Oracle® Fusion Middleware

Configuration Guide for Oracle Enterprise Repository

11*g* Release 1 (11.1.1.7)

E16580-14

October 2013

Documentation for administrators that describes Oracle Enterprise Repository configuration and customization tasks to support your service oriented architecture (SOA) Governance initiative.

ORACLE

Oracle Fusion Middleware Configuration Guide for Oracle Enterprise Repository, 11*g* Release 1 (11.1.1.7)

E16580-14

Copyright © 2008, 2013, 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, the following notice is applicable:

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

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

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

Contents

Preface xix

Audience xix
Documentation Accessibility xix
Related Documents xix
Conventions xx

What's New in This Guide for Release 11.1.1.7 xxi

1 Basic Configuration

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
1.3 Users 1-3
1.3.1 Creating a New User 1-3
1.3.2 Viewing User Information 1-4
1.3.3 User Search 1-5
1.3.4 Cloning a User 1-6
1.3.5 Editing User Information 1-6
1.4 Projects 1-6
1.4.1 Viewing Project Details 1-7
1.4.2 Creating a Project 1-8
1.4.3 Editing a Project 1-11
1.4.4 Reassigning Users/Usage 1-11
1.4.5 Locating a Project 1-12
1.4.6 Closing a Project 1-12
1.4.7 Adding a Relationship to a Project 1-13
1.5 Roles 1-16
1.5.1 User Role Descriptions 1-16
1.5.2 Creating a New Role 1-18
1.5.3 Editing a Role 1-20
1.5.4 Deleting a Role 1-20
1.6 Access Settings 1-21
1.6.1 Granting Permissions to a Role 1-21
1.6.2 Editing Permissions 1-23

1.6.3 Verifying User Permissions 1-23
1.6.4 Security Considerations 1-24
1.6.4.1 Access Setting and the Asset Editor 1-24
1.6.4.2 Permissions 1-25
1.7 Sessions 1-26
1.7.1 Viewing Sessions 1-26
1.7.2 Deleting a Single Session 1-27
1.7.3 Deleting Multiple Session 1-27
1.8 Types 1-27
1.8.1 Viewing Types 1-27

2 Customizing Oracle Enterprise Repository

2.1 Overview 2-1
2.1.1 Accessing the Oracle Enterprise Repository Home Page 2-1
2.1.2 Customizing the Oracle Enterprise Repository Home Page 2-2
2.2 Rebranding Oracle Enterprise Repository 2-2
2.2.1 Start with CSS 2-2
2.2.2 Customizing Skins 2-3
2.2.2.1 Choosing a Skin 2-3
2.2.2.2 Creating a Skin 2-3
2.2.2.3 Images 2-3
2.2.2.4 Activating Your Custom Skin 2-4
2.2.2.5 Installing a Oracle-supplied Custom Skin in Oracle Enterprise Repository 2-4
2.2.2.6 Using the Directory Structure 2-4
2.3 Role/Department-based Homepage Display 2-5
2.3.1 Installation 2-6
2.4 Install Oracle Enterprise Repository Solution Packs 2-7

3 Configuring Oracle Enterprise Repository to use External Authentication Tooling

3.1 Overview 3-1
3.1.1 LDAP/Active Directory 3-1
3.1.2 Enable LDAP Integration System Properties 3-2
3.1.3 Modify LDAP/Active Directory Properties 3-3
3.1.4 Security Considerations 3-5
3.1.4.1 Use Case Sample Scenarios 3-6
3.1.5 LDAP Property Examples 3-7
3.2 eTrust Single Sign-On 3-8
3.2.1 Overview 3-8
3.2.2 Configure Oracle Enterprise Repository For Use With Single Sign-On Authentication .. 3-8
3.2.2.1 Enable Single Sign-On Integration System Properties 3-9
3.2.2.2 Using the Oracle Enterprise Repository SSO Integration with Basic Authentication 3-12
3.2.2.3 Modify Application Property Files Manually 3-13
3.2.3 Advanced Options 3-14
3.2.3.1 Creating/Assigning Default Roles for New Users 3-15

