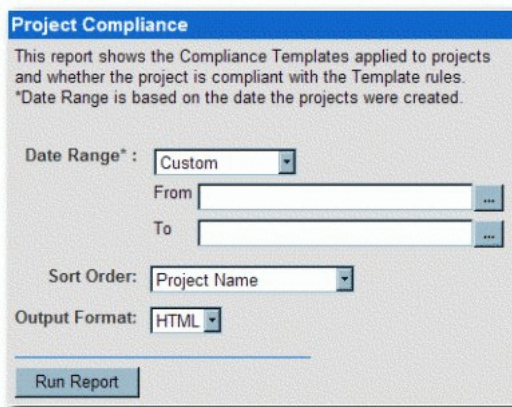
1. Click the **Project Compliance** link in the Reports sidebar, as shown in [Figure 14–19](#).

**Figure 14–19 Reports Section**



The Project Compliance report form opens in the main pane, as shown in Figure 14–20.

**Figure 14–20 Project Compliance Report Form**



2. Use the Date Range, Sort Order, and Output Format lists to set the report parameters.

---

**Note:** Date Range is based on the project creation date.

---

3. Click the **Run Report** button. The report is displayed in the selected format, listing each project, and indicating compliance templates in use, prescribed assets, usage status of prescribed assets, and any additional assets consumed by the project, as shown in Figure 14–21.



Figure 14-21 Project Compliance Report

Project Compliance		Use Date: 2006-10-01 - 2006-10-17
Project Name:	Example Project	Start Date:
Project Status:	Open	Closed Date:
Compliance Template in use: Sample Project Profile - Purchase Request Tracking System (1.0)		
REQUIRED ASSETS		
Prescribed Assets		
Another Sample Service (1.0)	2006-10-17	NO YES 2006-10-17
Sample Framework - EBUS (3.2)	2006-10-17	NO YES 2006-10-17
Sample Service - Account Detail (dynamic) (2.0)	2006-10-17	NO YES 2006-10-17
Sample XML Schema - MARC (1.0)	2006-10-17	NO YES 2006-10-17
Prescribed Environments		
Sample Environment - Tomcat (4.1)	2006-10-17	NO YES 2006-10-17
Prescribed Processes		
Sample Process - SCRUM (2.0)	2006-10-17	NO YES 2006-10-17
*Details Asset in Progress		Number of Required Assets: 6
		Number of Required Assets in use: 6
		Number of Required Assets not in use: 0
		Number of Additional Assets in use: 0
Number of Projects returned: 1		
©2006 BEA Systems, Inc. All rights reserved. Powered by BEA Application Enterprise Repository		



---

## Integration with Code Compliance Inspector

This chapter provides an overview and describes how to use the Code Compliance Inspector (CCI) with Oracle Enterprise Repository (OER).

This chapter includes the following sections:

- [Overview](#)
- [Checking Compliance and Synchronizing Report Data in OER](#)
- [Configuring Reports for Access from the OER Reports Menu](#)

### 15.1 Overview

Adherence to open standards and the enforcement of good coding practices are key principles of SOA governance. The Code Compliance Inspector is a tool that checks for good coding practices in SOA Suite projects.

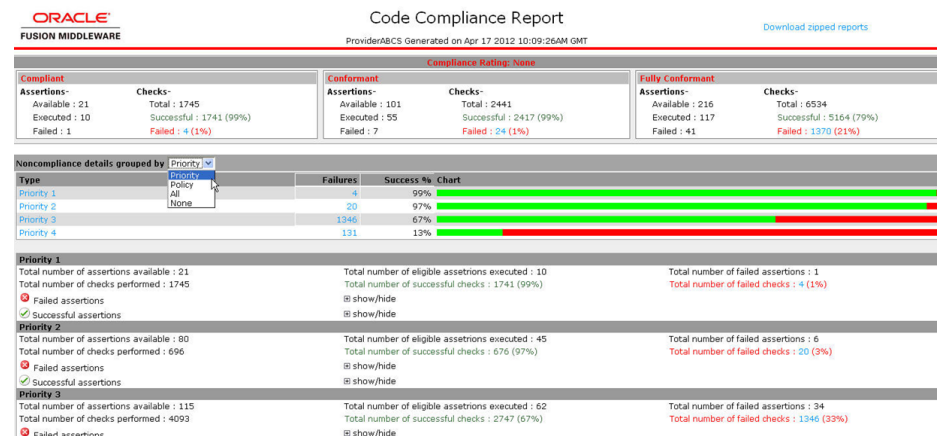
The CCI is delivered with a set of pre-defined assertions that are based on the Web Services Interoperability Organization Basic Profile (WS-I BP) to check for design consistency and good coding and documentation practices. The CCI qualifies code as Compliant, Conformant, or Fully Conformant in alignment with those open standards and best practices.

The results are displayed in the Code Compliance Report that shows the level of compliance, pass and failure percentages, and provides a graphical bar chart that groups results by priority and policies. The results also include the Top 10 violating composites. The overall compliance score for a SOA Suite project is shown in the header section of the report.

The Code Compliance Inspector can also be used to check for good coding practices in Oracle Application Integration Architecture (AIA) integration projects. In addition to assertions based on WS-I BP, the Code Compliance Inspector provides assertions based on AIA Integration developer guidelines to check for design consistency and coding practices. CCI compliance results and reports are also supported for AIA composites.

[Figure 15-1](#) shows an example of the overall Code Compliance Report. This report is linked through the OER Reports page. You can click through the composites to get more detail about composites. You can also see each composite individually using the Asset Detail page in OER. Additionally, you can post the report files to a server and share the compliance results with colleagues.

**Figure 15–1 Overall Code Compliance Report**



### 15.1.1 Overview of Steps for Generating CCI Reports and Publishing them to OER

This section provides an overview of the steps required to collect CCI reports that are ready to be associated in OER with harvested composites and published for view by OER users.

The CCI for OER consists of two command line utilities that are used by build administrators to generate reports that can be published in OER:

1. The checkCompliance command produces HTML compliance report files.
2. The cci-oesynch command synchronizes the results to the repository, enabling users to access the reports from the OER console.

#### Here are the overview steps

1. Configure an OER artifact store.
  - For more information, see [Section 15.2.1, "Configuring an OER Artifact Store for CCI Reports"](#).
2. Within in your build environment, configure the cci-oesynch.properties file.
  - For more information, see [Section 15.2.2, "Configuring the cci-oesynch.properties File"](#).
3. From your build environment, run the command line compliance script to generate CCI reports from the projects and prepare them for OER publication. You must run the command line tool to enable synchronization with OER.

For more information, see [Section 15.2.3, "Running the Check Compliance Command"](#).

4. Ensure that the SOA or AIA projects from which the reports are generated have been harvested into OER.

For more information about harvesting, see "Configuring and Using Automated Harvesting in Design-time and Runtime Environments" in the *Oracle Fusion Middleware Configuration Guide for Oracle Enterprise Repository*.

5. From your build environment that contains the CCI reports generated by the compliance check command, run the CCI-OER Synch command.

For more information, see [Section 15.2.5, "Running the CCI-OER Synch Command"](#).

6. Copy the newly generated CCI reports folder into the physical location associated with your OER artifact store. Note that if this location is hosted on the same system as the original reports folder, the CCI-OER Synch will automate this move. Otherwise, this is a manual step.

For more information, see [Section 15.2.6, "Copying the Files"](#).

For more information about artifact stores, see [Section 9.3.3, "Selecting a New Artifact Store"](#).

7. Within your OER installation, configure the Compliance Report properties file to enable access to the CCI reports from the OER Reports menu.

For more information, see [Section 15.3, "Configuring Reports for Access from the OER Reports Menu"](#).

## 15.2 Checking Compliance and Synchronizing Report Data in OER

The CCI command line tools are distributed with OER within the `<OER_Home>/modules/tools/solutions/12.1.3.0.0-ComplianceInspector.zip`.

Ensure that the contents of the zip file are installed on the machine where your build environment exists.

### Here are the set up steps:

1. Copy Compliance Inspector zip file to temporary folder.
2. Unzip the Compliance Inspector zip file.
3. Configure Compliance Inspector's `cci-oerSynch.properties` file located in: `ComplianceInspector/config/cci-oerSynch.properties`.

The properties file enables you to set static information about your environment.

---

**Note:** Wherever you unzip this file to, the property file is relative to the `ComplianceInspector` directory under `config`. You can rename this directory; we refer to this location as the `ComplianceInspector_Home`.

---

### 15.2.1 Configuring an OER Artifact Store for CCI Reports

1. Create and configure an artifact store(s) of the appropriate type (for example, HTTP) within OER.

For more information about using artifact stores, see [Section 9.3.2, "Configure Artifact Stores"](#).

2. Name the artifact store(s) appropriate for its purpose (for example, `CCI-Reports-Production`).

### 15.2.2 Configuring the `cci-oerSynch.properties` File

This command runs the Code Compliance Inspector on a specified SOA or AIA project folder and creates a new folder containing the CCI reports.

The `checkCompliance` command also uses the `cci-oerSynch.properties` file to retrieve the URL of the OER instance in order to look up both the composite Asset Type ID and the username with which you want to connect.

Use the `rep://<ARTIFACTSTORE-NAME>` as a part of the `compliance.report.web.root` property within the `cci-oerSynch.properties` file.

Example: `compliance.report.web.root=rep://CCI-Reports-Production/reports`

Reports will then have the following URL pattern:

```
rep://CCI-Reports-Production/reports/cci/AIADemo/reports/composites/AIADemoBatchJMSAdapter.html#hide_link
```

The second property: `compliance.report.web.root`, represents the base URI for the host that will contain the Compliance Report HTML files. The value supplied in this property is concatenated with the `inputDir` parameter when running the CCI-OER Synch tool. This concatenated value is what is embedded into the asset's metadata for the compliance report link.

In most cases, the value of the `compliance.report.web.root` property will point to the artifact store which was configured in [Section 15.2.1](#).

For example:

```
<a target="_blank" href="rep://CCI-REPORTS">rep://CCI-REPORTS</a>
```

### 15.2.3 Running the Check Compliance Command

Invoke CCI with the `checkCompliance.sh` on Linux or `checkCompliance.bat` command on Windows using the following switches:

- ***-inputDir*** {Absolute path to the folder that contains composite(s)}

This is a mandatory switch indicating where the input dir is located. If the `-inputMetaFile` switch is not specified, this input is not necessarily representative of a SOA Suite project. If the `-inputMetaFile` switch is provided, this specifies the project root directory (the source folder containing the project folder from `ComplianceInspector_Home`).
- ***-outputDir*** {Output folder where the compliance report will be generated}

This is a mandatory switch to indicate where the output reports will be stored. For example:

If your composites live here: `/tmp/cci/composites/AIADemo`

And you pass the output directory as: `/tmp`

Then CCI will put the produced files here: `/tmp/AIADemo`
- ***-policiesFile*** {Policies file name}

Use this optional switch to indicate which policies file CCI should run against, for example, `Policies-WS-I_11.x.xml`. The file is available under `ComplianceInspector/lib` or `ComplianceInspector/config` (Tool class path) or embedded in the `compliance.policy.engine.jar`.
- ***-policy*** {Policy name}

This is an optional switch to specify the policy to execute. If not given, the default policy from `Policies.xml` will be executed.
- ***-assertion*** {Assertion name}

Use this optional switch to indicate which assertion CCI should run against. This will run the tool for a specific assertion that you have defined, for example, `ABCSTargetNameSpacesCheck`.
- ***-inputMetaFile*** {absolute path to a SOA Suite project metafile}

Use this optional switch if you want to run reports for a specific SOA Suite project. The input metafile contains paths pointing to the specific directories that Code

Compliance Inspector needs to scan so that the output results are specific for the SOA Suite project. This file contains the names of all of the services that are used in a given SOA Suite project. When this option is specified, the `-inputDir` switch will point to the project root directory since the input metafile contains the directory path relative to this root. Here are some examples:

```
-inputMetaFile <dir path of the file>/GenerateScriptInput.xml
```

```
-inputMetaFile <dir path of the file>/MyPIPDP.xml
```

- ***-inputMetaFile ALL***

Use this optional switch if you want to run reports for all of your SOA Suite projects. When this option is specified, the `-inputDir` switch will point to the project root directory since the input metafile contains the directory path relative to this root.

- ***-version***

The `-version` flag tells you which version of CCI (CCI build date and time) you are using. This is an optional argument that displays the version information.

### Examples

Here are some examples for invoking the Code Compliance Inspector from a command line:

- Windows: `checkCompliance.bat -inputDir D:\composites\demo -outputDir D:\ComplianceOut`
- Linux: `checkCompliance.sh -inputDir /composites/demo -outputDir /ComplianceOut`

## 15.2.4 Harvesting the SOA or AIA Projects into OER

For more information about harvesting, see [Chapter 8, "Configuring and Using Automated Harvesting"](#).

## 15.2.5 Running the CCI-OER Synch Command

The CCI-OER Synch updates composite assets in OER with links to the associated CCI reports. You must have already configured an OER artifact store, or otherwise have determined the location from which OER will publish the reports, before you run the CCI-OER Synch command.

If you are running the CCI-OER Synch command on the system from which OER will publish the reports, then you can specify the folder location in the `cci-oesynch.properties` file using the `oer.app.report.location` property and the command will automatically move the reports from the input folder to the specified location.

If the destination OER report location is on a different system, then you must manually move the folder specified by the `-reportLocation` argument to the OER report location and ensure that the pathname to this location matches what is specified in the `compliance.report.web.root` property.

Examples:

For the artifact store:

```
compliance.report.web.root=rep\://INSTANCE/reports
```

For a direct HTTP resource:

compliance.report.web.root=http://www.example.com/CCI-REPORTS-Production

### Running CCI-OER Synch

1. Update the `cci-oerSynch.properties` file, located in the `ComplianceInspector/config` directory, with the machine details where you have OER installed.
2. Run `cci-oerSynch.bat` or `cci-oerSynch.sh`, located in the `ComplianceInspector/bin` directory. [Table 15–1](#) provides a list of the command line arguments you can pass:

**Table 15–1 Command Line Arguments for OER Synch**

Argument	Type	Description
<code>-url &lt;url to oer web service&gt;</code>	Optional	URL to OER web service. For example: <code>http://example.com:port/oer/services</code>
<code>-user &lt;oer user name&gt;</code>	Optional	Valid OER user login. This user must have privileges to update asset instances.
<code>-pwd &lt;oer password for user name specified&gt;</code>	Optional	Password for the OER user login.
<code>-reportLocation &lt;path to output directory generated by CCI&gt;</code>	Required	Location of the CCI HTML compliance reports, created by the compliance inspection tool, which you want to integrate into OER.

---

**Note:** If the optional command line arguments are not given, then the values will be taken from the `cci-oerSynch.properties` file. Command line arguments take precedence over the properties specified in the `cci-oerSynch.properties` file. Also, if the password is not given in the command line argument, then you will be prompted to enter it.

---

## 15.2.6 Copying the Files

1. Copy the newly generated CCI reports folder into the physical location associated with your OER artifact store.

Note that if this location is hosted on the same system as the original reports folder, the CCI-OER Synch will automate this move. Otherwise, this is a manual step.

2. To verify that this worked, check that the link was created on the composite asset in OER.



## 15.3 Configuring Reports for Access from the OER Reports Menu

After the CCI-OER Synch has linked CCI reports to composite assets in OER, a link to the report will appear on the composite detail page. CCI reports can be accessed from the OER Reports menu after the following configuration steps have been performed.

---

**Note:** The name attribute in the report elements of the ComplianceReport.xml file are referenced within the `<OER_Home>/applications/oes/oes-app/WEB-INF/config/reports/custom.toc` file. If these elements do not match, then the values of the `displayName` and `description` elements will not appear in the OER reports page.

The `<OER_Home>/applications/oes/oes-app/WEB-INF/config/reports/custom.toc` file enables you to add URL patterns to a reporting system that you may have integrated with that is external from OER. The `custom.toc` file indicates which other XML files it will access to gather the 'custom' report grouping information from and display the corresponding information referred to in the `custom.toc` file.

---

### To add and configure the compliance reports

1. Shut down the application server.

2. Edit the `<OER_Home>`

`Home>/applications/oes/oes-app/WEB-INF/config/reports/ComplianceReport.xml` file.

Here is an out-of-the-box example of `ComplianceReport.xml`:

```
<?xml version="1.0" ?>
<reports>
  <report name="complianceReportDoc">
    <displayName>Compliance Report Documentation</displayName>
    <description>Documentation for setting up Compliance Reports.
    </description>
  </report>
  <report name="complianceReport">
    <displayName>Compliance Report</displayName>
    <description>This is the index page for all technical compliance
reports.</description>
    <!-- update the host and port, according the your http enabled server
where you have hosted the compliance reports. Use the OER download servlet to
map the OER artifact store URL for browser consumption. Example:
<externalLink>http://server.example.com:7101/oes/com.flashline.cmee.
servlet.enterprisetab.Download?path=rep://INSTANCE/reports/cci/FPDemo/
index.html</externalLink> -->
    <externalLink>http://host.example.com/CCI/reports/index.html</externalLink>
  </report>
</reports>
```

3. If there is only one SOA Suite project being summarized, then modify the existing report named `complianceReport` and go to step 9.
  - If you are using an OER artifact store to reference the location of your reports (which is recommended) then you should use the following URL pattern as a part of the `externalLink` element in the `CustomReport.xml` file:

```
http://host.example.com:port/oer/com.flashline.cmee.servlet.enterpr
isetab.Download?path=rep://<ARTIFACTSTORENAME>/relative_
path/index.html
```

The Download servlet is required to decode and reformat the URL to give the user the correct URL that the artifact store represents. The use of the OER artifact store allows OER administrators to alter the hostname/port and path for the report server(s) without being forced to re-synch or modify multiple assets to update these details.

4. If more than one SOA Suite project will be summarized, then duplicate the report section a number of times equal to the number of projects to be reported by the Check Compliance tool.
5. Update the name attribute of each report element to a meaningful name (each must be unique).
6. Update the displayName, description and external link elements as appropriate to the SOA Suite project's report summary pages and save the edits.
7. Edit the `<OER_`  
`Home>/applications/oer/oer-app/WEB-INF/config/reports/custom.toc` file

Example:

```
<?xml version="1.0" ?>
<reportSections>
  <reportSection name="Custom">
    <summary></summary>
    <description>Custom reports provide you with the flexibility
to customize reports for your organization or build new reports that align
your organizational metrics with your program goals.</description>
  </reportSection>
  <reportSection name="Compliance Reports">
    <summary></summary>
    <description>Code Compliance Inspector report for design-time
compliance information</description>
    <report name="complianceReportDoc"/>
    <report name="complianceReport"/>
  </reportSection>
</reportSections>
```

8. Modify the report name elements to match those added or modified in the `CustomReport.xml` file.
9. Restart the OER application server for the changes to take effect, as shown in [Figure 15-2](#).

Figure 15–2 OER Compliance Report Links

The screenshot shows the Oracle Enterprise Repository interface. The 'Reports' menu is expanded, and the 'Compliance Reports' section is highlighted with a red circle. This section lists several reports, including 'Code Compliance Inspector report for design-time compliance information', 'Compliance Report (Project 1)', and 'Compliance Report (Project 2)'. The 'Compliance Report (Project 1)' and 'Compliance Report (Project 2)' links are also circled in red. The footer of the page indicates the user is logged in as Administrator2.

Here is a sample of a `ComplianceReport.xml` file that is using the OER artifact store to lead to the compliance report summary page:

```
<?xml version="1.0" ?>
<reports>
  <report name="complianceReport_project1">
    <displayName>Compliance Report (Project 1)</displayName>
    <description>This is the compliance report summary page for Composite
Project 1.</description>
    <!-- update the host and port, according the your http enabled server
where you have hosted the compliance reports -->
    <externalLink>http://host.example.com/CCI/reports/project1/index.html
</externalLink>
  </report>
  <report name="complianceReport_project2">
    <displayName>Compliance Report (Project 2)</displayName>
    <description>This is the compliance report summary page for Composite
Project 2.</description>
    <!-- update the host and port, according the your http enabled server
where you have hosted the compliance reports -->
    <externalLink>http://example.com:8080/tr130107/com.flashline.cmeec
.servlet.enterprisetab.Download?path=rep://INSTANCE/reports/cci/FPDemo/
index.html</externalLink>
  </report>
</reports>
```

The 'real' URL path to this location is:

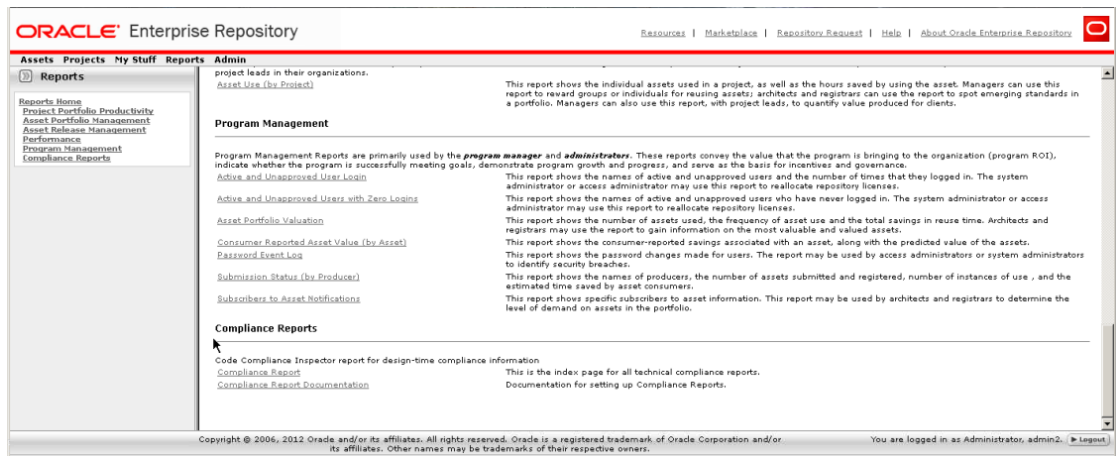
```
http://example.com:8080/tr130107/reports/cci/FPDemo/index.html
```

The italicized portion of the URL above is the relative path to the report files which is also the relative path after the repository name.