3.2.3.2 Create New Users/Allow Unapproved Users 3-16
3.2.3.3 Enable Unapproved/New User Login 3-16
3.2.3.4 New User Notification 3-16
3.2.3.5 Syncing Departments 3-16
3.2.3.6 Syncing Roles 3-17
3.2.3.7 Enable Debug Logging 3-18
3.3 Container Managed Setup 3-18
3.3.1 Overview 3-19
3.3.2 Configure Oracle Enterprise Repository for Container Managed Authentication . 3-19
3.3.3 Modify the Web Application's Web.xml File to Allow for Container Authentication 3-20

4 Configuring Advanced Role-based Access Control

4.1 Overview 4-1
4.2 Security Considerations 4-2
4.2.1 Access Settings and the Asset Editor 4-2
4.2.2 Enabling Custom Access Settings 4-2
4.3 Basic Concepts 4-3
4.3.1 Roles 4-3
4.3.2 Access Settings 4-4
4.4 Access Definitions 4-4
4.5 Process Overview 4-6
4.6 Granting and Denying Permissions 4-7
4.6.1 Grant/Deny Access to Specific Assets 4-7
4.6.1.1 Prerequisites 4-7
4.6.1.2 Create the Asset CAS 4-8
4.6.2 Grant/Deny Access to Specific Download Files Within an Asset 4-12
4.6.2.1 Prerequisites 4-12
4.6.2.2 Create the File CAS 4-13
4.7 Configuring Access Settings for Existing Roles 4-13
4.7.1 User Roles and Default Privileges 4-14
4.7.2 Access Options 4-15
4.7.2.1 Option I: Grant or deny access to specific assets using CAS 4-16
4.7.2.2 Option II: Grant or deny access to specific files and assets using CAS 4-20
4.8 Role-based Access Control Use Cases 4-25
4.8.1 Use Case 1: Expose Web Services to Customers and Trading Partners 4-25
4.8.1.1 Benefit 4-25
4.8.1.2 Overview 4-25
4.8.1.3 Solution 4-26
4.8.1.3.1 Prerequisite 4-26
4.8.1.3.2 Create and Assign the Roles 4-26
4.8.1.3.3 Organize the Metadata 4-27
4.8.1.3.4 Create the Relationships 4-28
4.8.1.3.5 Apply the Relationship 4-28
4.8.1.3.6 Results 4-30
4.8.1.3.7 Validation Test 4-31
4.8.2 Use Case 2: Manage Intellectual Property in an Outsourced Environment 4-31

4.8.2.1 Benefits ... 4-31
4.8.2.2 Overview ... 4-32
4.8.2.3 Solution ... 4-32
4.8.2.3.1 Solution for Outsourced Development Teams ... 4-32
4.8.2.3.2 Solution for Export Controls ... 4-35
4.8.3 Use Case 3: Establish a Federated Repository ... 4-37
4.8.3.1 Benefit ... 4-37
4.8.3.2 Overview ... 4-37
4.8.3.3 Solution ... 4-37
4.8.3.3.1 Prerequisite ... 4-37
4.8.3.3.2 Create the Role ... 4-37
4.8.3.3.3 Create the Custom Access Settings ... 4-38
4.8.3.3.4 Validation Test ... 4-41
4.8.4 Use Case 4: Manage the Asset Lifecycle ... 4-42
4.8.4.1 Benefit ... 4-42
4.8.4.2 Overview ... 4-43
4.8.4.2.1 Asset Lifecycle Stages ... 4-43
4.8.4.2.2 General Configuration ... 4-44
4.8.4.2.3 Create the Roles ... 4-44
4.8.4.2.4 Create the Custom Access Settings ... 4-44
4.8.4.3 Configuring the Use Case Solutions ... 4-49
4.8.4.3.1 Requirements Gathering Stage Solution ... 4-49
4.8.4.3.2 Design and Development Stage Solution ... 4-51
4.8.4.3.3 Beta Release Stage Solution ... 4-51
4.8.4.3.4 Release Stage Solution ... 4-52
4.8.4.3.5 Scheduled for Retirement Stage Solution ... 4-52
4.8.4.3.6 Retired Stage Solution ... 4-52
4.8.4.4 Asset Lifecycle Management Solution Validation Tests ... 4-52
4.8.5 Use Case 5: Limit Access to Source Code Files to Asset Production Teams ... 4-54
4.8.5.1 Benefit ... 4-54
4.8.5.2 Overview ... 4-54
4.8.5.3 Solution ... 4-54
4.8.5.3.1 Create the Roles ... 4-54
4.8.5.3.2 Create the Customer Access Settings ... 4-54
4.8.5.3.3 Results ... 4-57
4.8.5.3.4 Validation Test ... 4-58
4.8.6 Use Case 6: Grant Browse-only Repository Access to Specific Groups ... 4-59
4.8.6.1 Benefit ... 4-59
4.8.6.2 Overview ... 4-59
4.8.6.3 Solution ... 4-59
4.8.6.4 Validation Test ... 4-60