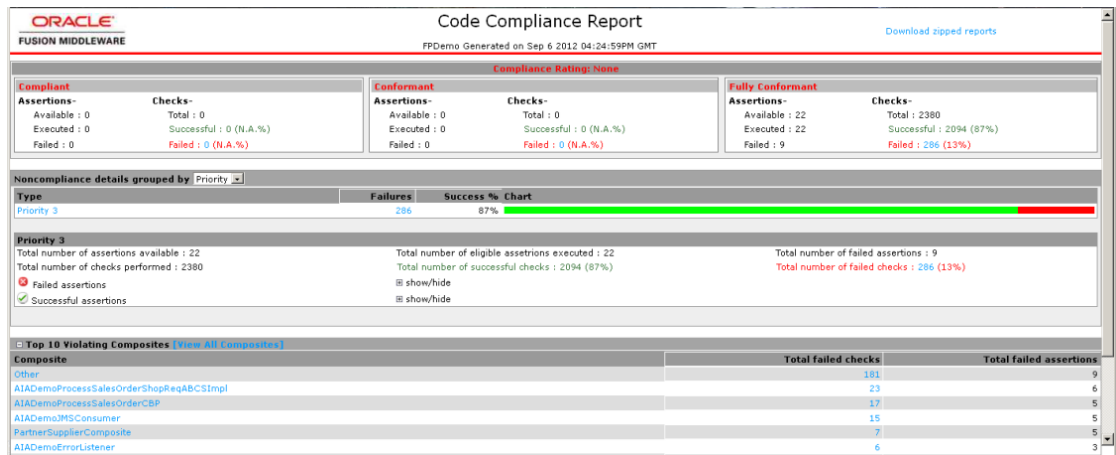
### 15.3.1 Reviewing the Reports

In OER, click the Compliance Report link shown in [Figure 15-3](#) to see the overall report for all composites, shown in [Figure 15-4](#)

**Figure 15-3 Compliance Report Link**



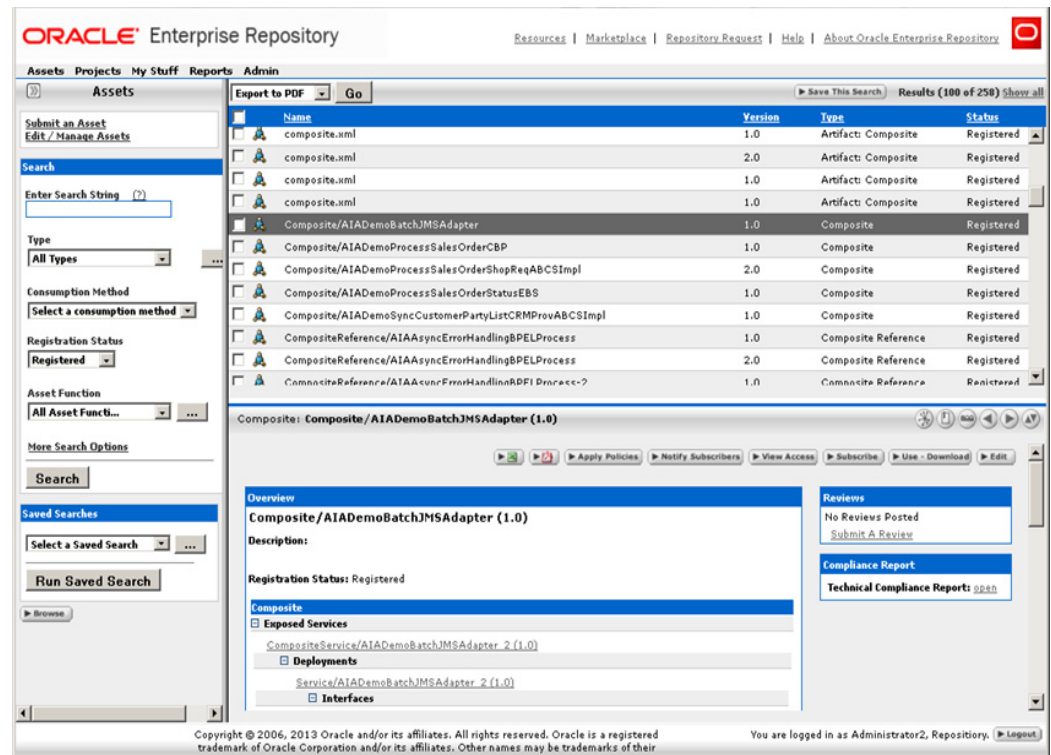
**Figure 15-4 Overall Code Compliance Report**



Click a violating composite, shown in [Figure 15-4](#) to see more detail about the violations and then fix the compliance problem.

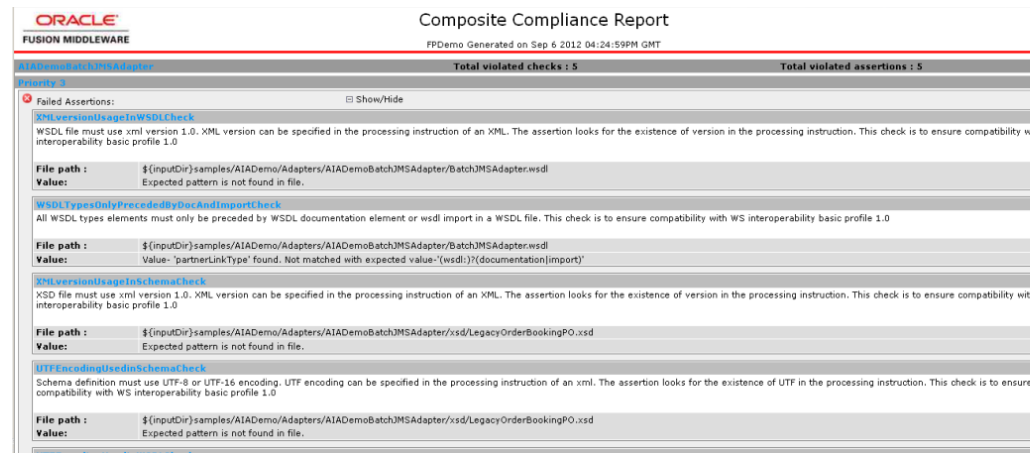
You can also view compliance information on individual composites through OER, as in [Figure 15-5](#).

Figure 15–5 Technical Compliance Report



Click the **open** link in the Compliance Report box to access detailed results, as shown in Figure 15–6.

Figure 15–6 Composite Compliance Report





---

---

## Configuring Policy Management

This chapter describes various aspects of policy management and how to use policies in Oracle Enterprise Repository.

This chapter contains the following sections:

- [Section 16.1, "Overview"](#)
- [Section 16.2, "Activating the Policy Management Feature"](#)
- [Section 16.3, "Creating a Policy Type"](#)
- [Section 16.4, "Adding the Policy Assertion Element to the Policy Type"](#)
- [Section 16.5, "Creating a Policy"](#)
- [Section 16.6, "Adding the Applied Asset Policies Element to a Type"](#)
- [Section 16.7, "Applying a Policy to an Asset"](#)
- [Section 16.8, "Viewing Policy Status"](#)
- [Section 16.9, "Validating Policy Assertions"](#)
- [Section 16.10, "Running the Oracle Enterprise Repository Policy Reports"](#)

### 16.1 Overview

Policies are applied to assets in order to communicate asset requirements that need to be considered during design and development, and to provide administrators with the means to enforce and monitor asset compliance with governance, architecture, and other organizational standards. For example, a policy might articulate corporate quality standards, identifying the platforms that an asset should run on, and identifying acceptable defect density rates.

A policy can be applied to multiple assets. Multiple policies can be applied to any asset. Each policy consists of at least one Assertion Statement:

- Each assertion has a name and description, and includes a Technical Definition. The Technical Definition accommodates additional metadata that may be required to automatically validate the assertion using third party testing and validation tools. This metadata may be Web service-specific policy information, XML, or any other format that can be read by an external system.
- For example, an assertion statement for Defect Density might state that defect density must be less than .1%

Sample policies are included in the Oracle Enterprise Repository Base datapack and are imported into your Oracle Enterprise Repository instance. For more information

about Oracle Enterprise Repository Import/Export capabilities, see [Chapter 5, "Import Export Tool"](#).

## 16.2 Activating the Policy Management Feature

To activate the policy management feature, perform the following steps:

### Step 1: Enable the Policy Management System Settings

This procedure is performed on the Oracle Enterprise Repository Admin screen.

1. Click **System Settings** in the left pane.
2. Navigate to the Policy Management section, as shown in [Figure 16–1](#).

**Figure 16–1 Policy Management**

Policy Management	
<b>Enable Compliance Templates</b> cmee.compliance-template.enable	<input checked="" type="radio"/> True <input type="radio"/> False Enables the Compliance Template System
<b>Enable Asset Policies</b> cmee.asset-policy.enable	<input checked="" type="radio"/> True <input type="radio"/> False Enables the Policy Management System
<b>Collapse Policy Assertions</b> cmee.asset-policy.collapse.assertions	<input type="radio"/> True <input checked="" type="radio"/> False Collapse the display of assertions by default when showing Asset detail.
<b>Asset Policy Relationship</b> cmee.asset-policy.relationship	Policy Applies to Defines the relationship type within the system to use for p... should be a Two-Way, Order Matters relationship.

3. Ensure the **Enable Asset Policies** property is set to True.
4. Use the options to make the appropriate selection for the Collapse Policy Assertions property.
  - True sets the default Asset Detail display of assertions to collapsed.
  - False sets the default Asset Detail display of assertions to expanded.
5. Select **Policy Applies To** in the Asset Policy Relationship list.
6. When finished, click the **Save** button. Policy management features are now activated.

---

**Note:** The settings are VISIBLE and OFF, by default, on install. The user must set the property to the correct value as described in the documentation to turn ON the feature.

---

### Step 2: Import the Policy Management Solution Pack

After configuring the Policy Management system settings, the sample policy types and policy examples can be obtained by installing the Policy-Management-Solution-Pack.zip from <ORACLE\_HOME>/oer/modules/tools/solutions, and then using the Import/Export Tool to import them into your environment.

---

**Note:** You must import the Policy Management Solution Pack after enabling the system settings in "[Step 1: Enable the Policy Management System Settings](#)". Importing the Solution pack before enabling these settings may cause problems.

---



## 16.3 Creating a Policy Type

If you prefer to create your own policy instead of using the policy template in the Base Data, you can do so, which is explained in this section. A policy type template can be created using any existing asset type or compliance template type as the basis. A Policy Type is included in the Policy Management Solution Pack.

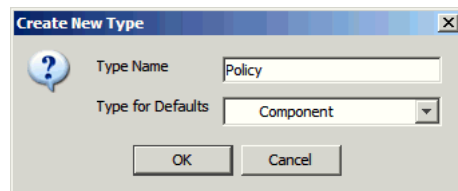
1. Click **Edit/Manage Assets** in the Oracle Enterprise Repository Assets screen to launch the Asset Editor.

**Figure 16–2 Assets Section**



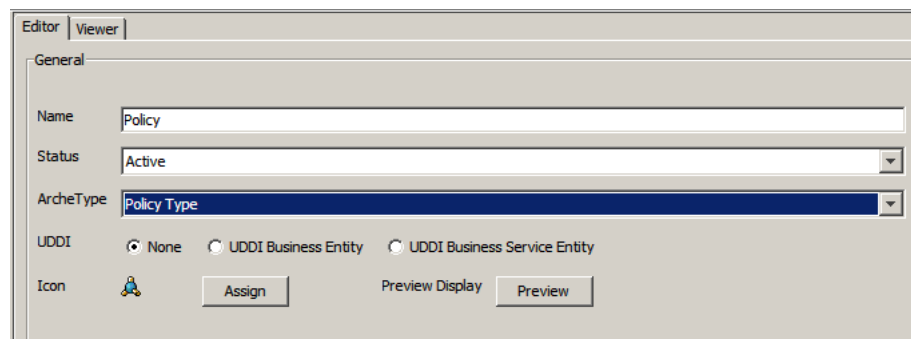
2. Open the Actions menu in the Asset Editor.
3. Click **Manage Types** to launch the Type Manager.
4. Open the File menu and click **New**. The Create New Type dialog is displayed.
5. Enter **Policy** in the Type Name field and select a template from the Type for Defaults list, as shown in [Figure 16–3](#).

**Figure 16–3 Create New Type Dialog**



6. Click **OK**.
7. Select **Policy Type** from the Archetype list in the Editor tab in the Type Manager.

**Figure 16–4 Type Manager**



You have successfully created your own policy type.

## 16.4 Adding the Policy Assertion Element to the Policy Type

A policy assertion states that conditions must be true for an asset. For example, a policy assertion might state that the asset should have a "defect density rate of less than

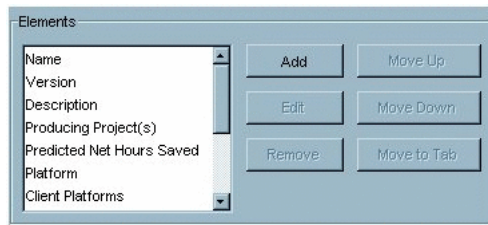
1%". Each assertion is validated individually. If the asset satisfies all assertions (meaning that all assertions have a "pass" status in Oracle Enterprise Repository), then the asset is compliant with the applied policy. In order for the new Policy template to function properly, the Policy Assertions element must be added. This simple procedure is performed in the Type Manager.

The Policy Assertions element may be added to any tab in the Policy template. The following instructions will add the Policy Assertions element to the Overview tab.

1. Select **Overview** in the Tabs section of the Type Manager.

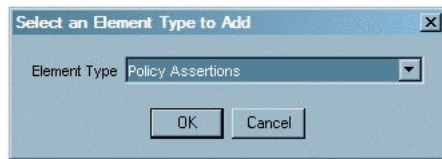
Any elements already assigned to the Overview tab is listed in the Elements section of the Type Manager, as shown in [Figure 16-5](#).

**Figure 16-5 Elements Section**



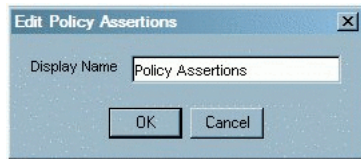
2. Click **Add**. The Select an Element Type to Add dialog is displayed.
3. Select **Policy Assertions** from the Element Type list, as shown in [Figure 16-6](#).

**Figure 16-6 Select an Element Type to Add**



4. Click **OK**. The Edit Policy Assertions dialog is displayed, as shown in [Figure 16-7](#).
5. Enter a name for the Policy Assertion in the Display Name text box. (The example below uses "Policy Assertions" as the name for the element.)

**Figure 16-7 Edit Policy Assertions**



6. Click **OK**. The Policy Assertions element now appears in the Elements list.

---

**Note:** Any asset type which contains Assets associated with Policies, requires the Policy Assertion element.

---

## 16.5 Creating a Policy

A Policy is a very specific type of asset, with a unique relationship to other assets. Policies are created using a Policy Type.

This procedure is performed in the Asset Editor.

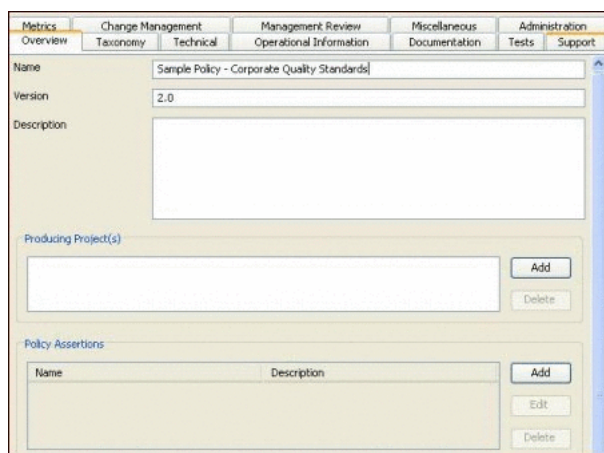
1. Open the File menu and click **New**. The Create a New Asset dialog is displayed.
2. Enter a name for the policy in the Name text box, as shown in [Figure 16–8](#).

**Figure 16–8 Create a New Asset Dialog**



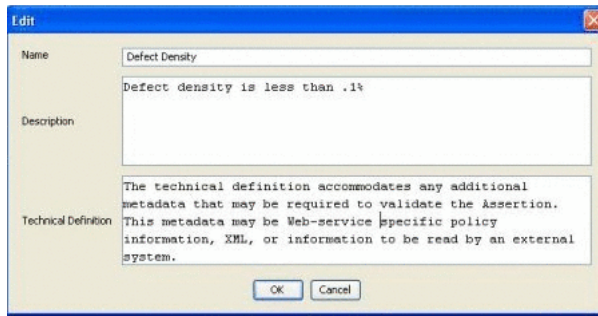
3. Enter a version number (optional).
4. Select the appropriate policy type from the Type pull-down.
5. Select the appropriate Initial State from that pull-down.
6. Click **OK**. The policy opens in the Asset Editor.

**Figure 16–9 Overview Tab**



7. Click **Add** in the Policy Assertions section. The Edit dialog is displayed.
8. Fill in the Name, Description, and Technical Definition sections as appropriate.

**Figure 16–10 Edit Dialog**



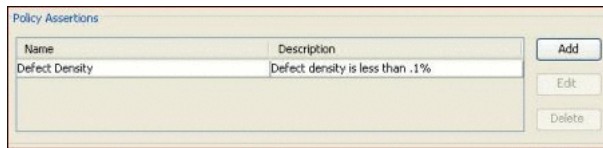

---

**Note:** Name and Description are human-readable information; Technical Definition should be machine-readable (XML or similar) information that allows an assertion to be passed to other processes and tooling for validation.

---

9. Click **OK**. The new policy assertion is displayed in the Policy Assertions section.

**Figure 16–11 Policy Assertions**



10. Repeat the procedure to add additional assertions to the policy.

## 16.6 Adding the Applied Asset Policies Element to a Type

Policies can only be applied to assets of types to which the Applied Asset Policies element has been added. For example, if a Corporate Quality Policy is applicable to all services, then the Applied Asset Policies element must be added to the Service asset type to allow the policy to be applied.

The procedure for adding the Applied Asset Policies element is performed in the Type Manager and is nearly identical to the Adding the Policy Assertion Element to the Policy Type procedure.

---

**Note:** Prior to enabling policies in Oracle Enterprise Repository, perform the following steps in the order mentioned:

1. Set the Enable Policies system setting to true, as you had already performed in "[Step 1: Enable the Policy Management System Settings](#)"
  2. Import the harvester solution pack. For more information about importing Harvester Solution Pack, see *Oracle Fusion Middleware Installation Guide for Oracle Enterprise Repository*.
  3. Import the Policy Management pack and fix pack, as you had already performed in "[Step 2: Import the Policy Management Solution Pack](#)".
- 

1. Open the selected asset type. In the right panel, select the **Editor** Tab.

2. In the Tabs section, select the tab to which the Applied Asset Policies element is added.
3. In the Elements section, click **Add**. The Select an Element Type to Add dialog is displayed.
4. Select the Applied Asset Policies option from the Element Type list.
5. Click **OK**. The Edit Applied Asset Policies dialog is displayed.
6. Enter an appropriate name in the Display Name text box.
7. Click **OK**. The Applied Asset Policies element appears bearing its assigned name in the list in the Elements section.
8. Change the right panel display to the Viewer Tab.
9. In the Hidden Elements section, select the **Applied Asset Policies** element. The Move Element dialog is displayed.
10. Select an appropriate Applied Asset Policies element from the Move Applied Asset Policies to list.
11. Save the Asset Type.

## 16.7 Applying a Policy to an Asset

---

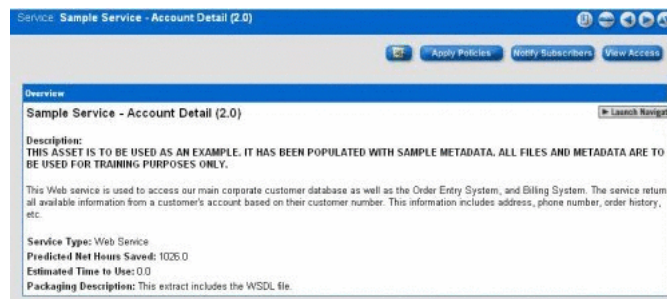
**Note:** User must be assigned the Apply Policy permission in the Policy Basic Access Setting.

---

The procedure for adding a policy to an asset is performed in the asset detail display.

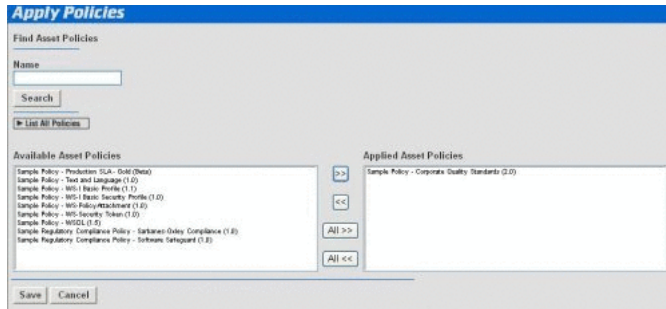
1. Using Oracle Enterprise Repository's Search or other means, locate the asset to which the policy is to be added.
2. Click the **Apply Policies** button in the asset detail display, as shown in [Figure 16–12](#).

**Figure 16–12 Overview**



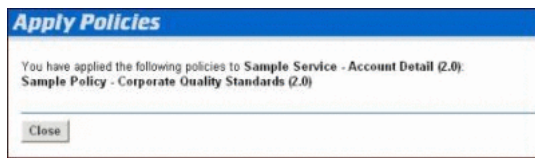
3. Click **List All Policies** to display a list in the Available Asset Policies box, as shown in [Figure 16–13](#).

**Figure 16–13 Apply Policies**



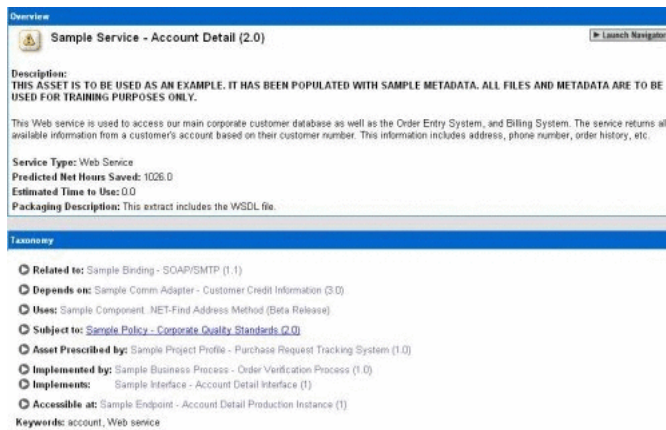
4. Use the arrow buttons to move selected policies to the Applied Asset Policies box. In the example above the Sample Policy - Corporate Quality Standards policy has been applied.
5. Click **Save**.
6. A confirmation message is displayed, as shown in [Figure 16–14](#).

**Figure 16–14 Apply Policies**



7. Click **Close**.
8. Refresh the asset detail frame.
9. The presence of the icon in the Overview section of the asset detail, as shown in [Figure 16–15](#), indicates that a policy has been applied. (A smaller version of the same icon indicates policy assignment for assets listed in search results.)

**Figure 16–15 Overview Sample Policy - Corporate Quality Standards**



Note that Sample Policy - Corporate Quality Standards is identified as an applied policy in the Relationships section of the asset detail, as shown in [Figure 16–16](#). Click the text link for any listed policy to display its details.

**Figure 16–16 Sample Policy - Corporate Quality Standards**

The Relationships section of the policy detail lists the assets to which Sample Policy - Corporate Quality Standards has been applied. Click the link for Sample Service - Account Detail (2.0) to return to the detail display for that asset.

---

**Note:** Multiple policies may be applied to any eligible asset.




---

## 16.8 Viewing Policy Status

Policy status is indicated by the specific icon that appears in the asset detail display and in search results. (Policy status is also indicated in the optional Navigator.)

Policy status is determined by the asset's compliance with assertion statements within applied policies. If multiple policies are applied, the asset achieves passed status only when in compliance with each applied policy.

Similarly, since a policy can consist of multiple assertion statements, an asset achieves passed status only when in compliance with all assertion statements within all applied policies.

Image	Description
	<p>Status: Applied</p> <p>Indicates that one or more policies have been applied to the asset, and that pass/fail status is indeterminate.</p> <p>Any combination of passed and indeterminate assertion statements results in indeterminate asset status.</p>
	<p>Status: Passed</p> <p>Indicates that the asset is in compliance with every assertion statement within every applied policy.</p>
	<p>Status: Failed</p> <p>Indicates that the asset is not in compliance with all assertion statements within all applied policies.</p> <p>Failure to comply with any assertion statement within any applied policy results in Failed status.</p>

## 16.9 Validating Policy Assertions

Policy assertions can be validated manually through Oracle Enterprise Repository or automatically using testing and validation tools. This document will cover the manual



validation of policy assertions. For automatic validation processes, refer to the documentation for your specific testing and validation tools.

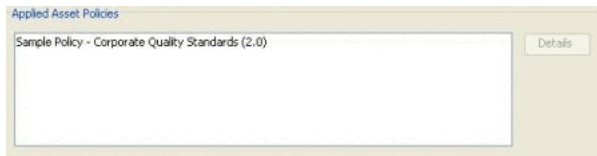
Manual validation of policy assertions is performed in the Asset Editor. See [Chapter 9, "Using the Asset Editor"](#) for more information about the Asset Editor.

1. Locate the asset to be updated.
2. Locate the Applied Asset Policies section within the asset.

The specific label and location of this section is determined by the configuration of the Policy element within the Type template.

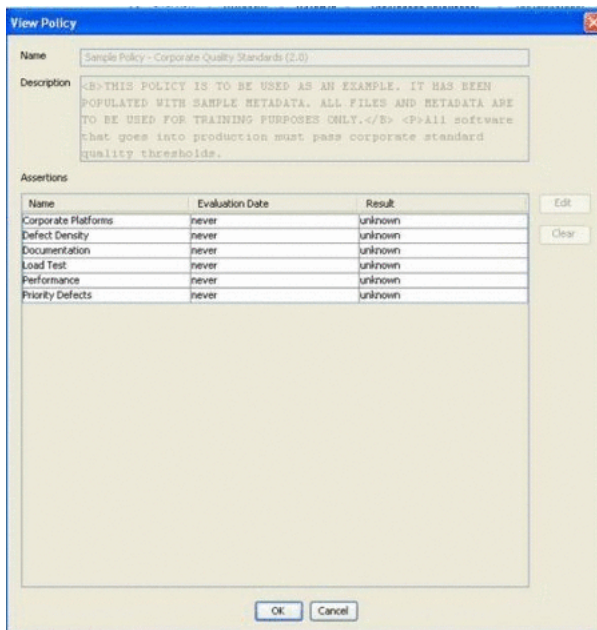
3. Select the policy to be updated in [Figure 16-17](#).

**Figure 16-17 Applied Asset Policies**



4. Click **Details**. The View Policy dialog is displayed, as shown in [Figure 16-18](#).

**Figure 16-18 View Policy Dialog**



The asset's compliance status with any policy listed in the View Policy section is indicated in the Result column. In the example above the status unknown indicates that the policy has been applied to the asset but compliance with that policy's assertion statement(s) has not been evaluated.

5. Select the assertion statement to be updated.
6. Click **Edit**. The Edit Assertion Result dialog is displayed, as shown in [Figure 16-19](#).



**Figure 16–19 Edit Assertion Result Dialog**

- Click the Evaluation Value list and select the appropriate status, as shown in [Figure 16–20](#).

**Figure 16–20 Evaluation Value List**

In this example the pass status is selected. Note that additional details can be provided in Evaluation Information.

- Click **OK**. The updated status is displayed in the Result column in the View Policy dialog, as shown in [Figure 16–21](#).

**Figure 16–21 View Policy Dialog**

Name	Evaluation Date	Result
Corporate Platforms	never	unknown
Defect Density	2007-06-28	pass
Documentation	never	unknown
Load Test	never	unknown
Performance	never	unknown
Priority Defects	never	unknown

**Note:** To change an assertion value back to unknown, select the assertion and click **Clear**.

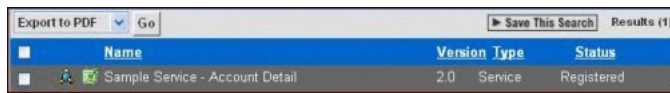
9. Continue to validate remaining assertions in the same manner.
10. When finished, click **OK** to close the View Policy dialog.
11. Save the changes to the asset.
12. An icon with a green tick mark appears in the asset detail, as shown in [Figure 16–22](#).

**Figure 16–22 Sample Service - Account Detail (2.0)**



It also appears next to the asset's name in search results, indicating policy compliance, as shown in [Figure 16–23](#).

**Figure 16–23 Search Results**



## 16.10 Running the Oracle Enterprise Repository Policy Reports

There are two different kinds of policy reports described in this section:

- ["Asset Portfolio Policy Status"](#)
- ["Policy Status Report"](#)

### Asset Portfolio Policy Status

This report provides a summary of the policy status (pass, fail, unknown) for an organization's entire asset portfolio. The report then allows you to view a list of assets for each status, for example, all of the assets that are failing. The detailed reports list all policies applied to each asset and the status of all policy assertions.

### Policy Status Report

This report provides a summary of all policies in an organization's policy portfolio, and the number of assets that are passing, failing, or have an unknown status. This report also shows a detailed list of the assets that are governed by each policy, and the number of assertions that are passing, failing, or have an unknown status.

To run the Oracle Enterprise Repository reports:

1. In Oracle Enterprise Repository Reports page, click a report name. The BI Publisher opens the Reports tab in a new browser window, as shown in

Figure 16–24.

**Figure 16–24 Reports Tab**

The screenshot shows the 'Reports' tab in a web application. The breadcrumb navigation is 'Home > Shared Folders > OER > Asset Use (By Project)'. There are links for 'View', 'History', 'Edit', and 'Configure'. The 'Start Date' is set to '2009-01-01' and the 'End Date' is '2009-12-31'. The 'Project Status' is set to 'Open'. The 'Sort Order' is 'Project Name'. Below these are 'Template' dropdowns set to 'Default' and 'HTML', and buttons for 'View', 'Export', and 'Send'. A 'Link to this report' link is also present.

2. Click the **View** button. The details of the selected report are displayed, as shown in Figure 16–25.

**Figure 16–25 Reports Tab - Detailed View**

The screenshot shows the 'Reports' tab in detailed view. The breadcrumb navigation is 'Home > Shared Folders > OER > Asset Use (By Project)'. There are links for 'View', 'History', 'Edit', and 'Configure'. The 'Start Date' is '2009-01-01' and the 'End Date' is '2009-12-31'. The 'Project Status' is 'Open'. The 'Sort Order' is 'Project Name'. Below these are 'Template' dropdowns set to 'Default' and 'HTML', and buttons for 'View', 'Export', and 'Send'. A 'Link to this report' link is also present.

**Asset Use By Project**

Use Date: 2009-01-01 to 2009-12-31  
Sort Order: Project Name

Project Name	Project ID	Department Name	Project Status	Total Project Hours
Dev Project	50302	Engineering Department	OPEN	0

Asset Name	Consumer	Use Date	Use Status	PNHS	Consumer Reported Value	Consumer Weighted Value	Usage Type
------------	----------	----------	------------	------	-------------------------	-------------------------	------------

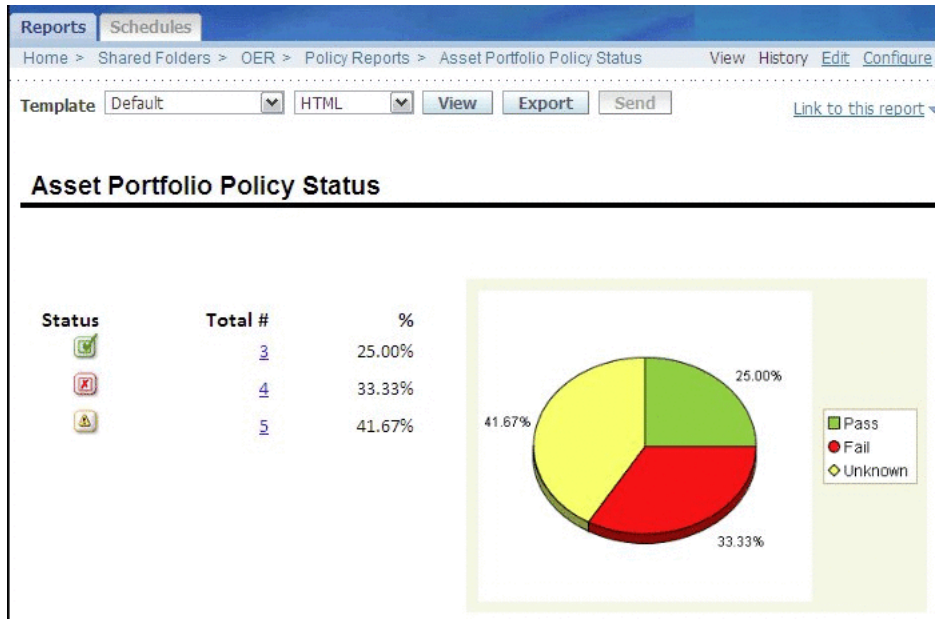
3. You can select different formats, as shown in Figure 16–26, from the second list that appears next to **Template**.

**Figure 16–26 List of Formats**

The screenshot shows the 'Reports' tab in a web application. The breadcrumb navigation is 'Home > Shared Folders > OER > Asset Use (By Project)'. There are links for 'View', 'History', 'Edit', and 'Configure'. The 'Start Date' is '2009-01-01' and the 'Sort Order' is 'Project Name'. Below these are 'Template' dropdowns. The first dropdown is set to 'Default' and the second dropdown is open, showing a list of formats: HTML, PDF, RTF, Excel, Excel2000, PowerPoint, MHTML, CSV, and Data. Buttons for 'View', 'Export', and 'Send' are also visible.

4. In the Asset Portfolio Management section, click the Asset Portfolio Policy Status option. The Asset Portfolio Policy Status page is displayed, as shown in Figure 16–27.

**Figure 16–27 Asset Portfolio Policy Status Page**



- To view the detail for a set of assets, click the number next to the icon for the appropriate Policy Status. The Asset Portfolio Policy Status - Status detail report is displayed, as shown in [Figure 16–28](#).

**Figure 16–28 Asset Portfolio Policy Status Report Page**

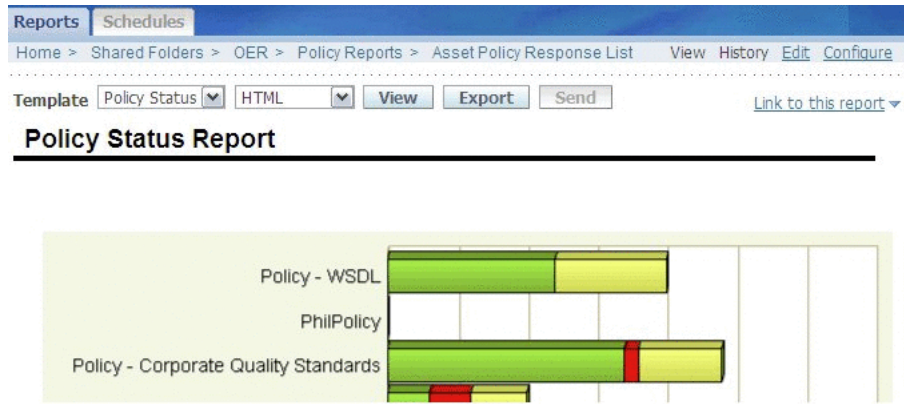
**Asset Portfolio Policy Status - Unknown**

[account.xsd](#)

Policy Name	Total Assertions	# Passed Assertions	# Failed Assertions	# Unknown Assertions
Policy - WS-I Basic Profile	2	0	0	2
Policy - WS-PolicyAttachment	1	0	0	1
Policy - WS-Security Token	1	0	0	1
Policy - WS-...	1	0	0	1

- In the Asset Portfolio Management section, click the Policy Status option. The Policy Status Report page is displayed, as shown in [Figure 16–29](#).

Figure 16–29 Policy Status Report Page





---

---

## Enterprise Manager Integration Utility

This chapter describes how to use and configure the Enterprise Manager Integration Utility to integrate Oracle Enterprise Manager with Oracle Enterprise Repository.

This chapter includes the following sections:

- [Overview](#)
- [Using the Enterprise Manager Integration Utility](#)
- [Configuring the Enterprise Manager Integration Utility](#)
- [Encrypting the Configuration File Passwords](#)
- [Known Issue](#)

### 17.1 Overview

Oracle Enterprise Manager (EM) is a comprehensive monitoring tool. It includes support for monitoring Oracle's SOA Products, J2EE Servers, Databases, JVMs, and Operating Systems.

The EM Integration utility closes the loop between runtime and design-time by providing a summary of run-time performance metrics from EM into Oracle Enterprise Repository. The EM Integration utility pulls metrics from EM and pushes to Oracle Enterprise Repository. The EM Integration tool currently supports and works with EM 12c.

EM can monitor multiple versions of a product or target simultaneously for one EMGC instance. For example, Oracle Service Bus 11g and 12c can be simultaneously monitored by EM, and hence metrics for both these instances can be pulled back into a single Oracle Enterprise Repository instance, which has projects harvested from Oracle Service Bus 11g or 12c. Oracle Enterprise Repository-EM Integration also supports gathering metrics for multiple product version targets into Oracle Enterprise Repository.

#### 17.1.1 Prerequisites

Before using the EM Integration utility, you must perform the following prerequisites:

- The EM Integration utility requires that the OER-Harvester datapack is installed in Oracle Enterprise Repository.
- The assets that receive metrics from EM must already be in Oracle Enterprise Repository. They should have been published to Oracle Enterprise Repository from SOA Suite/BPEL-PM, WebLogic, or Oracle Service Bus using the Oracle Enterprise Repository Harvester.



For more information about harvesting, see [Chapter 8, "Configuring and Using Automated Harvesting"](#).

- The EM Integration utility needs read access to the Enterprise Manager shared database views.
- The EM Integration utility requires Java SDK version 6 or higher.
- Install EM 12c to enable EM Integration tool to harvest metrics into Oracle Enterprise Repository. The Oracle Enterprise Repository EM Integration tool integrates with EM and works with both 11g and 12c version of SOA.
- When integrating with EM 12c, use the em-integration-settings.xml configuration file. However, there are a few changes required:
  - For SOASuite Metrics, change the metricName attribute from soainfra\_binding\_rollback to soa\_composite\_services\_rollback.
  - For Weblogic web services, change the metricName attribute from j2ee\_app\_jees\_ws\_port\_rollback to j2ee\_app\_jrf\_ws\_port\_properties.
  - Under target mappings, for the target type of j2ee\_application, change the primaryMetrics metricName from j2ee\_app\_jees\_ws\_port\_properties to j2ee\_app\_jrf\_ws\_port\_properties.
  - Under target mappings, for the target type of oracle\_soa\_composite, add the following targetMetric:

```
<targetMetric name="primaryMetrics" metricName="soainfra_composite_label_rollback_by_revision" metricColumn="soainfra_composite" />
```

## 17.1.2 Obtaining the Enterprise Manager Integration Utility

To obtain the Enterprise Manager Integration utility:

1. Navigate to this `Oracle_HOME\oer\modules\tools\solutions` directory on your computer. A list of all the solution packs for Oracle Enterprise Repository is displayed.
2. Unzip the `12.1.3.0.0-EM-Integration.zip` file to a temporary directory.
3. Import the `12.1.3.0.0-OER-Harvester-Solution-Pack.zip` file into Oracle Enterprise Repository using the Import/Export Tool. For more information, see [Chapter 5, "Import Export Tool"](#).

## 17.1.3 High Level Use Cases

You can use the EM Integration utility to:

- Publish metrics from Oracle Service Bus Services in EM to existing Oracle Enterprise Repository assets.
- Publish metrics from BPEL-PM Processes in EM to existing Oracle Enterprise Repository assets.
- Publish metrics from Web Services in EM to existing Oracle Enterprise Repository assets.
- Publish Endpoint and Deployment: BPEL assets in Oracle Enterprise Repository from deployment information in EM.
- Support links back to the detailed metrics in EM, from the Oracle Enterprise Repository Web UI.



- Support configurability and extensibility, so that new targets in EM can be mapped to Oracle Enterprise Repository assets.
- Support capturing metrics for Oracle SOA Suite composite targets in EM. EM Integration enables you to publish metrics for composites that invoke a service and are monitored in EM. However, unlike the other targets such as Oracle Service Bus, BPEL, or WebLogic, if the service endpoint does not exist in Oracle Enterprise Repository, it is not created during the integration process. If the assets do exist, they are updated with the metrics.

## 17.2 Using the Enterprise Manager Integration Utility

This section describes how to use the Enterprise Manager Integration Utility.

### 17.2.1 Running from Command Line

The EM Integration utility can be run in the command line using the `em-integration.bat` utility (for Windows) or `em-integration.sh` (for Linux and Unix).

Before running `em-integration.bat` or `em-integration.sh`, ensure that the environment variables mentioned in [Table 17-1](#) are set.

In Windows, from a command window, type `"set JAVA_HOME "` to see the value of the variable `"JAVA_HOME"` and `"set JAVA_HOME=C:\jdk6\bin\java"` to set the value of `"JAVA_HOME"` to `"C:\jdk6\bin\java"`.

In Linux, type `"echo $JAVA_HOME"` to see the value of the variable `"JAVA_HOME"`, and if not set then you can set a value of `"/usr/bin/java"` to `"JAVA_HOME"` using `"export JAVA_HOME=/usr/bin/java"`.

**Table 17-1** Command Line Script

Environment Variable	Description
JAVA_HOME	Ensure that the <code>JAVA_HOME</code> environment variable points to an installed java runtime (JRE) or SDK. This must be Java version 5 or higher.
JAVA_OPTS	Ensure that you set your <code>JAVA_OPTS</code> parameter as follows: <pre>set JAVA_OPTS=-Dhttp.proxyPort=80 -Dhttp.proxyHost=www-proxy.example.com -Dhttp.nonProxyHosts= "*.example.com localhost" ..."</pre> <code>JAVA_OPTS</code> refers to the extra options to the java executable. In normal cases, there is no must set this variable. However, a common exception when you must set this variable is when your computer is inside a firewall, and you must use an HTTP proxy to access external servers. See Also: <a href="http://java.sun.com/javase/6/docs/technotes/guides/net/proxies.html">http://java.sun.com/javase/6/docs/technotes/guides/net/proxies.html</a>

[Table 17-2](#) shows the options that can be specified using the EM Integration command line utility:

**Table 17-2** Command Line Options for the EM Integration

EM Integration Options	Description
<code>-settings &lt;file&gt;</code>	Refers to the configuration settings specified in the XML file.

**Table 17-2 (Cont.) Command Line Options for the EM Integration**

EM Integration Options	Description
-er_url <URL>	Specifies the URL for the Oracle Enterprise Repository instance.
-er_user <User Name>	Specifies the name of the Oracle Enterprise Repository user.
-er_password <Password>	Specifies the password of the Oracle Enterprise Repository user. To ensure security, the password must be encrypted. The Oracle Enterprise Repository Web console has a tool to encrypt passwords: <i>http://&lt;host&gt;:&lt;port&gt;/&lt;domain&gt;/diag/encryptstrings.jsp</i>
-em_url <URL>	Specifies the JDBC URL for the EM database.
-em_user <User Name>	Specifies the EM database user name.
-em_password <Password>	Specifies the EM database password. To ensure security, the password must be encrypted. The Oracle Enterprise Repository Web console has a tool to encrypt passwords: <i>http://&lt;host&gt;:&lt;port&gt;/&lt;domain&gt;/diag/encryptstrings.jsp</i> For more information about encrypting passwords from command-line, see <a href="#">Section 17.4, "Encrypting the Configuration File Passwords"</a> .
-version	Displays version information.
-help	Displays the online help for the EM Integration utility.

---



---

**Note:** Note none of the above options are mandatory and these can be omitted. If the options are omitted, then the utility makes use of the information in the `em-integration-settings.xml` file in the current directory where `em-integration.bat` resides. If the command line parameters are provided, then these override the information in `em-integration-settings.xml`.

---



---

Figure 17-1 shows the command line utility options and online help displayed by the `em-integration.sh -help` command.

**Figure 17-1 Enterprise Manager Integration Command-Line Utility Options**

```

EM-tool]$ ./em-integration.sh -help
Oracle Enterprise_Repository_EM-Integration version: v12.1.3.0.0-141011_1907-1631476
usage:
em-integration <-settings | -er_url -er_user -er_password | -em_url -em_user -em_password | -help>
- settings <file>           : Configuration Settings XML file
- er_url <URL>              : OER URL
- er_user <username>       : OER User Name
- er_password <password>   : OER Password
- em_url <URL>             : EM Database JDBC URL
- em_user <username>       : EM Database username
- em_password <password>   : EM Database password
- version                   : Print version information
- help                     : Print command line options

```

## 17.2.2 Scheduling from Enterprise Manager

Enterprise Manager can schedule command-line applications to run periodically. The EM Integration utility can be scheduled to run this way.

For more information, see "Job System" in *Oracle Enterprise Manager Administrator's Guide*.

It is recommended that you schedule the EM Integration to run once a day, in the evening or when there is low activity on the system.

## 17.2.3 Metric Publishing

The section describes the various metric publishing options:

### 17.2.3.1 Oracle Service Bus

EM Integration updates existing assets in Oracle Enterprise Repository of type "Endpoint", which have an endpointURI attribute that matches the endpointURI metric in EM, for Oracle Service Bus Proxy Services and External Services.

---

**Note:** An External Service is a service, which is invoked by an Oracle Service Bus Business Service.

---

If no "Endpoint" asset is found with the matching endpointURI, then the tool attempts to find a matching Service asset, by name and Oracle Service Bus service type (BusinessService or ProxyService). If it finds one, then it creates a new "Endpoint" asset and relates it to the service. In the case of Business Services, the endpoint is related to the External Service invoked by the Business Service.

If there are issues collecting metrics for the Oracle Service Bus target in EM, such as the metrics are not being collected or there are errors in EM Grid Control to collect the Oracle Service Bus service metrics, then see Oracle Enterprise Manager Getting Started with Fusion Middleware Management.

---

**Note:** The first time the metrics for Oracle Service Bus services are uploaded to the EM, the values are "0". If no metrics are collected, then turn the monitoring flag *On* using Oracle Service Bus Console -> Global Settings.

Also, ensure that if there are any additional monitoring flags for any Proxy/Business Service in Oracle Service Bus (within the service Operational Settings), enable them to gather metrics for all the services.

---

### 17.2.3.2 BPEL PM

EM Integration updates existing assets in Oracle Enterprise Repository of type "Deployment:BPEL", which has the "EntryPoint" relationship to an "Endpoint" asset that has an endpointURI attribute that matches the endpointURI metric in EM. EM stores the WSDL location of the endpoint.

If no "Endpoint" asset is found with the matching endpointURI, then the tool attempts to find a matching Service asset, by qualified name from the WSDL. If it finds one, then it creates a new "Endpoint" asset and relates it to the service.

If no "Deployment:BPEL" asset is found, then the tool attempts to find a matching Business Process: BPEL asset, by process name. If it finds one, then it creates a new "Deployment:BPEL" asset and relates it to the service.

### 17.2.3.3 WebLogic Web Services

EM Integration updates existing assets in Oracle Enterprise Repository of type "Endpoint", which have an endpointURI attribute that matches the endpointURI metric in EM.

If no "Endpoint" asset is found with the matching endpointURI, then the tool attempts to find a matching Service asset, by qualified name from the WSDL. If it finds one, then it creates a new "Endpoint" asset and relates it to the service.

### 17.2.3.4 Oracle SOA Suite 11g

EM Integration enables you to publish metrics only at a composite level. If an "endpoint" asset exists in OER for a composite Service being monitored in EM 11g, then metrics are updated for it in OER. However, unlike the other targets such as Oracle Service Bus, BPEL, or WebLogic, if the service endpoint does not exist in Oracle Enterprise Repository, it is not created during the integration process.

### 17.2.3.5 Artifacts Creation

For every "Endpoint" that is created, the EM Integration Utility creates a corresponding Artifact:WSDL asset, if it does not already exist in Oracle Enterprise Repository. The EM Integration Utility also creates the Artifact:WSDL and Artifact:XSD assets for any imported file. It follows the fingerprinting rules in Oracle Enterprise Repository Harvester to look for existing artifacts in Oracle Enterprise Repository.

### 17.2.3.6 Endpoint Creation

It is common for production endpoints to be monitored in EM, even if they are not yet captured in Oracle Enterprise Repository.

The EM Integration utility creates Endpoint assets, if it can find a matching service and creates Deployment:BPEL assets, if it can find a matching Business Process:BPEL.

### 17.2.3.7 Metrics to Update

In its default configuration, EM Integration attempts to update the following metrics in Oracle Enterprise Repository. These are all visible in the Operational tab for the asset in the Oracle Enterprise Repository web user interface.

Not all of these metrics are gathered in EM for all of the target types mentioned above, so some might be omitted. The em-integration-settings.xml file configures which metrics are gathered for which targets.

**See Also:** [Section 17.3.2.1, "Metric Mappings"](#)

- Daily Average Response Time
- Weekly Average Response Time
- Monthly Average Response Time
- Minimum Response Time
- Maximum Response Time
- Daily Requests
- Weekly Requests
- Monthly Requests

- Daily Faults
- Weekly Faults
- Monthly Faults
- Availability - Daily %
- Availability - Weekly %
- Availability - Monthly %
- Current Rated Maximum Requests per Second
- Start Date for Metrics Monitoring
- Last Updated
- Link back to detailed metrics in EM

## 17.3 Configuring the Enterprise Manager Integration Utility

This section describes how to configure the Enterprise Manager Integration Utility.

### 17.3.1 Setting the Repository and Enterprise Manager Connection Information for the Command-line Utility

Open the `em-integration-settings.xml` file, which is located at `<EM Integration Home>` (the directory to which you unzipped the `12.1.3.0.0-EM-Integration.zip` file) and modify the following XML chunk to point to the Oracle Enterprise Repository instance, the EM database, and the web console with the right credentials.

The EM database is used to retrieve the metric information that is sent to Oracle Enterprise Repository. The EM console URL is used as the base URL for links back to EM, that are created within Oracle Enterprise Repository.

The `introspectionSettings` section is used to configure the properties of any Endpoint assets that the utility creates. The `createMissingEndpoints` attribute indicates whether Endpoint and `Deployment:BPEL` assets are created in the scenario where EM monitors the endpoint or `Deployment:BPEL` but the corresponding assets do not exist in OER for the matching service. If false, then the rest of the section in the code below is ignored.