5 Diagnostics and Password Encryption

5.1 Accessing Oracle Enterprise Repository Diagnostics Page ... 5-1
5.2 Generating Encrypted Passwords ... 5-2
5.2.1 Encrypted Passwords Used in Oracle Enterprise Repository Deployment ... 5-2
5.2.2 Other Passwords ... 5-2

5.3 Encrypting the Configuration File Passwords ... 5-3
5.3.1 Harvester Configuration File ... 5-3
5.3.2 Exchange Utility Configuration File ... 5-3
5.3.3 Workflow Configuration File ... 5-4

6 Configuring and Using Automated Harvesting in Design-time and Runtime Environments

6.1 Getting Started with Harvester ... 6-1
6.1.1 Prerequisites ... 6-1
6.1.2 Harvester Functionality ... 6-2
6.1.3 Artifacts/Products Version Matrix ... 6-3
6.2 Configuring the Harvester ... 6-3
6.2.1 Obtaining the Harvester ... 6-4
6.2.2 Configuring the Harvester for the Command Line ... 6-5
6.2.2.1 Setting Repository Connection Information for the Command Line ... 6-6
6.2.2.2 Selecting the Artifacts to Harvest for the Command Line ... 6-6
6.2.3 Using Harvester within an Integrated Development Environment ... 6-9
6.2.4 Invoking the Harvester Using the Repository.Submit Ant Task ... 6-10
6.2.4.1 Specifying Parameters for the repository.submit Ant Task ... 6-10
6.2.4.2 Specifying Parameters as Nested Elements ... 6-11
6.2.4.3 Running the Harvester from Ant ... 6-14
6.2.4.4 Using the Third-Party Tasks ... 6-15
6.2.5 Invoking Harvester from WLST ... 6-15
6.2.6 Runtime Harvesting Details ... 6-17
6.2.6.1 Harvesting from Oracle SOA Suite Server ... 6-17
6.2.6.2 Harvesting Web Services from WebLogic Server ... 6-19
6.2.6.3 Harvesting Web Services from Oracle Service Bus ... 6-19
6.2.7 Performing Optional Harvester Configuration ... 6-20
6.2.7.1 Oracle Metadata Service URLs ... 6-20
6.2.7.2 Classpath URLs ... 6-21
6.2.8 Configuring Logging for the Harvester ... 6-21
6.2.9 Transaction Handling in Harvester ... 6-21
6.2.10 Advanced Configuration ... 6-21
6.3 Harvesting of Artifacts ... 6-23
6.3.1 Harvesting Artifacts ... 6-23
6.3.1.1 BPEL ... 6-25
6.3.1.2 WSDL ... 6-25
6.3.1.3 XSD ... 6-26
6.3.1.4 XSL ... 6-26
6.3.1.5 Harvesting Oracle SOA Suite ... 6-26
6.3.1.6 JCA Adapters ... 6-31
6.3.1.7 Applications via Oracle BSE (WSDL/SOAP) ... 6-32
6.3.1.8 Proxy Service or Business Service ... 6-33
6.3.1.9 Oracle Service Bus (OSB) Project ... 6-34
6.3.1.10 BPMN ... 6-35
6.3.2 Harvesting Oracle Business Process Analysis Projects ... 6-36
6.3.3 Harvesting Oracle Business Process Management 11*g* Projects ... 6-37