The `matchMultipleEndpoints` attribute indicates the required action to be taken if the utility finds multiple Endpoints or `Deployment:BPEL` assets in Oracle Enterprise Repository that match a target in EM. If true, then it updates all the endpoints with metrics (and log a warning). If false, then it skips the target (and log a warning).

The `matchMultipleServices` attribute indicates the required action to be taken if the utility finds multiple services or `Business Process:BPEL` assets in Oracle Enterprise Repository that match a target in EM, during endpoint creation. If true, then it attaches new Endpoints / `Deployment:BPEL`s to each of the matching assets (and log a warning). If false, then it skips the target (and log a warning).

The `introspectionDescription`, `introspectionVersion`, and `registrationStatus` attributes indicate properties that are set on any assets created.

**Example 17–1 Setting the Repository and Enterprise Manager Connection Information in `em-integration-settings.xml`**

```
<repository>
  <uri>http://localhost:7131/oes30/services/FlashlineRegistry</uri>
```

```

<credentials>
  <user>admin</user>
  <password>*****</password>
  <enableTransaction>>false</enableTransaction>
  <triggerEvent>>false</triggerEvent>
</credentials>
<timeout>120000</timeout>
</repository>
<introspectionSettings>
  <createMissingEndpoints>>true</createMissingEndpoints>
  <introspectionDescription>Found in EM </introspectionDescription>
  <introspectionVersion>1.0</introspectionVersion>
  <namespace>mythings</namespace>
  <!-- The Valid Registration states are 1)Unsubmitted 2)Submitted - Pending
Review 3)Submitted - Under Review 4) Registered -->
  <registrationStatus>Submitted</registrationStatus>
</introspectionSettings>
<emDatabase>
  <driverClass>oracle.jdbc.driver.OracleDriver</driverClass>
  <url>jdbc:oracle:thin@remotehost.there.com:1501:mysid</url>
  <credentials>
    <user>sysman</user>
    <password>*****</password>
  </credentials>
</emDatabase>

```

---



---

**Note:** It is recommended that you run the Enterprise Manager Integration Utility as a user with the Basic Access Settings for Assets - View, Edit, Accept, and Register.

---



---

Alternatively, the connection information can also be passed as parameters to the command line utility as follows:

```

C:\test\em-integration> em-integration - er_url
http://localhost:7101/oer/services/FlashlineRegistryTr -er_user admin
-em_url jdbc:oracle:thin:@remotehost.there.com:1501:mysid - em_user sysman

```

---



---

**Note:** To ensure security, passwords are not passed via the command line and they must be encrypted. See [Section 17.4, "Encrypting the Configuration File Passwords"](#) for more information.

---



---

## 17.3.2 Advanced Configuration

This section describes the following advanced configuration options for the Enterprise Manager Integration utility.

### 17.3.2.1 Metric Mappings

The mapping between metrics in EM and fields in Oracle Enterprise Repository are configured in `em-integration-settings.xml`, as shown in [Figure 17-2](#). To change these from the default settings, open this file and modify the following:

- `<metricMappingGroup>`: Mappings for a set of metrics with the same metric name in the EM database views. EM groups a set of related metrics under the same "metric name". Each individual metric has a different "metric column". This element has the following attribute:

- metricName: The EM metric name from the EM database views.
- <metricMapping>: Mapping for a single metric. This element has the following attributes:
  - metricColumn: The EM metric column from the EM database views.
  - rollupPeriod: Indicates which EM database view to query against, and how to aggregate the data. Must be one of {DAILY, WEEKLY, MONTHLY, CURRENT}.
  - viewColumn: Indicates which column in the EM database view to pull the data from. Must be one of {VALUE, AVERAGE, SUM, MINIMUM, MAXIMUM, STANDARD\_DEVIATION, SAMPLE\_COUNT, COLLECTION\_TIMESTAMP, ROLLUP\_TIMESTAMP, NOW(), DETAILS\_URL}.
    - \* The CURRENT rollup period does not support columns AVERAGE, SUM, MINIMUM, MAXIMUM, STANDARD\_DEVIATION, SAMPLE\_COUNT, or ROLLUP\_TIMESTAMP.
    - \* The DAILY, WEEKLY and MONTHLY rollup periods do not support columns VALUE or COLLECTION\_TIMESTAMP.
    - \* NOW() is a "virtual column" that contains the time of the query against EM.
    - \* DETAILS\_URL is a "virtual column" that contains a URL to the detailed information in EM. This is constructed by the EM Integration code - it's not in the EM database views.
  - oerField: The internal name of the custom data field in Oracle Enterprise Repository where this data is stored.
  - overwrite: Indicates whether to overwrite the custom data field in Oracle Enterprise Repository if it already has data. It represents either true or false; by default the value is true, if not specified.

**Figure 17–2 Metric Mappings**

```

<!--OSB Metrics-->
<metricMappingGroup metricName="OSBResourceURIMetrics">
  <metricMapping metricColumn="countStat" rollupPeriod="DAILY" viewColumn="SUM" oerField="daily-requests"/>
  <metricMapping metricColumn="countStat" rollupPeriod="WEEKLY" viewColumn="SUM" oerField="weekly-requests" />
  <metricMapping metricColumn="countStat" rollupPeriod="MONTHLY" viewColumn="SUM" oerField="monthly-requests" />
  <metricMapping metricColumn="errorStat" rollupPeriod="DAILY" viewColumn="SUM" oerField="daily-faults" />
  <metricMapping metricColumn="errorStat" rollupPeriod="WEEKLY" viewColumn="SUM" oerField="weekly-faults" />
  <metricMapping metricColumn="errorStat" rollupPeriod="MONTHLY" viewColumn="SUM" oerField="monthly-faults" />
  <metricMapping metricColumn="avgStat" rollupPeriod="DAILY" viewColumn="AVERAGE" oerField="average-response-time--seconds-"/>
  <metricMapping metricColumn="avgStat" rollupPeriod="WEEKLY" viewColumn="AVERAGE" oerField="weekly-average-response-time--milliseconds-"/>
  <metricMapping metricColumn="avgStat" rollupPeriod="MONTHLY" viewColumn="AVERAGE" oerField="monthly-average-response-time--milliseconds-"/>
  <metricMapping metricColumn="avgStat" rollupPeriod="MONTHLY" viewColumn="MAXIMUM" oerField="maximum-response-time--milliseconds-"/>
  <metricMapping metricColumn="avgStat" rollupPeriod="MONTHLY" viewColumn="MINIMUM" oerField="minimum-response-time--seconds-"/>

  <!--pick an arbitrary metric to pull the timestamps and details from-->
  <metricMapping metricColumn="avgStat" rollupPeriod="CURRENT" viewColumn="COLLECTION_TIMESTAMP" oerField="last-updated" />
  <metricMapping metricColumn="avgStat" rollupPeriod="CURRENT" viewColumn="NOW()" oerField="endpoint-publication-date" overwrite="false" />
  <metricMapping metricColumn="avgStat" rollupPeriod="CURRENT" viewColumn="DETAILS_URL" oerField="detailed-performance-information" />
</metricMappingGroup>

```

### 17.3.2.2 Target Finders

The logic for correlating targets in EM with assets in Oracle Enterprise Repository is different for each target type. EM Integration ships with target finders for the following products: Oracle Service Bus, BPEL-PM, WebLogic.

If you are an advanced user, then you can create your own target finder classes. These must be configured in `em-integration-settings.xml`, as shown in [Figure 17–3](#).

- <targetFinder>: Mappings for a target type in EM to a target finder class. This element has the following attributes
  - targetType: The EM target type from the EM database views.
  - targetFinderClass: The fully qualified classname of the finder class. This must implement com.oracle.oer.integration.em.TargetFinder, which is found in em-integration.jar.

**Figure 17–3 Target Finders**

```
<!-- target finders -->
<targetFinder targetType="bea_alsb" targetFinderClass="com.oracle.oer.integration.em.targets.OSBTargetFinder" />
<targetFinder targetType="oracle_integrationbpm" targetFinderClass="com.oracle.oer.integration.em.targets.BPELPMTargetFinder" />
<targetFinder targetType="oc4j" targetFinderClass="com.oracle.oer.integration.em.targets.OC4JTargetFinder" />
<targetFinder targetType="weblogic_j2eeserver" targetFinderClass="com.oracle.oer.integration.em.targets.WebLogicTargetFinder" />
<targetFinder targetType="websphere_j2eeserver" targetFinderClass="com.oracle.oer.integration.em.targets.WebSphereTargetFinder" />
```

### 17.3.2.3 Logging

EM Integration uses log4j for logging the detailed tasks performed and the log file is placed in the <EM Integration Home>. The logging options can be changed by updating the log4fl.properties file located in the <EM Integration> folder.

## 17.4 Encrypting the Configuration File Passwords

To ensure security, the passwords in the configuration file must be encrypted before running the Enterprise Manager Integration utility. You can encrypt the configuration file passwords in the EM using either of the following methods:

- ["From the Command Prompt"](#)
- ["From the Oracle Enterprise Repository Web Console"](#)

### From the Command Prompt

1. Navigate to the <em\_integration\_home> directory.
2. From a command prompt, run the password encryption tool as follows:
 

```
> encrypt.bat em-integration-settings.xml em-integration-settings.xml
```

The password encryption tool, (encrypt.bat/encrypt.sh), which is located in <Oracle\_home>/oer/modules/tools/solutions/12.1.3.0.0-OER-PasswordTools.zip, enables you to encrypt the passwords that are stored in the configuration file, em-integration-settings.xml.

### From the Oracle Enterprise Repository Web Console

1. Navigate to the Oracle Enterprise Repository web console at
 

```
http://<host>:<port>/<domain>/diag/encryptstrings.jsp
```

 (replace host, port, and domain to match to your computer credentials)
2. Scroll down to the Tools section and click the **Encrypt Strings for passwords** link to launch the Password encryption page.
3. Enter the clear text password into the String to Encrypt text box.
4. Click the **Submit Query** button.
5. Copy the resulting encrypted password string and paste it into the appropriate context or properties file(s).



## 17.5 Known Issue

This section describes the following known Enterprise Manager Integration Utility issue.

### 17.5.1 Using Incorrect Encrypted Password

If you configure the `em-integration-settings.xml` file with incorrect encrypted password, then a very long stack trace is displayed. An extract from the very long stack trace is as follows:

```
at org.apache.xerces.impl.XMLNSDocumentScannerImpl.scanStartElement
  (Unknown Source)
at org.apache.xerces.impl.XMLDocumentFragmentScannerImpl$FragmentContent
  Dispatcher.dispatch(Unknown Source)
at org.apache.xerces.impl.XMLDocumentFragmentScannerImpl.scanDocument
  (Unknown Source)
at org.apache.xerces.parsers.XML11Configuration.parse(Unknown Source)
at org.apache.xerces.parsers.DTDConfiguration.parse(Unknown Source)
at org.apache.xerces.parsers.XMLParser.parse(Unknown Source)
at org.apache.xerces.parsers.AbstractSAXParser.parse(Unknown Source)
at javax.xml.parsers.SAXParser.parse(SAXParser.java:395)
at org.apache.axis.encoding.DeserializationContext.parse
  (Deserialization
  Context.java:227)
at org.apache.axis.SOAPPart.getAsSOAPEnvelope(SOAPPart.java:696)
at org.apache.axis.Message.getSOAPEnvelope(Message.java:424)
at org.apache.axis.handlers.soap.MustUnderstandChecker.invoke(MustUnder
  standChecker.java:62)
at org.apache.axis.client.AxisClient.invoke(AxisClient.java:206)
at org.apache.axis.client.Call.invokeEngine(Call.java:2765)
at org.apache.axis.client.Call.invoke(Call.java:2748)
at org.apache.axis.client.Call.invoke(Call.java:2424)
at org.apache.axis.client.Call.invoke(Call.java:2347)
at org.apache.axis.client.Call.invoke(Call.java:1804)
at com.flashline.registry.openapi.service.v300.FlashlineRegistryTrSoap
  BindingStub.authTokenCreateWithLicense(FlashlineRegistryTrSoapBindingStub.
  java:741)at com.oracle.oer.sync.plugin.writer.oer.ALERConnectionCache.
  getAuthToken(ALERConnectionCache.java:128)
at com.oracle.oer.sync.plugin.writer.oer.ALERAssetQueries.getToken
  (ALERAssetQueries.java:82)
at com.oracle.oer.sync.plugin.writer.oer.ALERAssetQueries.assetTypeQuery
  ByUUID(ALERAssetQueries.java:159)
at com.oracle.oer.sync.framework.MetadataManager.putAssetType
  (MetadataManager.java:204)atcom.oracle.oer.sync.framework.impl.
  DefaultPluginManager.processIntrospector(DefaultPluginManager.java:104)
... 5 more
```



---

## Configuring Metrics

This chapter describes how to configure metrics in your organization.

This chapter contains the following sections:

- [Section 18.1, "Overview"](#)
- [Section 18.2, "Validate the Elements on the Asset Type"](#)
- [Section 18.3, "Running Queries Against the Database"](#)

### 18.1 Overview

This section addresses metrics enhancements available in Oracle Enterprise Repository. Information in this guide will assist those who wish to add metrics fields to asset types, which do not already have them. The metrics fields provided by this guide are, Total Development Hours (TDH), Production Investment (Pinv), Consumption Factor (Cfac), Predicted Number of Annual Reuse Opportunities (n), and Hourly Burden Rate (B). These fields most commonly appear on the Metrics tab in the Asset Editor.

#### 18.1.1 Metrics Elements

This section explains how to add metrics elements to a Type. These metrics are used to estimate the value of each asset and to determine the ROI of the asset portfolio.

A total of five metrics elements must be added, and the Development Hours element must already exist in the Type. If it does not, or if it has been deleted, it, too, must be added. The necessary changes are made in the Type Manager and the database; instructions for both are included here. The new metrics elements are listed below:

- **Total Development Hours (TDH)** - the total development effort expended to create this asset.
- **Production Investment (Pinv)** - the extra time or effort involved in building or harvesting the asset, and in packaging and documenting the asset to make it reusable. This value is represented as a percentage over the time initially required to build the asset for one-time use.
- **Consumption Factor (Cfac)** - the time an asset consumer spends locating, evaluating, and using an asset, represented as a percentage of the time necessary to build the asset from scratch.
- **Predicted Number of Annual Reuse Opportunities (n)** - indicates the number of times the asset is expected to be used over the course of a year

- **Hourly Burden Rate (B)** - the hourly overhead cost of the individual that produced the asset.

The metrics should be included in the ten base Asset Types, in addition to the other asset types that have been added to your system:

- Application
- Business Process
- Communication Adapter
- Component
- Environment
- Framework
- Pattern
- Process
- Service
- XML Schema

### 18.1.2 Adding a Reusable Metrics Tab

If you are installing Oracle Enterprise Repository for the first time, then you do not have to perform this step. You only need to add reuse metrics to older versions of the product that did not include these metrics. This procedure is performed in the Asset Editor.

Use this text to cut and paste element names:

- Total development hours (TDH)
- Production investment (Pinv)
- Consumption Factor (Cfac)
- Predicted number of annual reuse opportunities (n)
- Hourly burden rate (B)

---

---

**Note:** The `ValidateMetrics.sql` file must be run against the database.

---

---

1. Select **Manage Types** in the Actions menu. The Type Manager launches.
2. Select the Type to which the metrics are to be added from the Type Manager sidebar.
3. Select the **Editor** view.
4. In the Tabs section, click **Metrics**.
5. Click **Add** in the Elements section.
6. Select **Numeric Text Field** from the list, and click **OK**. The Edit Numeric Text Field dialog is displayed.
7. Copy and paste the element name in the Display Name text box.
8. Change the Decimal Places value to 0, as shown in [Figure 18-1](#).

**Figure 18–1 Edit Numeric Text Field Dialog**

The dialog box titled "Edit Numeric Text Field" has the following fields and controls:

- Display Name: [Text Input Field]
- Required:
- Indexed:
- Minimum Value: [0]
- Maximum Value: [99999999]
- Decimal Places: [0]
- Buttons: OK, Cancel

9. Click **OK**.
10. Repeat steps 4 - 9 for each metric.
11. When all metrics have been added to the Type, click **Save**.
  - Do not add the elements to the Viewer tab at this time.
12. Repeat the process for each of the base asset types.

## 18.2 Validate the Elements on the Asset Type

To validate the elements on the asset type, perform the following steps:

1. Select an asset belonging to one of asset types just modified. The easiest way to find such an asset is to right-click the Registered folder in the Asset Editor and sort by asset type.
2. Add a unique value to each metric element on the asset. Recommended values are 1, 2, 3, 4, and 5, as shown in [Figure 18–2](#).

**Figure 18–2 Administration Tab**

The Administration Tab shows the following data:

Metric Name	Value
Total development hours (TDH)	1
Production investment (Plnv)	2
Consumption Factor (CFac)	3.00
Predicted number of annual reuse opportunities (n)	4
Hourly burden rate (B)	5

Approved By: User, Joe    Approved Date: 1 Jan 2005    Unapprove

3. Determine the asset name, version and asset type name from the status bar on the edited asset.
4. Save the asset.

## 18.3 Running Queries Against the Database

The following database queries can be used to validate that the metrics fields were properly added.

You can run the following metrics queries against the database by performing the instructions provided below:

```
-- Metrics SQL (for Oracle, UDB and MsSQL Server)
-- Total Development Hours
select a.name as AssetName, a.version as Version, at.name as
AssetTypeName, axi.stringvalue as tdh
from assetxmlindex axi, assets a, assettypes at
where axi.assetid = a.id and axi.assettypeid = at.id
and axi.fieldid in
      (select id from assetxmlindexmappings where fieldname =
'/custom-data/total-development-hours--tdh-')
order by a.name, a.version;
-- Production Investment
select a.name as AssetName, a.version as Version, at.name as
AssetTypeName, axi.stringvalue as Pinv
from assetxmlindex axi, assets a, assettypes at
where axi.assetid = a.id and axi.assettypeid = at.id
and axi.fieldid in
      (select id from assetxmlindexmappings where fieldname =
'/custom-data/production-investment--pinv-')
order by a.name, a.version;
-- Consumption Factor
select a.name as AssetName, a.version as Version, at.name as
AssetTypeName, axi.stringvalue as Cfac
from assetxmlindex axi, assets a, assettypes at
where axi.assetid = a.id and axi.assettypeid = at.id
and axi.fieldid in
      (select id from assetxmlindexmappings where fieldname =
'/custom-data/consumption-factor--cfac-')
order by a.name, a.version;
-- Predicted Number of Annual Reuse Opportunities
select a.name as AssetName, a.version as Version, at.name as
AssetTypeName, axi.stringvalue as n
from assetxmlindex axi, assets a, assettypes at
where axi.assetid = a.id and axi.assettypeid = at.id
and axi.fieldid in
      (select id from assetxmlindexmappings where fieldname =
'/custom-data/predicted-number-of-annual-reuse-opportunities--n-')
order by a.name, a.version;
-- Hourly Burden Rate
select a.name as AssetName, a.version as Version, at.name as
AssetTypeName, axi.stringvalue as B
from assetxmlindex axi, assets a, assettypes at where axi.assetid =
a.id and axi.assettypeid = at.id
and axi.fieldid in
      (select id from assetxmlindexmappings where fieldname =
'/custom-data/hourly-burden-rate--b-')
order by a.name, a.version;
-- Predicted Net Hours Saved
select a.name as AssetName, a.version as Version, at.name as
AssetTypeName, axi.stringvalue as PNHS
from assetxmlindex axi, assets a, assettypes at
where axi.assetid = a.id and axi.assettypeid = at.id
and axi.fieldid in
      (select id from assetxmlindexmappings where fieldname =
'/custom-data/development-hours')
order by a.name, a.version;
-- End Metrics SQL
```

## 18.3.1 dbqueries

This section describes the instructions to run the metrics queries:

1. Open a connection to the database and run the six queries listed above. The asset name, version, asset type, and metric should be returned as the result. If a metric is not returned, it is likely that the metric name was not copied and pasted accurately. Each metric has its own query; it is recommended that these queries be run separately. Any metric that returns the expected results is validated.
2. If any metric query fails, remove the element and follow the steps to create the metric element, validate it, and run it against the database. If the PNHS element is missing from the selected asset type, it must be added, as described below:
  1. Add the string Development Hours.
  2. Click **OK**.
  3. Edit the Development Hours element and change the name to Predicted Net Hours Saved (PNHS). (The element name in the metrics is based on the Development Hours programmatic name.) The Predicted Net Hours Saved label must appear in the Asset Editor.
3. After all metrics are validated, open the asset type in the Oracle Enterprise Repository Type Manager.
4. Select the **Viewer** tab.

---

---

**Note:**

- All added elements should appear in the Hidden Elements area.
  - Only the Predicted Net Hours Saved (PNHS) metric is meaningful to asset consumers.
  - It is recommended that Predicted Net Hours Saved (PNHS) is displayed in the first column in the Overview group.
- 
- 

5. Save the asset type.
6. Repeat steps 1 - 5 to add metrics elements to other asset types.
7. Remove any sample data that may have been added to test the queries, and save your data.





---

---

## Configuring Assets in Progress

This chapter describes how to use the Assets in Progress feature to track the progress and availability of assets that are still in development.

This chapter contains the following sections:

- [Section 19.1, "Overview"](#)
- [Section 19.2, "Viewing Assets in Progress"](#)
- [Section 19.3, "Enabling Assets in Progress"](#)
- [Section 19.4, "Disabling Assets in Progress"](#)
- [Section 19.5, "Enabling Viewing of Assets in Progress Through Oracle JDeveloper"](#)

### 19.1 Overview

Oracle Enterprise Repository's Assets in Progress feature allows users to track the progress and availability of assets still in development. This allows organizations to realize compound gains in productivity by engaging asset producer, consumer, and maintenance teams in a collaborative process to increase the value of the asset portfolio in the natural course of software engineering. This feature helps organizations eliminate redundant development before it occurs. By exposing assets that are still in development, projects can include the use of these assets in their project plans, rather than wasting time and effort on redundant development.

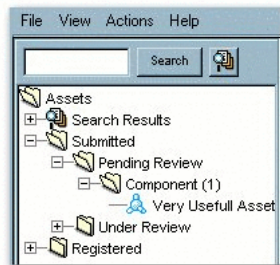
Assets in progress appear in Oracle Enterprise Repository as Unregistered assets. The actual Registration Status of any asset in progress (as indicated in the Status column in search results and in the asset detail) is determined during asset submission.

When the Assets in Progress feature is enabled, any assets submitted via the Submit an Asset link in the Oracle Enterprise Repository Assets screen are automatically assigned the Unregistered status. They will appear in searches filtered for Unregistered assets.

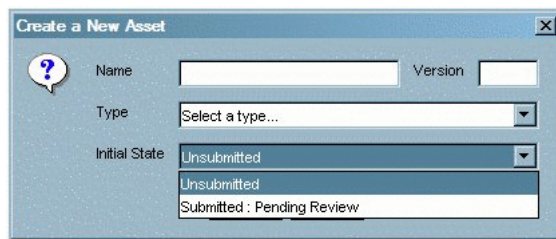
**Figure 19–1 Oracle Enterprise Repository Assets Screen**



Figure 19–1 shows the Oracle Enterprise Repository Assets screen. Assets submitted via Submit an Asset enter the Submitted - Pending Review folder in the file tree in the Asset Editor, as shown in Figure 19–2.

**Figure 19–2 Asset Editor - File Tree**

Assets submitted via the Asset Editor may be designated as Unsubmitted or Submitted - Pending Review in the Initial State list, as shown in [Figure 19–3](#).

**Figure 19–3 Create a New Asset**

Assets in either state enter Oracle Enterprise Repository as Unregistered assets, and will appear in searches filtered for Unregistered assets.

## 19.2 Viewing Assets in Progress

Display a list of all Assets in Progress.

This procedure is performed on the Oracle Enterprise Repository Assets screen.

1. Leave the Enter Search String text box blank.
2. Select **All Types** in the Type list.
3. Select **All Asset Functions** in the Asset Function list.
4. Select **Unregistered** from the Registration Status list, as shown in [Figure 19–4](#).

**Figure 19–4 Search Dialog**

5. Click **Search**. A list of all Assets in Progress is displayed in the upper frame of the main pane.

## 19.3 Enabling Assets in Progress

Two properties must be enabled in order to activate Assets in Progress. This procedure is performed on the Oracle Enterprise Repository Admin screen.

1. Click **System Settings** in the left pane.
2. Enter `Assets in Progress` in the Search field. The Assets in Progress setting appears in the Functional Settings pane, as shown in [Figure 19–5](#).
3. Ensure the property is set to **True**.
4. Click **Save**.
5. Enter the `cmee.asset.in-progress.visible` property in the Enable New System Setting text box.
6. Click **Enable**. The View Assets in Progress property appears in the Functional Settings pane, as shown in [Figure 19–5](#).

**Figure 19–5 Assets in Progress Properties**

Assets in Progress	
<b>Assets in Progress</b> <code>cmee.asset.in-progress</code>	<input checked="" type="radio"/> True <input type="radio"/> False Enables the assets in progress feature.
<b>View Assets in Progress</b> <code>cmee.asset.in-progress.visible</code>	<input checked="" type="radio"/> True <input type="radio"/> False Enables the ability to see assets in progress outside the Asset Editor.

7. Ensure the property is set to **True**.
8. Click **Save**. The Registration Status list will now appear in the Search section in the Oracle Enterprise Repository Assets screen, as shown in [Figure 19–6](#).

**Figure 19–6 Search Dialog**

### 19.3.1 Set Default Asset Status

Enable `oer.newasset.initial.state` to set the default asset status.

1. Enter `oer.newasset.initial.state` property in the **Enable New System Setting** text box.
2. Click **Enable**.
3. Select one of the available status options and click **Save**.

## 19.4 Disabling Assets in Progress

This procedure is performed on the Oracle Enterprise Repository Admin screen.

---

**Note:** Disabling the Assets in Progress feature prevents Asset Editor access to assets in the Unsubmitted state. Assets in the Submitted - Pending Review state will remain accessible to the Asset Editor. Before disabling Assets in Progress it is recommended that all necessary assets be promoted from Unsubmitted to Submitted - Pending Review.

---

1. Click **System Settings**.
2. Set the **Assets in Progress** property to `False`.
3. Set the **View Assets in Progress** property to `False`.
4. Click **Save**.

## 19.5 Enabling Viewing of Assets in Progress Through Oracle JDeveloper

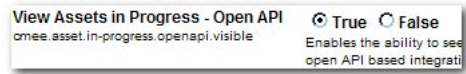
The Oracle Enterprise Repository Open API (part of the Oracle Enterprise Repository Extensibility Framework) makes it possible to enable viewing of both registered and unregistered assets in search results through Oracle JDeveloper.

You can enable the required property by performing the following procedure on the Oracle Enterprise Repository Admin page:

1. Click **System Settings** in the left pane.
2. Enter the property `cmee.asset.in-progress.openapi.visible` in the Enable New System Setting text box.

3. Click **Enable**. View Assets in Progress - Open API appears in the list of properties in the main pane, as shown in [Figure 19-7](#).

**Figure 19-7 View Assets in Progress**



4. Ensure the property is set to True.
5. Click **Save**.



---

---

## Configuring Contract Management

This chapter describes how to use the contract management solution pack in Oracle Enterprise Repository.

This chapter contains the following sections:

- [Section 20.1, "Overview"](#)
- [Section 20.2, "Import the Contract Management Solution Pack"](#)
- [Section 20.3, "Instructions for Service Providers"](#)
- [Section 20.4, "Instructions for Service Consumers"](#)

### 20.1 Overview

Contract management supports the negotiation of contracts between service providers and consumers. It also provides Terms of Use (i.e. the conditions under which service will be provided or used), and the enforcement of runtime policies in connection with the Terms of Use.

A formal contract to use services is advisable in any circumstance where access to production services is mission critical. By having advance notice of a Consumer's intent to access production services, the Provider can explicitly authorize access, plan for extra capacity, ensure policies are enforced, and manage service levels more easily for all consumers.

Contract Management provides the following:

- Service providers and consumers can articulate and negotiate service level and access requirements to avoid unset expectations in the use of services.
- Service providers can manage the service level, scalability and security risks associated with consumers coming on board to reuse their services.
- Service providers can automate a complex procedure that may entail customer acquisition, contract negotiations, the provisioning of rights, and the enforcement of policy.
- Policy Administrators are notified to enforce policies that the business requires, as specified in the contract.

#### 20.1.1 Prerequisites

- Harvester Solution Pack (12.1.3.0.0-OER-Harvester-Solution-Pack.zip)
- Policy Solution Pack (Policy-Management-Solution-Pack.zip)

## 20.2 Import the Contract Management Solution Pack

A sample contract is included in the Oracle Enterprise Repository Base datapack and is imported into your Oracle Enterprise Repository instance. For more information about Oracle Enterprise Repository Import/Export capabilities, see [Chapter 5, "Import Export Tool"](#).

## 20.3 Instructions for Service Providers

This is a standard agreement that consumers can use to request access to a specified service endpoint from the provider. This contract represents the agreement between [consumer] and [provider] for access to the [service name] Web Service.

### 20.3.1 Create the Contract

Use the Copy/Migrate option from the Oracle Enterprise Repository Asset Editor to create a new contract based on this sample. See [Chapter 9, "Using the Asset Editor"](#) for more information about the Asset Editor.

1. Select the contract and click **Edit**. The Asset Editor displays.
2. From the File Menu, select **Copy/Migrate**. The Copy/Migrate dialog displays.
3. Enter a **Name** for the new contract and a **Version Number**.
4. Select **Contract** from the Type list and select an **Initial State**.

### 20.3.2 Edit the Contract

Enter provider information on the Overview tab, as well as additional information about the contract. Do not change the consumer information fields.

1. Select the Overview tab and enter information about the provider, purpose, service charge, and any additional fields.
2. Change the placeholder text indicated by brackets [] in the **Description** field.
3. Save your changes.

### 20.3.3 Create Relationships

For each copy of the contract, create a Prescribed Asset(s) relationship on the Taxonomy tab to the Service, WSDL and Endpoint assets governed by the contract.

1. Select the Taxonomy tab, and then click **Add** in the Relationships section. The Add Relationships dialog displays.
2. Select Relationship Type: Prescribed Assets.
3. Enter a term to Search or select **List All Active Assets** in the Find Assets to Relate section.
4. Select a Service, WSDL, and an Endpoint asset and confirm relationship.

#### 20.3.3.1 Terms of Use

The Provider receives the contract request and must complete more details before sending the contract back to the consumer for approval.

Apply the Terms of Use (policy) to the service endpoint governed by this contract.



1. Make sure the endpoint is secure prior to offering consumer access to the endpoint.
2. Make a copy of the EasyLink URL for the boilerplate service contract (the copy of this asset). See "Using EasyLinks" in the *Oracle Fusion Middleware Developer's Guide for Oracle Enterprise Repository* for more information about using EasyLinks.
3. Go to the Service asset and paste the EasyLink URL into the **{click here to request a contract to use the service endpoint}** field.
4. **Optional:** Delete the Instructions for Providers for the consumer version of the contract. Do not delete the Instructions for Consumers.
5. From the Overview tab, edit the **Description** field to remove the Instructions for Providers text.

## 20.4 Instructions for Service Consumers

Create a copy of this contract: Use Copy/Migrate from the Oracle Enterprise Repository Asset Editor to create your own copy of the this contract. See [Chapter 9, "Using the Asset Editor"](#) for more information about the Asset Editor.

1. Select the contract and click **Edit**. The Asset Editor displays.
2. From the File menu, select **Copy/Migrate**. The Copy/Migrate dialog displays.
3. From the Overview tab, enter your information in the consumer fields. Do not change the provider information fields.
4. From the Administration tab, click **Submit** to submit the contract. It is submitted to the provider for approval and registration. You will receive an email notification when the provider grants access to the service.



---

# Configuring Oracle Enterprise Repository to Exchange Metadata with the Oracle Service Registry

This chapter describes how to configure Oracle Enterprise Repository to exchange metadata with the Oracle Service Registry.

This chapter contains the following sections:

- [Section 21.1, "Getting Started With the Oracle Registry Repository Exchange Utility"](#)
- [Section 21.2, "Configuring the Oracle Registry Repository Exchange Utility"](#)
- [Section 21.3, "Using the Oracle Enterprise Repository Exchange Utility"](#)

## 21.1 Getting Started With the Oracle Registry Repository Exchange Utility

This section describes how to get started with Oracle Registry Repository Exchange Utility and sample use cases using the utility.

This section contains the following topics:

- [Section 21.1.1, "What is the Oracle Registry Repository Exchange Utility?"](#)
- [Section 21.1.2, "Example Use Cases"](#)

### 21.1.1 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.

- Being triggered by Oracle Business Process Management workflows to automatically publish assets to the Oracle Service Registry.

#### 21.1.1.1 Valid Metadata Entities

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

- All UDDI enabled services and SCA Service type as well as any custom services of any Oracle Enterprise Repository Asset Type developed by end users.
- Business Entities that provide Business Services.

For more information about how the Oracle Registry Repository Exchange Utility arrives at the Business Entities, see [Section 21.3.2.1](#) and [Section 21.3.2.2](#).

- 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

For more information, see the `-publish_tmodel` option in [Table 21-4](#).

- 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.
- Non-WSDL based services are published to the UDDI Registry.
- Services without an HTTP transport are not published to the UDDI Registry.
- Service Keys - When Oracle Enterprise Repository promotes services to multiple service registries, the service keys remain consistent and the same service key appears in each service registry. If services are published from an Oracle Service Registry to Oracle Enterprise Repository, then Oracle Enterprise Repository uses that service key whenever it promotes services to other service registries.

#### 21.1.1.2 Prerequisites

Before using the Oracle Registry Repository Exchange Utility, you must perform the following prerequisites:

- Oracle Registry Repository Exchange Utility requires Java SDK version 6 or higher.

### 21.1.2 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 services and assets are published to multiple UDDI registry environments, a single UDDI ServiceKey is preserved in each registry.

- 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.
- The Exchange Utility publishes certain kinds of Non-WSDL services from Oracle Enterprise Repository to Oracle Service Registry. These services are not bi-directional, which means that the Exchange Utility cannot publish back to Oracle Enterprise Repository from Oracle Service Registry.
- 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.

---

---

## 21.2 Configuring the Oracle Registry Repository Exchange Utility

This section describes how you can install and configure Oracle Registry Repository Exchange Utility.

This section contains the following topics:

- [Section 21.2.1, "Installing and Configuring the Oracle Registry Repository Exchange Utility"](#)
- [Section 21.2.2, "Installing Oracle Service Registry"](#)
- [Section 21.2.3, "Configuring the Oracle Registry Repository Exchange Utility Configuration File"](#)
- [Section 21.2.4, "Configuring Oracle Enterprise Repository Categorizations in the UDDI Mappings File"](#)
- [Section 21.2.5, "Configuring the tModelKey UDDI Setting"](#)
- [Section 21.2.6, "Understanding the Oracle Registry Repository Exchange Utility's Property File"](#)

### 21.2.1 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:

- [Section 21.2.1.1, "Install the Oracle Registry Repository Exchange Utility"](#)
- [Section 21.2.1.2, "Configure Using the Asset Type Name"](#)

#### 21.2.1.1 Install the Oracle Registry Repository Exchange Utility

When Oracle Enterprise Repository is installed, by default, the Oracle Registry Repository Exchange Utility files are found at

```
<Oracle_HOME>\oer\modules\tools\solutions
```

The Oracle Registry Repository Exchange Utility files for Oracle Enterprise Repository are as follows:

- **12.1.3.0.0-RR-ExchangeUtility.zip:** Contains the package for the Exchange Utility.

You can unzip the `12.1.3.0.0-RR-ExchangeUtility.zip` file into the directory on your file system where Oracle Enterprise Repository is installed, typically `Oracle_HOME\oer`. When the zip file containing the Oracle Registry Repository Exchange Utility is unzipped to your file system, it creates the following structure.

```
orrxu <ExchangeUtility Tool Home>
|
lib
```

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

### 21.2.1.2 Configure Using the Asset Type Name

When a solutions pack is imported to a system that has existing Asset Types with different UUIDs than the pack, then those Asset Types are created and versioned. The `orrxu.properties` file specifies asset types by name, which causes results to be inaccurate. This is because it tries to use the asset types that already existed, and not the ones that were included in the new solutions pack.

In previous versions of Oracle Enterprise Repository, some Asset Types were not created by the solutions pack, but rather manually, which generated a random UUID.

If you have an existing asset with the same name as the datapack, then the work around is to rename the asset in their system, before they import the new datapack.

## 21.2.2 Installing Oracle Service Registry

You can install Oracle Service Registry and acquire the UDDI Inquiry URL such as `http://host:port/registry/uddi/inquiry`.

For more information about installing Oracle Service Registry, see <http://download.oracle.com/otndocs/tech/soa/OSR11gR1ProductDocumentation.pdf>.

## 21.2.3 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. It contains the following topics:

- [Section 21.2.3.1, "Setting the Repository Connection Information"](#)
- [Section 21.2.3.2, "Setting the Registry Connection Information"](#)
- [Section 21.2.3.3, "Setting the Repository Query"](#)
- [Section 21.2.3.4, "Setting the Destination Registries"](#)
- [Section 21.2.3.5, "Setting the Registry Query"](#)
- [Section 21.2.3.6, "Setting the Source Registry"](#)

### 21.2.3.1 Setting the Repository Connection Information

Open the orrxu.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>*****</password>
    //To ensure security, the password must be encrypted.
    //The password encryption tool (encrypt.bat/encrypt.sh), which is located in
    <Oracle_home>/oer/modules/tools/solutions/12.1.3.0.0-OER-PasswordTools.zip,
    allows you to encrypt
    // the passwords that are stored in the Oracle Registry Repository Exchange
    Utility configuration (orrxu.xml) file.
  </credentials>
</repository>
```

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

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

---

**Note:** It is recommended that you run the Exchange Utility as a user with the Basic Access Settings for Assets - View, Edit, Accept, and Register.

---

For more information about encrypting passwords, see [Chapter 12, "Password Encryption"](#).

### 21.2.3.2 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 or more <registry> nodes with the connection info, as shown below:

```
<registries>
  <registry name="osr">
    <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>*****</password>
      //To ensure security, the password must be encrypted.
      //The password encryption tool (encrypt.bat/encrypt.sh), which is located in
      <Oracle_home>/oer/modules/tools/solutions/12.1.3.0.0-OER-PasswordTools.zip,
      allows you to encrypt
      // the passwords that are stored in the Oracle Registry Repository Exchange
      Utility configuration (orrxu.xml) file.
    </credentials>
  </registry>
  <registry name="osr2">
    <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>*****</password>
        //To ensure security, the password must be encrypted.
        //The password encryption tool (encrypt.bat/encrypt.sh), which is located in
<Oracle_home>/oer/modules/tools/solutions/12.1.3.0.0-OER-PasswordTools.zip,
allows you to encrypt
// the passwords that are stored in the Oracle Registry Repository Exchange
Utility configuration (orrxu.xml) file.
    </credentials>
</registry>
</registries>

```

### 21.2.3.3 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. Some of the ways that services can be queried are:

- [Section 21.2.3.3.1, "Query by Name"](#)
- [Section 21.2.3.3.2, "Query by Registration Status"](#)
- [Section 21.2.3.3.3, "Query by Categorizations"](#)
- [Section 21.2.3.3.4, "Query by Filters"](#)

**21.2.3.3.1 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="" />
  </repositoryQuery>
</query>

```

The Query by Name also supports wildcard char % which finds the matching Service name, for e.g. if the query has `<Service name="Hello%">`, the query would find all the services whose names start with "Hello". e.g. `<Service name="%">` would return all the services in OER.

**21.2.3.3.2 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 are published to Oracle Service Registry, while ignoring all other matched services that are not in this state.

```

  <registrationStatus>Registered</registrationStatus>

```

The acceptable values for registration status, which are not case sensitive, are as follows:

- registered
- rejected



- under\_review
- pending\_review
- submitted
- unsubmitted
- undefined

**21.2.3.3.3 Query by Categorizations** When the `<serviceCategorizations>` element is configured, only the services with the specified categorization is published. For example, when using the following categorization, only the Released services are published to Oracle Service Registry.

```
<serviceCategorizations type="AssetLifecycleStage" value="Stage 4 - Release"/>
```

---

**Note:** The serviceCategorization **type** supports all categorizations found under the taxonomy of the service asset. For example:

```
<serviceCategorizations type="Classification" value="Approved"/>
<serviceCategorizations type="AssetFunction" value="Governance"/>
```

---

**21.2.3.3.4 Query by Filters** You may want to limit the kinds of services that are promoted to a UDDI registry from Oracle Enterprise Repository. The Filters element further narrows the selection based on the type of service besides registration status, service categorizations, and endpoint asset lifecycle. When configured, services that meet the specified criteria are the only services published. Each `<filter>` element has three attributes: type, exclude, and value.

- **Type** (required field): There are two types of filters: metadata and wsdllocation.
  - **metadata:** In Oracle Enterprise Repository, a service asset has several types of metadata. This data can be filtered based on the metadata type name and the xpath for the field.

An example is Service Type. This is a piece of metadata applied to all services created by the Harvester. The metadata type is internal.introspector and the xpath for the field is `/sync/Service_Type`.

The format for the filter is as follows (prefixed by "metadata."):

```
<filter type="metadata.internal.introspector.store/sync/Service_Type" value="Proxy Service" />
```

- **wsdllocation:** When harvesting services into Oracle Enterprise Repository, the WSDL location can be stored either on a remote server or as a local file information in Oracle Enterprise Repository.

---

**Note:** When a service with a local file information is published, information on that WSDL will not be added to Oracle Service Registry, only metadata of the service.

---

- **Value** (required field): Each type has its own set of possible values. This value is what is used to determine which services to filter on.
  - **metadata values:** Accepted values depend on the metadata type being filtered. For example:

**Service\_Type:** Proxy Service or Split-Join Service

```
<filter type="metadata.internal.introspector.store/sync/Service_Type" value="Proxy Service" />
```

**Scope:** global or local

```
<filter type="metadata.internal.introspector.store/sync/Scope" value="global" />
```

Local indicates services that are not consumable.

**Deployment\_Status:** run-time or design-time

```
<filter type="metadata.internal.introspector.store/sync/Deployment_Status" value="run-time" />
```

- **wsdllocation values:** remote, local. The following will publish services with a remote WSDL.

```
<filter type="wsdllocation" value="remote" />
```

- **Exclude** (optional field): The exclude flag, if set to the value of true, publishes all services except those that match the criteria in the filter. The following publishes all services except those with a WSDL stored in a local file information.

```
<filter exclude="true" type="wsdllocation" value="local" />
```

If the Exclude flag is set, then there can be multiple filters of the same type. This means that a search can be performed where a service is of the type Business Service or Split-Join Service. However, if the Exclude flag is not set, then multiple filters of the same type will not work, meaning a search cannot be performed where a service is of the type Business Service and Split-Join Service.

You can only combine multiple FILTER criteria when you are excluding the criteria from the search results. The EXCLUDE Business Service and the EXCLUDE proxy service will work. However, the INCLUDE business service and the INCLUDE proxy service will not work.

For example,

```
<filter type="metadata.internal.introspector.store/sync/Service_Type" value="Business Service" />
or
<filter exclude="true" type="metadata.internal.introspector.store/sync/Service_Type" value="Business Service" />
<filter exclude="true" type="metadata.internal.introspector.store/sync/Service_Type" value="Proxy Service" />
```

The following example will not work:

```
<filter type="metadata.internal.introspector.store/sync/Service_Type" value="Business Service" />
<filter type="metadata.internal.introspector.store/sync/Service_Type" value="Proxy Service" />
```

### Example 1

The following example is the default configuration for Exchange Utility. It publishes services that are not of the service type Proxy Service from Oracle Service Bus and only services that have remote WSDLs.

```
<repositoryQuery>
  <services>
    <service name=" " />
  </services>
```

```

<!--Search criteria for the registration status of the service in
Oracle Enterprise Repository -->
  <registrationStatus></registrationStatus>
  <!--Search criteria for a categorization assigned to the service in
Oracle Enterprise Repository -->
  <serviceCategorizations type="AssetLifecycleStage" value=""/>
  <!--Name of UDDI Registries to publish to. This name corresponds with
UDDI Registry definitions -->
  <!--below under the <registries> section -->
  <destinationRegistries>
  <destinationRegistry name="TEST_UDDI">
  <endpointAssetLifecycleStatus></endpointAssetLifecycleStatus>
  </destinationRegistry>
  </destinationRegistries>
  <!--Filter: identify metadata to apply filter on services to
publish-->
  <filters>
  <filter type="wsdllocation" value="remote" />
  <filter type="metadata.internal.introspector.store/sync/Scope"
value="global" />
  <filter exclude="true"
type="metadata.internal.introspector.store/sync/Service_Type" value="Proxy
Service" />
  </filters>
</repositoryQuery>

```

**Example 2**

The following will publish services with both remote and locally stored WSDLs and only services of the type Proxy Service.

```

<filters>
<filter type="metadata.internal.introspector.store/sync/Service_Type" value="Proxy
Service" />
</filters>

```

**Example 3**

The following publishes only Web services and exclude Business, Proxy, and Split-Join services:

```

<filters>
<filter type="metadata.internal.introspector.store/sync/Service_Type" value="Web
Service" />
</filters>

```

**21.2.3.4 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. Each destination registry may contain an `endpointAssetLifecycleStatus` property. This serves to filter service endpoints of a certain Asset Lifecycle Stage Categorization type to be published to that registry, which means only endpoints that possess the given Asset Lifecycle Stage Categorization are published to the registry that is defined as such. The `endpointAssetLifecycleStatus` property is an optional property.

These registries are used when services are picked from Oracle Enterprise Repository and are moved to Oracle Service Registry (for example, Oracle Enterprise Repository > Oracle Service Registry).

```
<services>
  <service name="%"/>
</services>
<destinationRegistries>
<destinationRegistry name="DEV">
<endpointAssetLifecycleStatus>Stage3- Build</endpointAssetLifecycleStatus>
</destinationRegistry>
<destinationRegistry name="PROD">
<endpointAssetLifecycleStatus>Stage4-Release</endpointAssetLifecycleStatus>
</destinationRegistry>
</destinationRegistries>
```

Modify the destination registries and add the `endpointLifecycleStatus` element to them.

### 21.2.3.5 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"/>
  </businessEntities>
  <services>
    <service name="AddCustomerService%" />
  </services>
  <qualifiers>
    <qualifier>approximateMatch</qualifier>
  </qualifiers>
  <sourceRegistry>osr</sourceRegistry>
</registryQuery>
```

Follow these configuration guidelines:

- Ensure that the `businessEntities` name or `service name` value is not empty.
- For the `businessEntities` name, the exact name must be specified.
- 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 ways:

- 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" is fetched and placed in Oracle Enterprise Repository.
- Select one or more Business Entity and all the services in those Business Entities are 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.**

---



---

### 21.2.3.6 Setting the Source Registry

The <sourceRegistry> element tells the registry where the Services are 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 (for example, Oracle Service Registry > Oracle Enterprise Repository).

```
<sourceRegistry>osr</sourceRegistry>
```

## 21.2.4 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, which 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" />
      <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" />
    </categorizationType>
  </categorizationMappings>
</uddi:uddiSettings>
```

An Oracle Enterprise Repository Categorization is 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.