6.3.4 Harvesting Oracle Service Bus 11*g* Assets 6-38
6.3.5 Harvesting Oracle Service Bus 10*g* Assets 6-38
6.3.6 Harvesting a SOA Project from JDeveloper 6-39
6.3.7 Harvesting Deployed Composites from JDeveloper 6-40
6.3.8 Associate JDeveloper Application with Oracle Enterprise Repository Project 6-40
6.3.9 Consume WSDL Services in Oracle Enterprise Repository from Jdeveloper 6-40
6.3.10 Detecting Duplicate Artifacts 6-41
6.3.11 Downloading Harvested Artifacts 6-41
6.3.12 Searching Harvested Assets 6-41
6.3.12.1 Using Consumption Method 6-42
6.3.12.2 Using Metadata 6-42
6.3.13 Previewing of the Created Assets in Harvester 6-43
6.3.14 Best Practices 6-44
6.3.14.1 Recommended Privileges for Harvesting 6-44
6.3.14.2 Do Not Override the Namespace Parameter 6-44
6.3.14.3 Namespaces in WSDL Files 6-44
6.3.14.4 Harvesting Completed Work 6-45
6.3.14.5 Harvesting and Maintenance Releases of XSD 6-45
6.3.14.6 Harvesting Metadata throughout the Asset Lifecycle 6-45
6.3.14.7 Downloading WSDL Artifacts 6-46
6.3.14.8 Harvesting Retired Endpoints 6-46
6.3.14.9 Harvesting SOA Projects 6-46
6.3.14.10 Harvesting from Oracle JDeveloper 11*g* 6-47
6.3.14.11 Harvesting a File in the Harvester Folder 6-47
6.3.14.12 Running Harvester and Exchange Utility on the Same Computer 6-47
6.3.14.13 Harvesting Composite Service WSDL 6-47
6.3.15 Known Issues 6-47
6.3.15.1 Asset Types Must be Present in the System 6-47
6.3.15.2 Two Versions of an Asset Type 6-47
6.3.15.3 Do Not Delete the Harvester-Specific Metadata Entries in a in a Harvested Asset ... 6-48
6.3.15.4 Supported Remote Server Types 6-48
6.3.15.5 Using Incorrect Encrypted Password 6-48
6.3.15.6 View in Repository Option 6-49

7 Configuring Metrics

7.1 Overview 7-1
7.1.1 Metrics Elements 7-1
7.1.2 Adding a Reusable Metrics Tab 7-2
7.2 Validate the Elements on the Asset Type 7-3
7.3 Running Queries Against the Database 7-3
7.3.1 dbqueries 7-5

8 Configuring E-Mail Notifications and Distribution Lists

8.1 Overview 8-1
8.1.1 Using E-Mail Notifications and E-Mail Distribution Lists 8-1
8.2 Setting Up E-Mail Notification 8-1

8.3 Managing E-mail Distribution Lists ... 8-2
8.3.1 Updating a Distribution List ... 8-4
8.3.2 When List Members Are Copied on E-mails ... 8-4
8.4 Working with E-Mail Templates ... 8-4
8.4.1 Viewing E-Mail Templates ... 8-6
8.4.2 Editing E-Mail Templates ... 8-7

9 Configuring Oracle Enterprise Repository Express Workflows

9.1 Introduction to Oracle Enterprise Repository Express Workflow ... 9-1
9.2 Configuring Express Workflows ... 9-2
9.2.1 Step 1:..... Install the OER Express Workflows SOA Composite Application 9-3
9.2.2 Step 2: Configure OER Event Manager to deliver events to the deployed SCA ... 9-4
9.2.2.1 Configuring Web Service Endpoints ... 9-5
9.2.2.2 Enabling the Event Manager ... 9-6
9.2.3 Step 3: Define and Configure your Express Workflows ... 9-6
9.3 Running the Sample Workflow ... 9-7
9.4 Modifying the Sample Workflow ... 9-8
9.4.1 Modifying the Express Workflows Configuration File ... 9-8
9.4.1.1 Producing Projects ... 9-8
9.4.1.2 Approval Flows ... 9-9
9.4.1.3 Event-to-Action Mappings ... 9-10

10 Configuring Oracle Enterprise Repository to Exchange Metadata with the Oracle Service Registry

10.1 Getting Started With the Oracle Registry Repository Exchange Utility ... 10-1
10.1.1 What is the Oracle Registry Repository Exchange Utility? ... 10-1
10.1.1.1 Valid Metadata Entities ... 10-2
10.1.1.2 Prerequisites ... 10-2
10.1.2 Example Use Cases ... 10-2
10.2 Configuring the Oracle Registry Repository Exchange Utility ... 10-3
10.2.1 Installing and Configuring the Oracle Registry Repository Exchange Utility ... 10-4
10.2.1.1 Install the Oracle Registry Repository Exchange Utility ... 10-4
10.2.1.2 Configure Using the Asset Type Name ... 10-4
10.2.2 Installing Oracle Service Registry ... 10-4
10.2.3 Configuring the Oracle Registry Repository Exchange Utility Configuration File 10-5
10.2.3.1 Setting the Repository Connection Information ... 10-5
10.2.3.2 Setting the Registry Connection Information ... 10-5
10.2.3.3 Setting the Repository Query ... 10-6
10.2.3.3.1 Query by Name ... 10-6
10.2.3.3.2 Query by Registration Status ... 10-7
10.2.3.3.3 Query by Categorizations ... 10-7
10.2.3.3.4 Query by Filters ... 10-7
10.2.3.4 Setting the Destination Registries ... 10-10
10.2.3.5 Setting the Registry Query ... 10-10
10.2.3.6 Setting the Source Registry ... 10-11

10.2.4 Configuring Oracle Enterprise Repository Categorizations in the UDDI Mappings File 10-11
10.2.5 Configuring the tModelKey UDDI Setting 10-12
10.2.6 Understanding the Oracle Registry Repository Exchange Utility's Property File 10-15
10.2.7 XU Support 10-16
10.2.7.1 XU Support for Manually-Created Services 10-17
10.2.7.2 XU Support for Non-WSDL Services Harvested from Oracle Service Bus..... 10-17
10.2.7.3 Configuring XU to Publish Non-Harvested or Custom Asset Types and Relationships 10-18
10.3 Using the Oracle Enterprise Repository Exchange Utility 10-19
10.3.1 Running the Oracle Registry Repository Exchange Utility 10-20
10.3.1.1 Invoking the Oracle Registry Repository Exchange Utility Using Workflows 10-21
10.3.1.1.1 Prerequisities..... 10-21
10.3.1.1.2 Synchronizing Using Timer Based Workflows..... 10-21
10.3.1.1.3 Synchronizing When Events are Triggered..... 10-22
10.3.1.1.4 Example Use Cases 10-22
10.3.2 How the Exchanged Metadata Is Synchronized 10-23
10.3.2.1 Synchronizing the Metadata Published from Oracle Enterprise Repository to Oracle Service Registry 10-23
10.3.2.1.1 Business Entities 10-23
10.3.2.1.2 Service Keys 10-24
10.3.2.1.3 Endpoint 10-24
10.3.2.1.4 Categorizations 10-25
10.3.2.1.5 Registration and Active Status 10-25
10.3.2.1.6 Sample Flow of Metadata from Oracle Enterprise Repository to Oracle Service Registry 10-27
10.3.2.2 Synchronizing the Metadata from Oracle Service Registry to Oracle Enterprise Repository 10-27
10.3.2.2.1 Business Entities 10-28
10.3.2.2.2 Endpoint 10-28
10.3.2.2.3 Categorizations 10-28
10.3.2.2.4 Sample Flow of Metadata from Oracle Service Registry to Oracle Enterprise Repository 10-28
10.3.3 Searching for Oracle Service Registry Exchanged Metadata in Oracle Enterprise Repository 10-29
10.3.4 Checking the Oracle Registry Repository Exchange Utility Log File 10-30
10.3.5 Known Issues 10-30
10.3.5.1 Resynchronizing Oracle Service Registry Services..... 10-30
10.3.5.2 Publishing Services to Oracle Service Registry 10-31
10.3.5.3 Running an Incorrect Version of Java..... 10-31

11 Configuring Prescriptive Reuse

11.1 Overview 11-1
11.2 Creating a Compliance Template Type 11-2
11.3 Creating a Compliance Template 11-5
11.4 Apply a Compliance Template to a Project..... 11-6
11.5 Using Assets Associated with Compliance Templates (Prescribed Assets)..... 11-7

11.6 Running a Project Compliance Report 11-9

12 Configuring Assets in Progress

12.1 Overview 12-1
12.2 Viewing Assets in Progress 12-2
12.3 Enabling Assets in Progress 12-3
12.3.1 Set Default Asset Status 12-4
12.4 Disabling Assets in Progress 12-4
12.5 Enabling Viewing of Assets in Progress Through an IDE 12-4