## 21.2.5 Configuring the tModelKey UDDI Setting

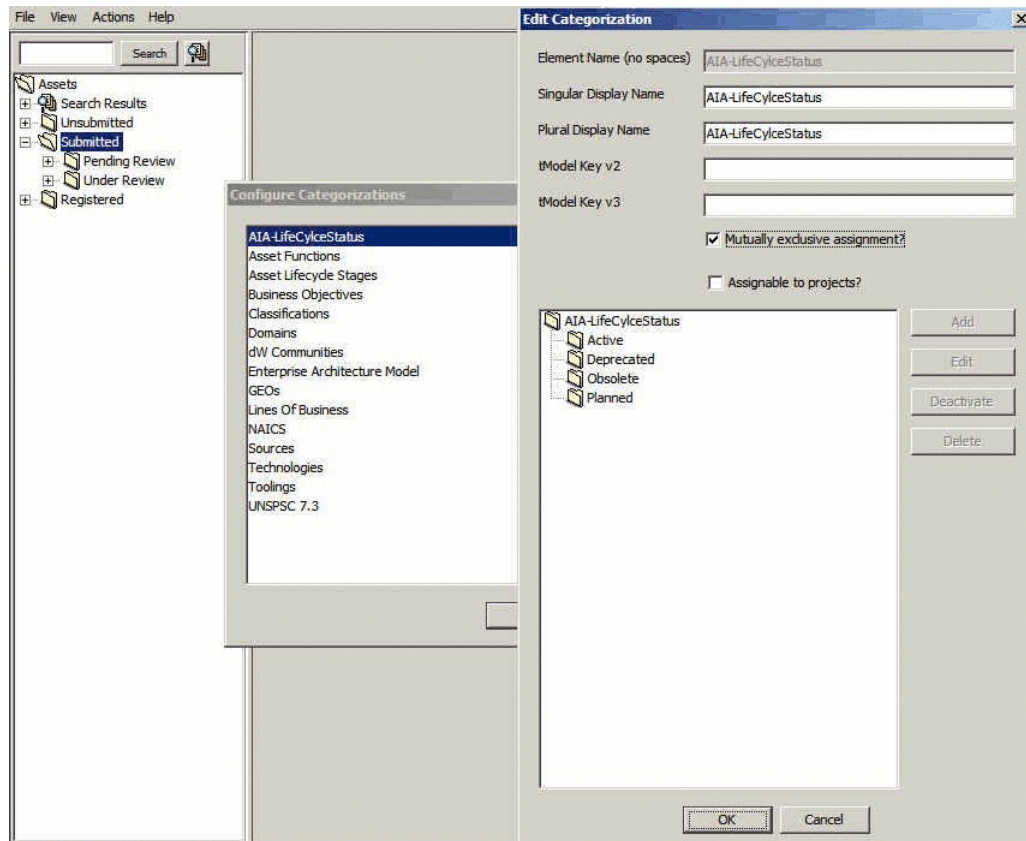
This section describes the steps to configure the tModelKey UDDI setting when creating a new categorization type in Oracle Enterprise Repository.

Some Oracle Service Registry published services contain extra metadata in the form of Categorizations that need to be pulled into Oracle Enterprise Repository. This allows you to create custom categorization types in Oracle Enterprise Repository and bind them to these custom categorizations.

Oracle Registry Repository Exchange Utility now allows binding Oracle Service Registry custom categorizations with Oracle Enterprise Repository custom

categorizations by using the "tModel Key v3" option when creating a categorization type in Oracle Enterprise Repository, as shown in [Figure 21–1](#). The key entered in this field must be the tModel Key of the associated Oracle Service Registry categorization.

**Figure 21–1 The Edit Categorization Dialog**



The steps to configure the tModelKey UDDI setting are as follows:

1. In the Asset Editor window, select **Actions**, and then **Configure Categorizations**. The Configure Categorization dialog is displayed. This creates custom categorization types that mirror Oracle Service Registry categorizations.

---

**Note:** The tModel Key v3 field is set to `uddi:oracle.com:aia:service:lifecyclestatus`, which is exactly what is in the Oracle Service Registry service category's tModel key.

---

2. Click **Add** to add a categorization. The Add Categorization dialog is displayed. For example, add the `AIA-LifeCycleStatus` categorization.
3. Enter the values in the Add Categorization dialog, as shown in [Figure 21–2](#).

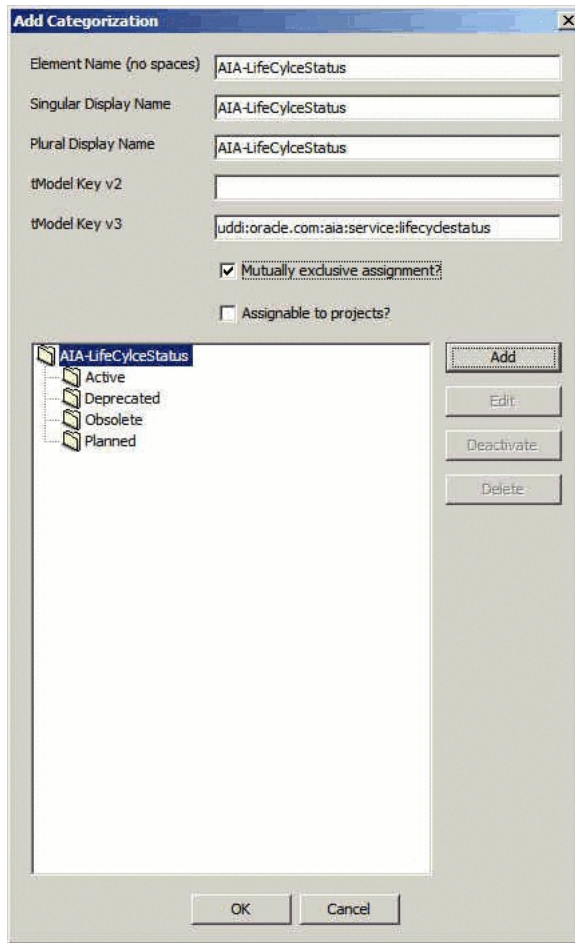
**Figure 21–2 The Add Categorization Dialog**

4. Select **AIA-LifeCycleStatus** in the pane below the tModel Key v3 field, and then click **Add**. The Add Categorization dialog is displayed.
5. Enter Active in the Name field, as shown in [Figure 21–3](#).

**Figure 21–3 The Add Categorization Dialog**

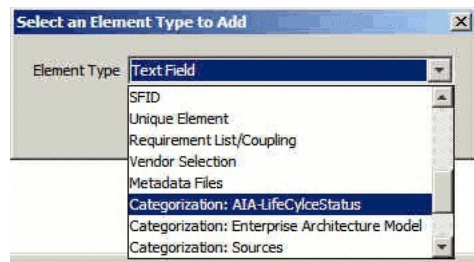
6. Click **OK**.
7. Similarly, add the following values for the categorization: Deprecated, Obsolete, and Planned, as shown in [Figure 21–4](#).

**Figure 21–4 The Add Categorization Dialog**

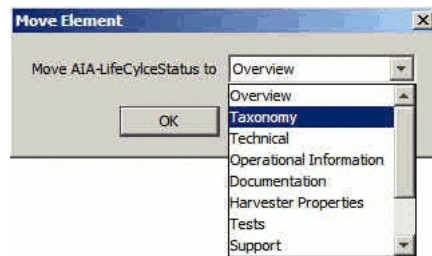


8. Click **OK**.
9. In the Asset Editor window, click **Actions, Manage Types**. The Type Manager window is displayed.
10. Select the **Service** asset type. The Service asset type details are displayed in the right pane.
11. In the Tabs pane, select **Taxonomy**. The corresponding elements for the Taxonomy are displayed in the Elements pane.
12. Click the **Add** button adjacent to the Elements pane. The Select an Element Type to Add dialog is displayed.
13. Select **Categorization: AIA-LifeCycleStatus** from the Element Type list, as shown in [Figure 21–5](#).



**Figure 21–5 The Select An Element Type to Add Dialog**

14. Click **OK**. The Edit Categorization: AIA-LifeCycleStatus dialog is displayed.
15. Click **OK**.
16. Click the **Viewer** tab.
17. In the Hidden Elements pane, select **AIA-LifeCycleStatus** and click **Display In Group**. The Move Element dialog is displayed.
18. Select **Taxonomy** in the Move AIA-LifeCycleStatus to field, as shown in [Figure 21–6](#).

**Figure 21–6 The Move Element to Dialog**

19. In the Exchange Utility tool, generate the **UDDIMappings.xml** file. Specify the connection information to the Oracle Enterprise Repository instance in the orrxu.xml file.
20. From the Exchange Utility tool, install `orrxu.bat -map`.
21. Verify that the UDDIMappings.xml file contains the newly created Oracle Enterprise Repository categorization:

**Figure 21–7 UDDIMappings.xml File**

```
<categoryType alerCategorizationTypeName="AIA-LifecycleStatus" alerCategorizationTypeId="50000"
  uddiCategoryTModelKey="uddi:oracle.com:aia:service:LifecycleStatus">
  <category alerCategorization="Active" uddiKeyName="LifecycleStatus" uddiKeyValue="Active"/>
  <category alerCategorization="Deprecated" uddiKeyName="LifecycleStatus" uddiKeyValue="Deprecated"/>
  <category alerCategorization="Obsolete" uddiKeyName="LifecycleStatus" uddiKeyValue="Obsolete"/>
  <category alerCategorization="Planned" uddiKeyName="LifecycleStatus" uddiKeyValue="Planned"/>
</categoryType>
```

When Oracle Service Registry services are received into Oracle Enterprise Repository, the Oracle Enterprise Repository service asset now has this categorization value set and is visible in the Oracle Enterprise Repository web UI.

## 21.2.6 Understanding the Oracle Registry Repository Exchange Utility's Property File

This section describes the properties in the Property file, `orrxu.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` - 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.
- `cmee.import.uddi.publish.ifendpointmissing` : By default, if a service in Oracle Enterprise Repository does not have one or more endpoints or the existing endpoints have invalid access points, then Exchange Utility will not publish it. If this setting is set to true, then services with no endpoints are published to the Oracle Service Registry.

---

---

**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 `orrxu.properties` file.

---

---

- `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 [Chapter 2, "System Settings Overview"](#).

## 21.2.7 XU Support

By default, XU supports the publication of WSDL-based HTTP Services to OSR that originate in OER from harvesting operations using the asset types provided in the Harvester solution pack. These services, together with their interfaces, artifacts, endpoints and system-supplied relationships, are rendered in OSR in a way that allows them to be consumed by tools that integrate with OSR, including the Oracle Service Bus console.

---

---

**Note:** All services are consumable by Oracle Service Bus in Oracle Service Registry unless otherwise noted.

---

---

**Table 21–1 XU Support for Harvested WSDL Services**

Service Type/Origin	Service Model in OER Includes	XU Publishes from OER to OSR?	Notes
Harvested WSDL-based HTTP Services with WSDL attachment	Artifact: WSDL attached to interface	Yes, if WSDL has remote URI	Services do not require an endpoint in order to be published, however they need the endpoint in order to be consumed by OSB.

### 21.2.7.1 XU Support for Manually-Created Services

Configured differently, XU also supports a wide variety of non-WSDL HTTP service types in OER originating from different sources and with different models.

**Table 21–2 XU Support for Manually-Created\* Services**

Service Type/Origin	Service Model in OER Includes	XU Publishes from OER to OSR?	Notes
Manually-created WSDL-based HTTP Services with attachment (no harvesting)	Artifact: WSDL manually attached to Interface	Yes, if WSDL has remote URI	Different asset and relationships can be used and mapped in XU mapping file, if the MODEL is consistent with SOA model that has Service, interface, endpoint, Artifact: WSDL
Manually-created non-WSDL-based HTTP Services with XSD attachment	Artifact: XSD or similar Artifact manually attached to interface via relationship mapped in XU. No interface type defined.	Yes, if XSD has remote URI; XSD is published as tModel: uddi-org:resource:type with value=xsd	<ul style="list-style-type: none"> <li>▪ OSB will not consume the XSD</li> <li>▪ Published as Messaging Service Type with Input XML/Output XML</li> </ul>
Manually-created non-WSDL HTTP Services with artifact (e.g. REST with WADL)	Different type of Artifact (non-XSD, e.g. WADL) attached to Interface via relationship mapped in XU	Yes, as long as artifact has remote URI; artifact is published as tModel: uddi-org:resource:type with value=xml	<ul style="list-style-type: none"> <li>▪ OSB will not consume the XSD</li> <li>▪ Published as Messaging Service Type with Input XML/Output XML</li> </ul>
Manually-created non-WSDL HTTP Services without artifact (e.g. REST)	No attached XSD or similar Artifact	Yes	Published as Messaging Service Type with Input XML/Output XML

**Note:** Services that are manually created in OER (not harvested) must, at a minimum, have an endpoint and an interface in order to be published to OSR in a form that is consumable by Oracle Service Bus.

\*Also applies to services that are created using the REX API

### 21.2.7.2 XU Support for Non-WSDL Services Harvested from Oracle Service Bus

Configured differently, XU also supports a wide variety of non-WSDL HTTP service types in OER originating from different sources and with different models.

**Table 21–3 XU Support for Non-WSDL Services Harvested from Oracle Service Bus**

Service Type/Origin	Service Model in OER Includes	XU Publishes from OER to OSR?	Notes
OSB-Harvested WSDL-based HTTP Services with WSDL attachment	Artifact: WSDL attached to Interface	Yes, if WSDL has remote URI	Similar to WSDL services harvested by OER Harvester.
OSB-Harvested non-WSDL-based HTTP Services	Any XML (not REST)	Yes	
OSB-Harvested non-WSDL-based HTTP Services	Any SOAP (not REST)	Yes	
OSB-Harvested non-WSDL-based HTTP Services	Messaging with MFL only; Artifact:MFL attached	No	<ul style="list-style-type: none"> <li>▪ Not consumable by OSB in OSR.</li> <li>▪ MFL artifacts do not have a remote URI in OER.</li> </ul>
OSB-Harvested non-WSDL-based HTTP Services (including REST Proxy services)	All Messaging except MFL; may or may not have an Artifact:XSD	Yes, however XSD artifacts are not published without a remote URI	
OSB-Harvested REST HTTP business services	No Artifact:XSD or similar Artifact attached to interface		Not consumable by OSB in OSR.

---

**Note:** Services must, at a minimum, have an endpoint and an interface in order to be published to OSR in a form that is consumable by Oracle Service Bus.

---

Only one type of XU-mapped relationship (e.g. between interface and Artifact) can be used in the publication of non-WSDL or manually-created services in any given XU session. This is concurrently supported with harvested WSDL-based services. XU can publish non-WSDL services with a single designated system-supplied relationship (either system supplied or manually-supplied) but not more than one type at once.

### 21.2.7.3 Configuring XU to Publish Non-Harvested or Custom Asset Types and Relationships

XU defines what asset types and relationships make up a service and its endpoints. For example, the OER Service asset type defines the UDDI Business Service. The Endpoint asset defines the access point, and the relationship, Deployed-To, relates the two asset types together. With these default settings, services created by the Harvester are published to OSR.

However, if you create services without the Harvester or use your own asset types and relationships which differ from the default configuration, you must follow these configuration steps, in order for the services to be published to OSR, such as non-WSDL-based HTTP services harvested from OSB .

Only one service asset type can be published at a time. When using asset types and relationships other than the ones defined by default:

- Change the system setting in OER, **`cmee.import.uddi.service.assettype`**, to point to the custom asset type. By default, it points to Service, but it can point to any asset type desired.
- Modify the `orrxu.properties`:
  - **`cmee.import.uddi.accesspoint.assettype=<custom endpoint asset type>`**

#### For WSDL Services

- **`cmee.import.uddi.artifactwsdl.assettype=<custom artifact WSDL asset type>`**

---

**Note:** This custom artifact WSDL type NAME must begin with **Artifact**.

---

#### For Non-WSDL Services

- **`cmee.import.uddi.definingartifact.relationship=<custom artifact asset type>`**
- **`cmee.import.uddi.artifactwsdl.relationship=<custom relationship between endpoint/interface and artifact asset>`**
- **`cmee.import.uddi.interface.service.relationship=<custom relationship between service and interface assets>`**
- **`cmee.uddi.service.endpoint.relationship=<custom relationship between service and endpoint>`**

#### Non-Harvested Services

- In the Type Manager, select the asset type being used to define the access point (Endpoint out-of-the-box) and add the following Custom Data URL element: **endpoint-uri**.
- This endpoint-uri field must be manually populated with the access point of the service to be published to the UDDI Registry.

#### *Example 21–1 Service definition in the WSDL*

```
<service name="OERSearchService">
<port name="OERSearchPort" binding="typens:OERSearchBinding">
<soap:address location="http://api.oracle.com/search/oer"/>
</port></service>
```

## 21.3 Using the Oracle Enterprise Repository Exchange Utility

This section describes how you can use the Oracle Registry Repository Exchange Utility to synchronize metadata and search for exchanged metadata in Oracle Enterprise Repository.

This section contains the following topics:

- [Section 21.3.1, "Running the Oracle Registry Repository Exchange Utility"](#)
- [Section 21.3.2, "How the Exchanged Metadata Is Synchronized"](#)
- [Section 21.3.3, "Searching for Oracle Service Registry Exchanged Metadata in Oracle Enterprise Repository"](#)
- [Section 21.3.4, "Checking the Oracle Registry Repository Exchange Utility Log File"](#)
- [Section 21.3.5, "Known Issues"](#)

## 21.3.1 Running the Oracle Registry Repository Exchange Utility

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

```
> orrxu.bat <options>
```

Table 21–4 defines the configuration options available when running the Oracle Registry Repository Exchange Utility.

**Table 21–4 Oracle Registry Repository Exchange Utility Command-line Options**

Option	Required	What it Does
-mode <mode>	Yes	Configures the utility to run in either publish or receive mode. <ul style="list-style-type: none"> <li>■ To publish Services to Oracle Service Registry from Oracle Enterprise Repository, use:  <code>orrxu.bat -mode publish</code></li> <li>■ To receive Services from Oracle Service Registry into Oracle Enterprise Repository, use:  <code>orrxu.bat -mode receive</code></li> </ul>
-config <File Name>	Optional	Passes the <code>orrxu.xml</code> file as a parameter. Example usage: <code>orrxu.bat -mode publish -config C:\rm\uddi\orrxu.xml</code> If this parameter is omitted, the configuration XML file is loaded from the current directory where the <code>orrxu.bat</code> file is located using the system's Classpath.
-map <Dir Name>	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>orrxu.bat -map c:/rm/uddi</code> For more information about the <code>UDDIMappings.xml</code> file, see <a href="#">Section 21.2.4, "Configuring Oracle Enterprise Repository Categorizations in the UDDI Mappings File"</a> .
-publish_tmodel <File Name>	Optional	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 is 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. Example usage: <code>orrxu.bat -publish_tmodel c:/rm/UDDIMappings.xml</code> For more information about the <code>UDDIMappings.xml</code> file, see <a href="#">Section 21.2.4, "Configuring Oracle Enterprise Repository Categorizations in the UDDI Mappings File"</a> .

### 21.3.1.1 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.

This section contains the following topics:

- [Section 21.3.1.1.1, "Prerequisites"](#)
- [Section 21.3.1.1.2, "Synchronizing Using Timer Based Workflows"](#)
- [Section 21.3.1.1.3, "Synchronizing When Events are Triggered"](#)
- [Section 21.3.1.1.4, "Example Use Cases"](#)

For more information about the `workflow.xml` file and configuring workflows, see [Chapter 13, "Configuring Oracle Enterprise Repository Express Workflows"](#).

#### 21.3.1.1.1 Prerequisites

This section describes the prerequisite configuration steps required prior to running Oracle Registry Repository Exchange Utility workflows:

1. Download the Oracle Registry Repository Exchange Utility solution pack, `12.1.3.0.0-RR-ExchangeUtility.zip`, from `<Oracle_HOME>\oer\modules\tools\solutions`.
2. Extract the zip file and copy the following files from it to `<Oracle_Home>\obpm\enterprise\server\aler_engine` directory:
  - `plugins`
  - `orrxu.properties`
  - `orrxu.xml`
  - `log4fl.properties`
  - `types.properties`
  - `UDDIMappings.xml`

These files are required to run the Oracle Registry Repository Exchange Utility workflows from Oracle BPM.

3. Configure `orrxu.xml` for the connection information of Oracle Enterprise Repository instance that generates events from and for the Oracle Service Registry instance, and also for publishing/receiving assets.
4. Ensure that the passwords in the files are encrypted using the `encrypt` tool.

#### 21.3.1.1.2 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 21–5](#) shows the names and descriptions of the timer based workflows:

**Table 21–5 Timer Based Workflows**

Workflow Name	Description
<code>autoSyncAlerToUddi</code>	Moves services from Oracle Enterprise Repository to Oracle Service Registry.

**Table 21–5 (Cont.) Timer Based Workflows**

Workflow Name	Description
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="AlerToUddiSyncOrrxuConfig.xml"
    mappingFileName="AlerToUddiSyncOrrxuMapping.xml" timerInterval="2d"/>
  <autoSyncUddiToAler configFileName="UddiToAlerSyncOrrxuConfig.xml"
    mappingFileName="UddiToAlerSyncOrrxuMapping.xml" timerInterval="3d"/>
</automation>
```

### 21.3.1.1.3 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>my_orrxu.xml</alrrxConfigFileName>
<alrrxMappingFileName>my_uddiMappings.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.

---

**Note:** The files specified in `<alrrxConfigFileName>` and `<alrrxMappingFileName>` are to be present in the `aler_engine` folder in `obpm`.

---

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.

### 21.3.1.1.4 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 is processed by the workflows. The workflows will move this to Oracle Service Registry automatically, if it is configured to do so.

## 21.3.2 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. This section contains the following topics:

- [Section 21.3.2.1, "Synchronizing the Metadata Published from Oracle Enterprise Repository to Oracle Service Registry"](#)
- [Section 21.3.2.2, "Synchronizing the Metadata from Oracle Service Registry to Oracle Enterprise Repository"](#)

### 21.3.2.1 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. It contains the following topics:

- [Section 21.3.2.1.1, "Business Entities"](#)
- [Section 21.3.2.1.2, "Service Keys"](#)
- [Section 21.3.2.1.3, "Endpoint"](#)
- [Section 21.3.2.1.4, "Categorizations"](#)
- [Section 21.3.2.1.5, "Registration and Active Status"](#)
- [Section 21.3.2.1.6, "Sample Flow of Metadata 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 [Section 21.3.5, "Known Issues"](#).

---

**21.3.2.1.1 Business Entities** Check if the Service asset being published has a related (by configured relation) Business Entity asset:

- If yes, use the existing Business Entity asset to relate 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 `orrxu.properties` file.

**21.3.2.1.2 Service Keys** When you publish services to Oracle Service Registry, the Exchange Utility automatically generates a new service key and assigns it to the service in Oracle Enterprise Repository as well as Oracle Service Registry. You can also supply your own service key on a service through the Asset Editor.

---

---

**Note:** When supplying your own service key, it must contain only alpha-numeric characters, plus `:', '!', '%', '_',` and `'-`. When the service is published to a UDDI Registry, the service key is converted to an `org.apache.axis.types.URI` object. Therefore, it must conform to the URI syntax.

---

---

For Types that have the UDDI Plugin applied, you could use the Service Key field, which is located in the Technical tab under the UDDI Service section. If this field is populated, then the service key is synchronized to Oracle Service Registry. However, once this service is published, this field will no longer be editable.

Oracle Enterprise Repository determines the published status by the existence of UDDI Registries that the service has been published to.

---

---

**Note:** If the UDDI Registry tables are cleared out from all Endpoints or the Service using the Delete feature, the Service Key becomes writable. However, changing the service key and re-publishing to a UDDI Registry can cause the service keys to be out of sync. This is not supported. If the Service Key is changed, the corresponding Services in the UDDI Registry should be deleted and republished.

---

---

To see this, a table displaying the UDDI Registries is available to the Endpoints that belong to the Service. This table can be found in the Technical tab. However, if the Service is defined by a WSDL that is stored locally in Oracle Enterprise Repository, the Endpoints will not be published to the UDDI Registry. Due to this, the table can be found in the Service asset within the Technical tab.

If the Exchange Utility configuration is not using the Endpoint asset type, then you must add the UDDI Registries plugin manually. To do this, go to the Type Manager and select the asset type that is being used. Select the desired tab and under Elements, select **Add**. From the list, select **UDDI Registries**. Also, ensure that you add this plugin to the Viewer tab.

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

- If yes, then 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 the WSDL is stored locally in Oracle Enterprise Repository, the existing binding template in Oracle Service Registry will not be updated.
- If no, then 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. The Binding

Templates are created only for WSDLs that are located remotely (<http://>). WSDLs that are stored locally in Oracle Enterprise Repository are not included.

- If Asset Life cycle is attached to the endpoint and if the Endpoint Asset Life cycle Query is used, then 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.

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

- If yes, then load the Categorization mapping for each of the applied Categorizations from the `UDDIMappings.xml` file.
  - If the mapping is found, then 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, then automatically create the t-models.
- If no, then the Categorization will not be applied to the Service in Oracle Service Registry.

**21.3.2.1.5 Registration and Active Status** The Registration and Active Status are added to Category Bags. [Figure 21–8](#) illustrates how references appear in Oracle Service Registry.

**Figure 21–8 Oracle Service Registry t-model Categories for a WSDL Service**

tModel	key name	key value
<a href="#">uddi-org:wsdl:types</a>	uddi.org:wsdl:types	service
<a href="#">uddi-org:xml:localName</a>	uddi.org:xml:localName	GoogleSearchService
<a href="#">uddi-org:xml:namespace</a>	uddi.org:xml:namespace	urn:GoogleSearch
<a href="#">bea-com:aler:uuid</a>	bea-com:aler:uuid	d8ec9419-f3d5-11de-bb14-e34b7860e65a
<a href="#">bea-com:aler:ActiveStatus</a>		Active
<a href="#">bea-com:aler:RegistrationStatus</a>		Registered
<a href="#">oracle-com:oer:internalname</a>	oracle-com:oer:internalname	{e2e_App}/{urn:GoogleSearch}/Service/GoogleSearchService
<a href="#">oracle-com:oer:version</a>	oracle-com:oer:version	1.0

**Category groups**  
No category groups found.

Back View as XML Delete Edit

[Figure 21–9](#) illustrates how two endpoints that are linked to a service in Oracle Enterprise Repository appear in Oracle Service Registry.

**Figure 21–9 WSDL Bindings in Oracle Service Registry**

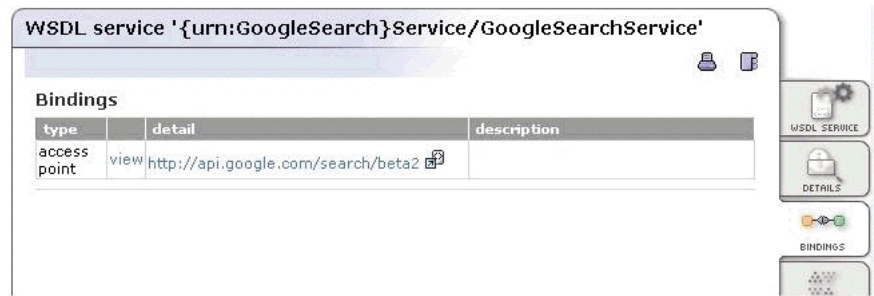
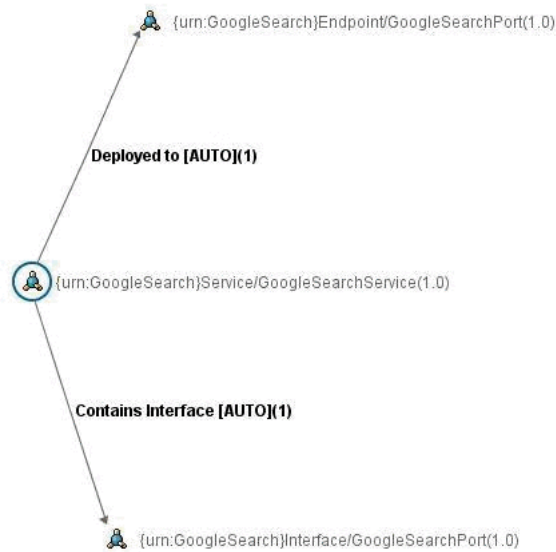
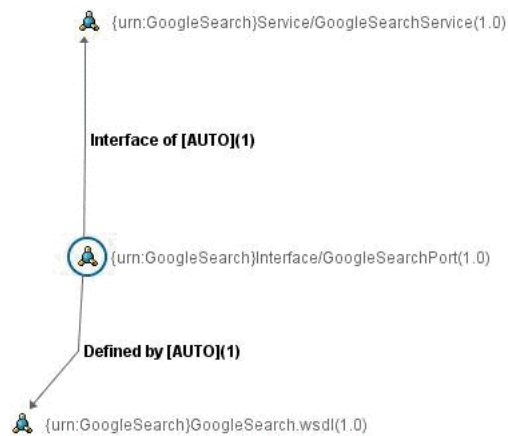


Figure 21–10 and Figure 21–11 illustrate how the different entities and their relationships appear in the Oracle Enterprise Repository Navigator.

**Figure 21–10 Entity Relationship in Oracle Enterprise Repository Navigator**

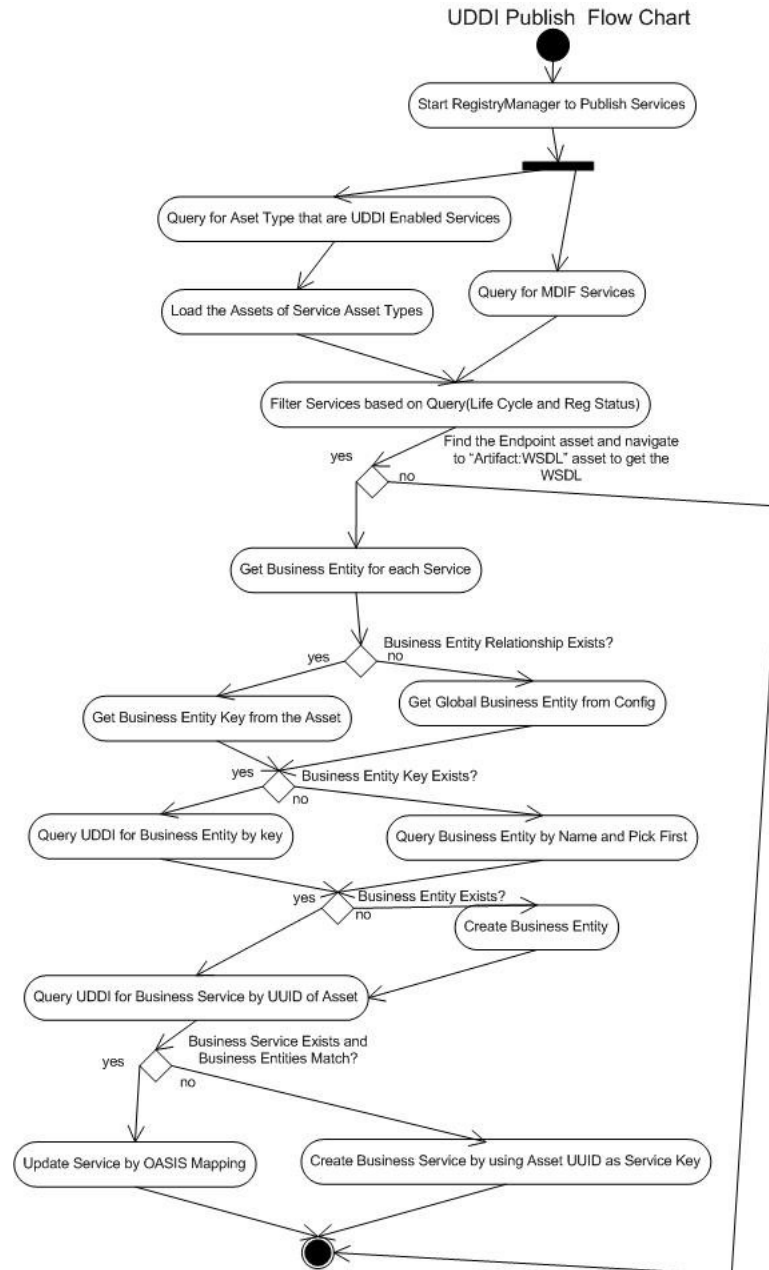


**Figure 21–11 Another Entity Relationship in Oracle Enterprise Repository Navigator**



**21.3.2.1.6 Sample Flow of Metadata from Oracle Enterprise Repository to Oracle Service Registry** Figure 21–12 illustrates the Oracle Enterprise Repository > Oracle Service Registry metadata synchronization described in this section.

**Figure 21–12 Flow of Metadata Published from Oracle Enterprise Repository to Oracle Service Registry**



### 21.3.2.2 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. It contains the following topics:

- [Section 21.3.2.2.1, "Business Entities"](#)
- [Section 21.3.2.2.2, "Endpoint"](#)

- [Section 21.3.2.2.3, "Categorizations"](#)
- [Section 21.3.2.2.4, "Sample Flow of Metadata from Oracle Service Registry to Oracle Enterprise Repository"](#)

**21.3.2.2.1 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 is used.
- If no, the Business Entity on the Oracle Service Registry side is used. If the Business Entity does not exist in Oracle Enterprise Repository, it is 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 `orrxu.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.

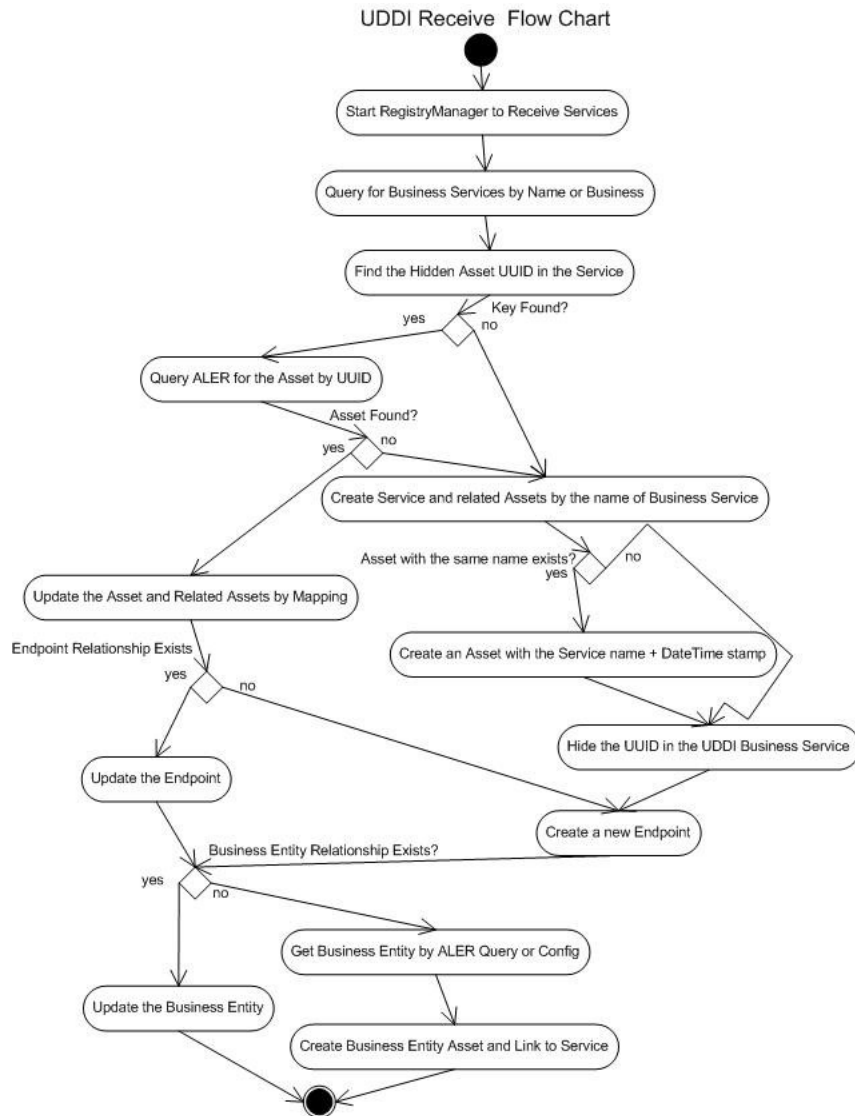
**21.3.2.2.2 Endpoint** Check if the Service asset being received has one or more Endpoint assets related, as follows:

- 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.

**21.3.2.2.3 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.

**21.3.2.2.4 Sample Flow of Metadata from Oracle Service Registry to Oracle Enterprise Repository** [Figure 21–13](#) illustrates the Oracle Service Registry > Oracle Enterprise Repository metadata synchronization described in this section.

**Figure 21–13 Flow of Metadata Received from Oracle Service Registry**

### 21.3.3 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 is displayed.
3. Select the **Filter by Additional Criteria** option to reveal the additional filtering criteria options.

4. In the Select a Field list, 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 21-14](#).

**Figure 21-14 More Search Options Dialog**

Enter Search String:

Type:  Registration Status:

Consumption Method:

Filter by Categorizations

Filter by Additional Criteria

Select a Field:

Enter an XPath: (optional)

Returned assets meet all criteria added to this box.

6. Click the **Search** button at the bottom of the dialog.
- For more information about using the Oracle Enterprise Repository search options, see "Finding Assets" in the *Oracle Fusion Middleware Developer's Guide for Oracle Enterprise Repository*.

### 21.3.4 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 log4fl.properties file located in the <ExchangeUtility Tool Home> directory.

### 21.3.5 Known Issues

This section describes the known issues when using the Oracle Registry Repository Exchange Utility. It contains the following topics:

- [Section 21.3.5.1, "Resynchronizing Oracle Service Registry Services"](#)
- [Section 21.3.5.2, "Publishing Services to Oracle Service Registry"](#)
- [Section 21.3.5.3, "Running an Incorrect Version of Java"](#)



### 21.3.5.1 Resynchronizing Oracle Service Registry Services

When synchronizing a service to Oracle Service Registry that was previously synchronized, there is a 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 and restarted to see the updated values.

### 21.3.5.2 Publishing Services to Oracle Service Registry

1. Once services from Oracle Enterprise Repository are published to Oracle Service Registry, it is possible that you may encounter an error when trying to open the WSDL Location URL of the service in Oracle Service Registry.

**Workaround:**

Copy and paste the URL of the WSDL location to the same browser instance where Oracle Enterprise Repository console is logged into.

2. When the Oracle Enterprise Repository Registry Exchange Utility was used to publish a service to the Oracle Service Registry, a new service key was generated for the service in the Service Registry. Now, service keys are kept consistent between the service asset and all Service Registry instances. Since, Oracle Enterprise Repository Exchange Utility supports publishing service endpoints to multiple registry at the same time, the service key is kept consistent for a service across all registries.

**Workaround:**

None.

3. When an Oracle Enterprise Repository service that does not contain a service key is submitted to Oracle Service Registry, a service key is created from the service asset's UUID. This service key is then maintained across all registries when this service is published to multiple registries.

**Workaround:**

None.

### 21.3.5.3 Running an Incorrect Version of Java

While running Oracle Enterprise Repository Registry Exchange Utility, If you are not using the minimum required JDK 1.6 version, then the following error message is displayed:

```
Exception in thread "main" java.lang.UnsupportedClassVersionError: Bad
version number in .class file
    at java.lang.ClassLoader.defineClass1(Native Method)
    at java.lang.ClassLoader.defineClass(ClassLoader.java:620)
    at
java.security.SecureClassLoader.defineClass(SecureClassLoader.java:124)
@   at java.net.URLClassLoader.defineClass(URLClassLoader.java:260)
@   at java.net.URLClassLoader.access$100(URLClassLoader.java:56)
@   at java.net.URLClassLoader$1.run(URLClassLoader.java:195)
    at java.security.AccessController.doPrivileged(Native Method)
```



# Part V

---

## Appendixes

This part presents appendixes for administering Oracle Enterprise Repository.

This part contains the following appendixes:

- [Appendix A, "Other Configurations"](#)



---

---

## Other Configurations

This appendix describes customization and configuration options not covered previously.

This appendix contains the following topics:

- [Section A.1, "Creating and Configuring an Upload Submission Directory"](#)
- [Section A.2, "Creating an Asset Audit Log"](#)
- [Section A.3, "Oracle Enterprise Repository Default Artifact Store Integration"](#)
- [Section A.4, "Maintaining XSLT Templates for use with the Export-to-PDF Feature"](#)
- [Section A.5, "Support for UTF-8/Unicode Characters"](#)
- [Section A.6, "Data Rules"](#)
- [Section A.7, "Harvester Properties"](#)

### A.1 Creating and Configuring an Upload Submission Directory

You can create and configure an upload submission directory.

#### **Create a Submission Upload Directory**

On the application server, create a directory that contains asset files that are uploaded during asset submission.

#### **Set up a Shared Directory for a Submission Upload Directory**

Using the appropriate procedure for the operating system in use, create a shared directory.

- For Windows this is usually a UNC share.
- In UNIX variants an application makes directories and files available (i.e., Samba, Apache, NFS, etc.).

This section contains the following topics:

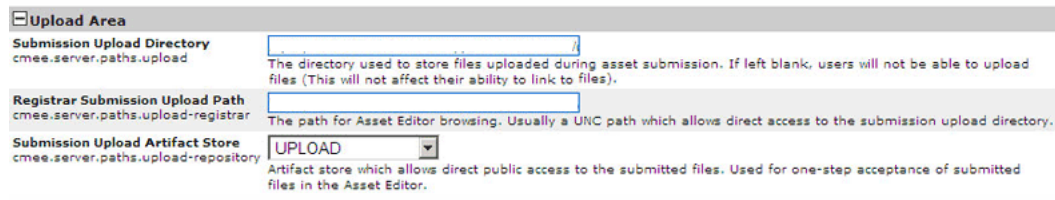
- [Section A.1.1, "Asset Upload System Settings"](#)
- [Section A.1.2, "Creating and Configuring an Artifact Store"](#)
- [Section A.1.3, "Selecting a New Artifact Store"](#)

#### **A.1.1 Asset Upload System Settings**

This procedure is performed on the Oracle Enterprise Repository Admin screen.

1. In the left panel, click **System Settings**.
2. Locate the Upload Area section in the Server Settings group.

**Figure A–1 Upload Area Section**



3. Using the following information as a guide, change the Upload Area settings to reflect your organization and its environment.

### Submission Upload Directory

- **cmee.server.paths.upload**
- Defines the directory used to store files uploaded with an asset submission. If left blank, you cannot upload files. The directory should be placed inside the Oracle Enterprise Repository web app. For example: `<OER_Home>/applications/oeer/oeer-app/custom/upload` where `upload` is a directory created inside the `custom` directory. It is not necessary that the upload directory be a child of a custom directory but it is a recommended practice. This property enables the following features:
  - Enables the Add Attachment Now option in the Asset Submission form. (Required if files are to be attached to assets during the submission process.)
  - Enables Uploaded Submission Files option in the Asset Editor.

### Registrar Submission Upload Path

- **cmee.server.paths.upload-registrar**
- Defines the directory path for registrar browsing, typically a UNC path, which allows direct access to the submission upload directory. The upload folder should have a windows share. For example: `\\<machineName>\<sharename>\upload`. Leave blank if the upload repository is not UNC.
  - Enables file browsing through the Asset Editor (Optional).
  - The path, when hosted by UNC, requires Windows OS or Samba/Shareity.
  - Enables the appearance of the Browse button in Uploaded Submission Files in the Asset Editor.

### Submission Upload Artifact Store

- **cmee.server.paths.upload-repository**
- Specifies a configured artifact store to allow direct public access to the submitted files. Used for one-step acceptance of submitted files in the Asset Editor. When this property is set:
  - Asset files that are available for use/download are visible.
  - The Accept button is visible in the uploaded submission files section of the Asset Editor.

## Proxy Downloads

Oracle Enterprise Repository includes an additional option that allows the Repository application server to act as a repository proxy for the user. This feature automatically removes username and password text that is supplied in the HTTP URL during file downloads. This feature is applicable to any artifact store providing that Oracle Enterprise Repository can have network connectivity to resources in that store. Contact Oracle Enterprise Repository support for details.

## Examples of Artifact Store Configurations

For instructions on configuring one of the following artifact stores, see *Creating and Configuring an Artifact Store*.

- UNC
  - Hostname: `unc.example.com`
  - Path: `upload/`
  - Username: `user_oer`
  - Password: `flash`
  - URL: `file:///unc.example.com/upload/`
- HTTP
  - Hostname: `http.example.com`
  - Path: `oer-web/upload`
  - Username: `user_oer`
  - Password: `flash`
  - URL: `http://user_oer:flash@http.example.com/oer-web/upload/`
- FTP
  - Hostname: `meta.example.com`
  - Path: `upload`
  - Username: `user_oer`
  - Password: `flash`
  - URL: `ftp://user_oer:flash@meta.example.com/upload/`
- ClearCase
  - Hostname: `clearcase.example.com`
  - Path: `ccaseweb/bin/ccweb`
  - Username: `user_oer`
  - Password: `flash`
  - URL:  
`http://clearcase.example.com/ccaseweb/bin/ccweb/?dir=/&elem=&cmd=view&user=user_oer&password=flash`
- PVCS
  - Hostname: `pvcs.example.com`
  - Path: `dav`
  - Username: `user_oer`

- Password: flash
- URL: `http://user_oer:flash@pvcs.example.com/dav/`
- Raw URI
  - Name: Example RAW URI
  - RAW URI: `http://www.example.com/path1/path2?parameter1&parameter2`
  - URL: `http://www.example.com/path1/path2?parameter1&parameter2`

## A.1.2 Creating and Configuring an Artifact Store

An Artifact Store is where the files relevant to assets in Oracle Enterprise Repository are stored. Launch the Asset Editor window from the Oracle Enterprise Repository home page.

See [Section 9.3.2, "Configure Artifact Stores"](#) for more information about configuring artifact stores.

## A.1.3 Selecting a New Artifact Store

After configuring an artifact store as described in [Section 9.3.2, "Configure Artifact Stores"](#), you must select it using the Submission Upload Artifact Store system setting on the Oracle Enterprise Repository Admin page.

See [Section 9.3.3, "Selecting a New Artifact Store"](#) for information about selecting a new artifact store.

## A.2 Creating an Asset Audit Log

With the proper system configuration, XML metadata for individual saved or changed assets can be exported to a text file in order to create an audit trail. The text file lives on the Application Server and is not available for review within the application. Contact Oracle support for the properties necessary to activate this feature.

This procedure is performed on the Oracle Enterprise Repository Admin screen.

1. In the left panel, click **System Settings**. The System Settings section is displayed in the main pane.
2. Enter the property supplied by Oracle Enterprise Repository support in the Enable New System Setting text box.
3. Click the **Enable** button. Asset Save Export Filename appears in the list of properties in the main pane.
4. Enter the name of the file to which asset logs are to be stored in the Asset Save Export Filename text box.
5. Click **Save**.

## A.3 Oracle Enterprise Repository Default Artifact Store Integration

Integrating one of the default artifact stores with your system enables the use of an artifact store in Oracle Enterprise Repository. Typically, a URL used to reach a file using one of the default types resembles the following:

```
http://server.host.com:port/additionalPathStructure/fileName
```

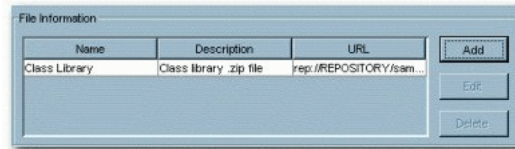


### Adding Files to an Asset

This procedure is performed in the Oracle Enterprise Repository Asset Editor.

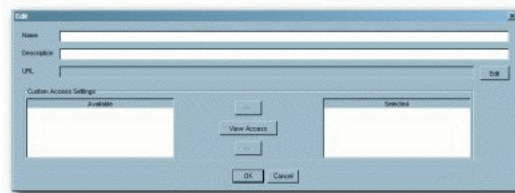
1. Use Search or other means to locate the asset to which a file is added.
2. Locate the File Information element on the Overview tab, as shown in [Figure A-2](#).

**Figure A-2 File Information Section**



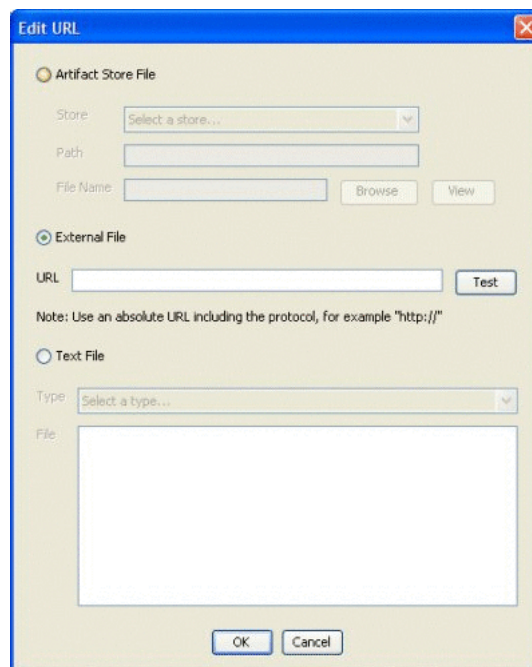
3. Click the Add button. The Edit dialog is displayed, as shown in [Figure A-3](#)

**Figure A-3 Edit Dialog**



4. Fill in the Name and Description text boxes.
5. Click the Edit button (next to the URL field). The Edit URL dialog is displayed, as shown in [Figure A-4](#).

**Figure A-4 Edit URL Dialog**



6. Use the options to select the source of the file to be added.

#### Artifact Store File

- a. Select an item from the Host list. (Configuration determines available selections.)
- b. As necessary:
  - Enter the appropriate `additionalPathStructure/fileName` in the Path text box, or...
  - Click **Browse** to locate the file to be added.
  - This option is available when using Artifact Store types, ClearQuest, File Stores, and PVCS.

#### External File

1. Enter the file URL in the URL text box.
2. Click **Test** to verify that the URL is valid.
7. If necessary, a supplementary text file may be created and added to the file:
  - a. Select the **Text File** option.
  - b. Select the type of text file from the Type list.
  - c. Enter text as necessary in the File text box.
8. When finished, click **OK** to close the Edit URL dialog.
9. Click **OK** to close the Edit dialog. The added file appears in the list in the File Information element.

## A.4 Maintaining XSLT Templates for use with the Export-to-PDF Feature

FO is an XML format that describes paged documents. An FO processor, such as the Apache XML Project's FOP, consumes FO to generate PDF output. The production of a PDF document from an XML document is a two-step process:

1. The XML document is transformed to an FO document using an XSLT (XSL-FO) stylesheet.
2. The FO document is processed to generate PDF (or other output as defined in the processing XSLT stylesheet).

Since the Oracle Enterprise Repository XSLT Printing functionality is applied at the Type level, only one template is needed to print any assets of the corresponding Type. However, there is no limit to the number or scope of the templates that can be created or applied to a specific Type.

### A.4.1 Maintaining XSL:FO Templates for use with Oracle Enterprise Repository

This section describes how to maintain XSL:FO templates for use with Oracle Enterprise Repository. This section contains the following topic:

#### Maintaining the XSL template from an Asset XML export

Prerequisites:

- Oracle Enterprise Repository has been configured for XSLT printing (as described above).
- The process described below requires the use of Altova XML Spy.

1. Configure an instance of Oracle Enterprise Repository with the XSLT Printing functionality as described above.
2. Login to Oracle Enterprise Repository.
3. Use Search or other means to select a registered asset of the Type for which the XSL template is to be created.

---

---

**Note:** Selecting a well-documented asset will facilitate previewing the XSL:FO template.

---

---

4. Create a local work folder that contains the following sub-directories:
  - xml
  - schema
  - xsl
  - sps
5. Transfer the XML document from the `<oer_home>/reports/export` directory to the newly created xml sub-directory.
6. Open that XML document with Altova XML Spy.
7. Select Generate DTD/Schema from the DTD/Schema menu in Altova XML Spy.
8. When prompted, select the following options for the generated schema file:
  - DTD/Schema file format: W3C Schema
  - List of Values: Unlimited
  - Attribute/Element type detection: Best possible
  - Represent complex elements as: Complex types
  - Elements that were used once: Make local definition
9. Click **OK** to generate the schema.
10. Save the resulting XSD file to your schema sub-directory in the work folder.
11. When prompted, "Do you want to assign generated DTD/Schema to your XML document?", Select No.
12. Close Altova XML Spy. The generated XML schema from the previous step contains data element values that must be removed before processing the schema.
13. Use the find-and-replace function in a text editor to remove all lines containing the expression: `<xs:enumeration value=`.
14. Save the changes to the schema file.
15. Exit the text editor.
16. Open the modified schema file with Altova StyleVision.

---

---

**WARNING:** Opening a schema file containing any element related to the value field will cause a StyleVision error.

---

---

17. Use StyleVision to modify the layout of the printed document.  
To view layout modifications:

1. Select Assign Working XML File... in the File menu.
2. Select the exported XML document.

This will generate a preview of the output of the XSL:FO template populated with live information. The most common layout method is one that closely mimics the layout of the asset detail in Oracle Enterprise Repository.
18. When the layout is completed, save the StyleVision formatted document to the sps sub-directory in the local work folder for later retrieval and modification.
19. Open the File menu.
20. Select Save Generated Files.
21. Select Save Generated XSL:FO File...
22. Save the XSL document to the xsl sub-directory in the local work folder.

Use a filename that is appropriate for the Type to which the XSL document is associated.
23. Copy the generated XSL file to the <oer\_home>/reports/xsl folder (as designated in the cmee.asset.xml.paths.xsl-source property listed above).
24. In Oracle Enterprise Repository, select the asset that was exported to generate the original XML schema.
25. Click the **Print** button in the asset detail to export the asset's XML schema. A dialog is displayed.
26. Follow the prompts in the dialog to select the newly created XSL:FO template, which is used to generate the PDF document. The generator will create the PDF document.
27. Click the **View Document** button to open the PDF document.
28. Compare the PDF document to the template design created in StyleVision.
29. Make changes as necessary.

## A.5 Support for UTF-8/Unicode Characters

The Export to PDF feature cannot properly process UTF-8/Unicode characters unless the FOP engine uses ArialUnicode or a similar font that is capable of rendering Unicode.

### Configuration to Render Unicode Characters

1. Locate the appropriate true type font (TTF) file.
2. Run the following command to create the XML definition of the file:

```
java -cp fop.jar org.apache.fop.fonts.apps.TTFReader {TTF_FILE_
LOCATION} {FONT_NAME}.xml
```

---

**Note:** In the command:

- Replace TTF\_FILE\_LOCATION with the location of the TTF file.
  - Replace FONT\_NAME with the name of the font.
-

3. Create a configuration file named `userconfig.xml`. This file will configure FOP to use the selected font. For more information, see <http://www.sagehill.net/docbookxsl/AddFont.html>.
4. Place the `userconfig.xml` file in the classpath. Suggested: The classes directory under `WEB-INF`.
5. Configure the XSLTs to use the new font whenever appropriate.  
This generally requires the addition of `font-family="{FONT_NAME}"=` to the appropriate `fo:block` elements.  
If the `userconfig.xml` file is not on the classpath, the Export to PDF process will replace Unicode characters with octothorpes.

### **Sample XSLTs which work in both Multibyte Character Set and Single Byte Character Set**

- `Application.xml`
- `Business Process.xml`
- `Communication Adapter.xml`
- `Component.xml`
- `Environment.xml`
- `Framework.xml`
- `Pattern.xml`
- `Process.xml`
- `Project Profile.xml`
- `Service.xml`
- `XML Schema.xml`

### **Sample XSLTs which work only in Single Byte Character Set**

- `Artifact_DTD.xml`
- `Artifact_MFL.xml`
- `Artifact_WS-Policy.xml`
- `Artifact_WSDL.xml`
- `Artifact_XQuery.xml`
- `Artifact_XSD.xml`
- `Artifact_XSLT.xml`
- `Assembly Model.xml`
- `Binding.xml`
- `Binding_File Source.xml`
- `Binding_JMS.xml`
- `Binding_Relational Database.xml`
- `Binding_SB-Custom.xml`
- `Binding_SB-DSP.xml`
- `Binding_SB-EJB.xml`

- Binding\_SB-Email.xml
- Binding\_SB-File.xml
- Binding\_SB-FTP.xml
- Binding\_SB-HTTP.xml
- Binding\_SB-JMS.xml
- Binding\_SB-Local.xml
- Binding\_SB-MQNative.xml
- Binding\_SB-Oracle.xml
- Binding\_SB-PeopleSoft.xml
- Binding\_SB-SAP.xml
- Binding\_SB-SB.xml
- Binding\_SB-SFTP.xml
- Binding\_SB-Siebel.xml
- Binding\_SB-Tuxedo.xml
- Binding\_SB-WLI.xml
- Binding\_Web Service.xml
- Component\_Data Service.xml
- Component\_DSP Web Service Mapper.xml
- Component\_Java.xml
- Component\_SB Proxy Service.xml
- Component\_SCA Composite.xml
- EA - Sample Asset Type.xml
- EA - Waiver.xml
- Endpoint.xml
- Interface.xml
- Interface\_Data Service.xml
- Interface\_Java.xml
- Interface\_Messaging.xml
- Interface\_Web Service.xml
- Procedure.xml
- SCA Composite.xml
- SCA Reference.xml
- SCA Service.xml
- SOA - Business Entity.xml
- Training.xml

## A.6 Data Rules

This section contains the following topics:

- [Section A.6.1, "Flags on Asset Type Elements"](#)
- [Section A.6.2, "Grandfathered Data"](#)
- [Section A.6.3, "Element Storage"](#)
- [Section A.6.4, "Elements Required to Exist on Asset Types"](#)
- [Section A.6.5, "Elements Tied to Functionality"](#)

### A.6.1 Flags on Asset Type Elements

Many elements, which can be added to Asset Types, have a pair of flags to enhance Oracle Enterprise Repository functionality. These flags are toggled in the Type Manager when Adding or Editing an element.

**Required Flag** - These are elements, which are expected to be populated, for the Asset Editor to save the asset. The flag does not restrict save through the Open API.

**Indexed Flag** - These are elements, which are available for Advanced Searches on Assets. It is most meaningful for assets, which are stored in Custom Data.

### A.6.2 Grandfathered Data

There are several operations in Oracle Enterprise Repository, which affect the types in the repository and future operations on assets of those types, but which do not affect metadata already associated with assets.

- Elements removed from an asset type

If an element is removed from an asset type, assets that have values specified for that element keep those values in the database and continue to persist those values with future saves of the asset. The values will not appear to the UI, but are available through API operations, custom reporting, and restoration if the element is readded to the asset type.
- Data values in elements which are restricted by Acceptable Value Lists

An Acceptable Value List specifies the values that are legal for when an asset with a list or multiple aelection list is being assigned a value. Oracle Enterprise Repository also considers any value already assigned to the element to be a legal value. When the list is changed, the assets using values from the list are not subsequently saved.

In order to propagate a change in the list of values to assets using those values, an OpenAPI program could be utilized. That program should search for element/value pairs that will return those assets using the removed value. The program should then change those values to something legal and save the assets.

For example, A Traffic Light value list contains the values "Red", "Yellow", "Green". Any new asset will have the values "Red", "Yellow", "Green" available for populate of the Traffic Light list.

There is an existing asset that has the value "Left-Turn". This value was legal at one point in the past, but is no longer part of the Traffic Light value list. For this existing asset, the legal values in the Traffic Light list are "Red", "Yellow", "Green", "Left-Turn".

### A.6.3 Element Storage

Asset metadata in Oracle Enterprise Repository is stored in elements, which belong to one of four different groups. Each of these groups is persisted and behaves in different ways. The four groups are:

- **Mandatory-Data**

These are elements, which must be part of an Asset Type.

Mandatory Data elements are persisted to the database in true relational fashion, with columns holding the data, accessible by an Asset ID Foreign Key. All Mandatory Data elements have an XML Mapping, which begins with "mandatory-data/".

- **Custom-Data**

These are elements, which are suitable for customization and removal. Newly created elements in the Type Manager appear in this section.

Custom Data elements are persisted in a single block of XML, which is stored in the database, as a CLOB. All Custom Data elements have an XML Mapping, which begins with "custom-data/".

- **Admin-Data**

These are the elements, which appear on the Administration tab of the Asset Editor. No customization of these elements is possible. These elements are used for Registration Workflow and Status and Custom Access Settings.

Admin Data elements are persisted to the database in true relational fashion, with columns holding the data, accessible by an Asset ID Foreign Key. This data is modified by firing actions on an asset, for example, AssetRegister().

- **Custom Metadata Framework (CMF) metadata**

The metadata that is persisted by this framework is considered System-Supplied and is protected from modification by the Oracle Enterprise Repository UI. This framework is primarily utilized by integrations, which utilize specific structures of metadata and are often tied to XML Schemas.

CMF metadata elements are persisted as BLOBs in blocks, which conform to the CMF Entry Type. Generally, these blocks are XML.

### A.6.4 Elements Required to Exist on Asset Types

There are several elements, which cannot be removed from any Asset Type:

- **Name**

This element is the name of the asset. The name and version pair must be unique within Oracle Enterprise Repository. It is limited to 128 characters. It is possible to limit a search to this element by using the prefix "name:". In Asset Detail, the name is part of the Asset Header.

- **Version**

This element is the version of the asset. The name and version pair must be unique within Oracle Enterprise Repository. It is limited to 128 characters. In Asset Detail, the version is part of the Asset Header.

- **Description**

This element is the description of the asset. This element can contain HTML and is treated as HTML by the browser when the asset is loaded. It is limited to 4000



characters. It is possible to limit a search to this element by using the prefix "description:" or "desc:".

- **Notification Email**

This element is the email address, which will receive notifications when the asset is used. The email address is a single address, without comma or other separation. If multiple recipients need to be notified, then the email address must be a group, established by an email system and outside of Oracle Enterprise Repository.

- **Keywords**

This element contains the keywords for the asset. These are included in all searches using a typed search string.

- **File Information**

This element contains the URIs to the downloadable files on the asset. This element never appears in Asset Detail.

- **Uploaded Submission Files**

This element contains the list of files, which have been uploaded to Oracle Enterprise Repository and have not yet been accepted into File Information. This element never appears in Asset Detail.

## A.6.5 Elements Tied to Functionality

- **Policy:** The Policy feature requires two elements, one relationship type and two system settings. It uses the Policy archetype in the Type Manager.
  - The Policy (instance of type with archetype Policy Type) requires a plugin to host the list of Policy Assertions. This plugin is "Policy Assertions" (**PolicyAssertionPlugin**).
  - An asset (instance of type with archetype Asset Type) requires a two-way, order matters relationship to be governed by a Policy. The default relationship type is "Policy Applied To".
    - \* The relationship type linking a policy to a governed asset is specified by the `cme.asset-policy.relationship` system setting.
  - The governed asset requires a plugin to allow the True/False response on each Policy Assertions. This plugin is "Applied Asset Policies" (**AppliedAssetPolicyPlugin**).
  - The System Setting "Enable Asset Policies" (`cme.asset-policy.enable`) allows use of the Policy Type archetype, the "Policy Assertions" and the "Applied Asset Policies".
- **Compliance Templates:** This feature requires three relationships and one system setting. It uses the Compliance Template archetype in the Type Manager.
  - Compliance Templates require relationships to prescribe assets for use. The list of required relationship names are:
    - \* "Assets-to-be-Used" [Asset Prescribed by/Prescribed Asset(s)]
    - \* "Required-Environments" [Environment Prescribed by/ Prescribed Environment(s)]
    - \* "Processes-to-be-Followed" [Process Prescribed by/ Prescribed Process(es)]

- The System Setting "**Enable Compliance Templates**" (`cmee.compliance-template.enable`) allows use of the Compliance Template archetype.
- **Producing Project:** The Producing Project feature requires one element and one system setting.
  - Producing Project requires a plugin to map the asset to the list of projects which contributed to its creation. This plugin is "Producing Projects" (`ProducingProjectsPlugin`).
  - The System Setting "**Producing Projects**" (`cmee.project.producing.enabled`) allows use of Producing Projects.
- **Unique Element:** The Unique Element feature requires one element and two system settings.
  - The Unique Element requires a plugin to persist the unique value. This plugin is "Unique Element" (`UniqueElementPlugin`).
  - The System Setting "**Enable Custom Unique Element**" (`cmee.unique-element.enable`) allows use of the Unique Element.
  - The System Setting "**Unique Element Per Type**" (`cmee.unique-element.per-assettype`) determines if the Unique Element applies to the entire system or within individual types.
- **Predicted Net Hours Saved (PNHS):** PNHS drives most reporting. It requires one element.
  - PNHS requires a Numeric Text Field to persist the number of hours saved. It must be a positive number. The Numeric Text Field must have an XML Mapping of "custom-data/development-hours".
  - If this element needs to be added to an asset type, create a Numeric Text Field named "development hours" to properly create the XML Mapping. Then Edit the element and change the name to "Predicted Net Hours Saved" to properly label it. Do not use quotes when adding or editing the element.
- **Total development hours (TDH):** This element drives some reporting. It requires one element.
  - Total development hours (TDH) requires a Numeric Text field to persist the number of hours saved. It must be a positive number. The Numeric Text field must have an XML Mapping of "custom-data/total-development-hours--tdh-".
  - If this element needs to be added to an asset type, then create a Numeric Text field named Total development hours (TDH). Do not use quotes when adding or editing the element.
- **Harvesting:** The Harvester tool allows integration with many products, including Oracle SOA Suite, Oracle Service Bus, Oracle Enterprise Manager, Oracle BPEL PM, UDDI Registries, and others. For harvesting to be possible the Harvester Solution Pack is required to be imported and all of the asset types included in that Solution Pack must exist and be active. The Harvester will write metadata into a combination of mandatory-data, custom-data, and CMF metadata sections of an asset.
  - Fields utilized by Harvesting include:
    - \* **Name:** The asset name is a concatenation of several harvested attributes ·

For abstract entities, the Name is the "**internalName**", which is a combination of the "**namespace**", "**shortName**", and "**discriminator**".

For artifact entities, the Name is a combination of the "**namespace**" and "**path**".

For type entities, the Name is the "**internalName**", which is a combination of the entity's "**type**" and "**shortName**".

- \* **Version:** The asset "**version**" set with default to (1.0) by harvester.
- \* **FileInformation:** The asset file information contains the list of files available on an asset's Use/Download operation. For artifact entities, the File Information is constructed to contain either the URI of the file or the file contents. The parameter "path" contains the URI.
- \* **harvesterProperties:** Metadata from an harvesting operation places its data into a CMF metadata entry named "internal.introspector.store". The metadata is exposed to asset detail within the viewer named "**Introspection Standard Attributes Viewer**". The expected properties within the internal.introspector.store vary by harvesting source and Oracle Enterprise Repository asset type. For more information about the harvester properties, see [Section A.7, "Harvester Properties"](#).

Fields on asset types used by various integrations with Oracle Enterprise Repository:

Field	Core OER	XU	OSB Harvester	SOA Suite Harvester	raw WSDL Harvester	AIA Harvester	EM Harvester	Ovation Harvester
Name	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Version	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Description	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
File Information	Yes							
harvesterProperties		Yes	Yes	Yes	Yes	Yes	Yes	Yes
UDDI Service Key	Yes	Yes						
UDDI Registries		Yes						

### A.6.6 Description of the Asset Type Catalog

The Asset Type Catalog lists those Asset Types, which are shipped with the Base Data pack and the Harvester Solution Pack. It is contained within an Excel spreadsheet named Asset Type Catalog.xls. Each asset type is included with the complete list of elements found on that type. For each element several pieces of information are listed, as follows:

- **Asset Type:** The asset type on which the element resides.
- **Source:** The Solution or Data Pack, which ships the element with the asset type.
- **Tab:** The Editor tab on which the element can be found. The elements are not ordered within the tab.
- **Element:** The name of the element as it appears within the Asset Editor.
- **Type:** The data type for the element using the data types listed when adding a new element in the Type Manager.

- **XML Mapping:** The XPath to the data as it would appear in the XML generated by an `assetRead()` operation;
- **Properties:** A list of properties relevant to the data type of the element, strings have the maximum length, numbers have the scale, precision and range, lists have the name of the value list, etc.
- **Value List:** For elements that are restricted to the list of values specified by an acceptable value list, the legal values are shown. Oracle Enterprise Repository allows grandfathered data to be retained in value restricted elements.
- **Required:** The value of the Required flag for the element.
- **Indexed:** The value of the Indexed flag for the element.
- **Notes:** A column to allow Oracle Enterprise Repository customers to add their own notes about their use of the element.

## A.7 Harvester Properties

Property Name	Applies to Asset Type(s)	Description
Date Harvested	All	Date that the asset was harvested into Oracle Enterprise Repository
Harvested by	All	Oracle Enterprise Repository user who did the harvesting
Harvester Description	All	Optional description that was set when running the harvester
Harvester Version	All	Version of the Harvester that was used
Namespace	All non-artifact types	Namespace of the project, as set when running the harvester
Product Name	All	Product that the artifacts came from
Product Version	All	Version of the product that the artifacts came from
intname	All non-artifact types	Internal unique name of the asset, used by harvester for duplicate detection
(various)	Business Process: BPEL,Deployment: BPEL,Service (type=Mediator),Human TaskBusiness Rule,Spring Context	SCA component properties, as read from composite.xml
(various)	Composite,Composite Deployment	SCA composite properties, as read from composite.xml
(various)	Adapter Configuration	JCA properties of the adapter configuration, as read from the WSDL or .JCA file defining it
Binding Type	Endpoint,Composite Deployment Service,Composite Deployment Reference	Type of binding used by an endpoint
Deployment URL	Endpoint,Deployment: BPEL,Composite Deployment, Composite Deployment Service,Composite Deployment Reference	URL of the server which the project was remotely harvested from.

Property Name	Applies to Asset Type(s)	Description
EJB Version	Endpoint, Service	"EJB Version" property of EJB endpoints, as read from composite.xml. Indicates whether it's an EJB 2.1 or 3.0 service.
Endpoint URI	Endpoint	URI at which the endpoint can be invoked. In the case of Adapter Configuration
Events Published	Business Process: BPEL,Deployment: BPEL,Service (type=Mediator),Human TaskBusiness Rule,Spring Context,Composite,Comp osite Deployment	QNames of events subscribed to by this component, or by components in this composite, as defined in composite.xml
Events Subscribed To	Business Process: BPEL,Deployment: BPEL,Service (type=Mediator),Human TaskBusiness Rule,Spring Context,Composite,Comp osite Deployment	QNames of events subscribed to by this component, or by components in this composite, as defined in composite.xml
Interface Type	Interface,Composite Service, Composite Reference	Type of interface offered by a service
Invoked Operations	Business Process: BPEL	Operations that this business process invokes on services it orchestrates.
Registry Name	Endpoint, Service	"Registry Name" property of ADF BC endpoints, as read from composite.xml
Request Format	Endpoint	Oracle Service Bus Request format as used with the Oracle Service Bus Messaging transport
Response Format	Endpoint	Oracle Service Bus Repsonse format as used with the Oracle Service Bus Messaging transport
SCA Name	Endpoint,Composite Deployment Service,Composite Deployment Reference	Identifier of the interface or binding as found in composite.xml. Used internally by harvester for determining wire and event relationships.
Service Type	Service	Describes type of service
Transport Type	Endpoint	Type of Oracle Service Bus transport used by an Oracle Service Bus endpoint

Source Product	Source Binding / binding type / address type	Oracle Enterprise Repository Transport Type
Oracle Service Bus	Mixed	(depends on OSB transport type, see below)
Oracle Service Bus	XML	(depends on OSB transport type, see below)
Oracle Service Bus	abstract XML	(depends on OSB transport type, see below)
Oracle Service Bus	SOAP	(depends on OSB transport type, see below)
Oracle Service Bus	abstract SOAP	(depends on OSB transport type, see below)
Oracle Service Bus	.flow	(depends on OSB transport type, see below)
SOA Suite	binding.adf	adf bc
SOA Suite	binding.ejb	ejb

Source Product	Source Binding / binding type / address type	Oracle Enterprise Repository Transport Type
SOA Suite	binding.b2b	b2b
SOA Suite	binding.ws	(depends on wsdl, see below)
SOA Suite	binding.wsif	(depends on wsdl, see below)
SOA Suite	binding.jca	jca
raw WSDL	soap, soap12	soap
raw WSDL	http	http
raw WSDL	java	java
raw WSDL	ejb	ejb
raw WSDL	jca	jca
raw WSDL	jms	ims

Source Product	Source Transport Type	Oracle Enterprise Repository Transport Type
Oracle Service Bus	http	http
Oracle Service Bus	jms	jms
Oracle Service Bus	file	file
Oracle Service Bus	ftp	ftp
Oracle Service Bus	local	local
Oracle Service Bus	other	(as it appears in OSB)

Source Product	Source interface / Component / File Type	Oracle Enterprise Repository Service Type
Oracle Service Bus	.biz	Business Service
Oracle Service Bus	.proxy	Proxy Service
Oracle Service Bus	.flow	Split-join Service
SOA Suite	interface.wsdl	Web Service
SOA Suite	interface.java	Java Service
SOA Suite	implementation.mediator	Mediator Service
raw WSDL	(all)	Web Service

Source Product	Source Interface/Binding/File type	Oracle Enterprise Repository Interface Type
Oracle Service Bus	.flow	Web Service
Oracle Service Bus	abstract SOAP	Any Web Service
Oracle Service Bus	Abstract XML	Any Web Service
Oracle Service Bus	Mixed	Messaging
Oracle Service Bus	XML	Web Service
Oracle Service Bus	SOAP	Web Service
SOA Suite	interface.wsdl	Web Service

---

<b>Source Product</b>	<b>Source Interface/Binding/File type</b>	<b>Oracle Enterprise Repository Interface Type</b>
SOA Suite	interface.wsdl	Java Service
raw WSDL	(all)	Web Service