13 Configuring Automated Usage Detection

13.1 Overview 13-1
13.2 Enabling SFID 13-2
13.3 Adding the SFID Element to a Type 13-2
13.4 Adding an SFID to an Asset 13-5
13.4.1 Update an SFID File 13-5
13.4.2 Remove an SFID File 13-6
13.5 SFID Tool Kit 13-6
13.5.1 Installation and Configuration 13-6
13.5.1.1 Enabling Harvesting of Unrecognized SFIDs 13-6
13.5.1.2 Object Model 13-7
13.5.1.3 Description of Sample Directory Crawler 13-7
13.6 SFID Manager 13-8
13.6.1 Enabling the SFID Manager 13-8
13.6.2 Using the SFID Manager 13-9
13.7 Calculating the SFID Programmatically 13-11
13.8 SFID Limitations 13-11

14 Configuring Policy Management

14.1 Overview 14-1
14.2 Activating the Policy Management Feature 14-2
14.3 Creating a Policy Type 14-3
14.4 Adding the Policy Assertion Element to the Policy Type 14-4
14.5 Creating a Policy 14-5
14.6 Adding the Applied Asset Policies Element to a Type 14-6
14.7 Applying a Policy to an Asset 14-7
14.8 Viewing Policy Status 14-9
14.9 Validating Policy Assertions 14-9
14.10 Running the Oracle Enterprise Repository Policy Reports 14-12

15 Configuring Contract Management

15.1 Overview 15-1
15.1.1 Prerequisites 15-1
15.2 Import the Contract Management Solution Pack 15-2
15.3 Instructions for Service Providers 15-2

15.3.1 Create the Contract 15-2
15.3.2 Edit the Contract 15-2
15.3.3 Create Relationships 15-2
15.3.3.1 Terms of Use 15-2
15.4 Instructions for Service Consumers 15-3

16 Configuring Reporting with BI Publisher

16.1 Deploy Oracle Enterprise Repository Pre-configured BI Publisher Content 16-1
16.1.1 Configure BI Publisher with Oracle Enterprise Repository 16-1
16.2 Security Model 16-2
16.2.1 The Administrator User 16-2
16.2.2 The OER User and OER_REPORTS Role 16-2
16.2.3 Set Local Time Zone for Report Footer 16-2
16.3 Configuring the Oracle Enterprise Repository JDBC Data Source 16-2
16.4 Configuring the BI Publisher Server - Cache Expiration 16-4
16.5 Configuring the Oracle Enterprise Repository System Settings to Enable Reporting 16-4

17 System Settings Overview

17.1 Access System Settings 17-1
17.2 Search for System Settings 17-2
17.3 Enable System Settings Properties 17-3
17.4 Move Settings to Database 17-3
17.5 System Settings in Oracle Enterprise Repository 17-3
17.5.1 Functional Settings 17-4
17.5.1.1 General User Interface 17-4
17.5.1.2 Search 17-9
17.5.1.3 Print Asset Detail via PDF 17-12
17.5.1.4 Projects and Navigator 17-13
17.5.1.5 Assets in Progress 17-15
17.5.1.6 Policy Management 17-16
17.5.1.7 Notifications 17-17
17.5.1.8 Access Control 17-18
17.5.1.9 Software File Identification (SFID) 17-19
17.5.1.10 Use - Download 17-19
17.5.1.11 Custom Links 17-20
17.5.1.12 Reports 17-22
17.5.1.13 Miscellaneous 17-23
17.5.2 Server Settings 17-25
17.5.2.1 General 17-26
17.5.2.2 Server Resources 17-28
17.5.2.3 Email 17-29
17.5.2.4 Upload Area 17-31
17.5.2.5 Audit 17-31
17.5.2.6 Caching 17-32
17.5.3 Enterprise Authentication Settings 17-32
17.5.3.1 General 17-33
17.5.3.2 Authentication Token 17-35

17.5.3.3 Password Settings ... 17-36
17.5.3.4 Plugin Login Settings ... 17-37
17.5.3.5 LDAP and Active Directory Settings ... 17-39
17.5.3.6 Single Sign-On (SSO) Settings ... 17-42
17.5.3.7 Container Login Settings ... 17-45
17.5.3.8 Cookie Login Settings ... 17-46
17.5.3.9 Guest Login Settings ... 17-47
17.5.4 Open API Settings ... 17-47
17.5.4.1 Open API ... 17-48
17.5.5 External Integration Settings ... 17-48
17.5.5.1 UDDI ... 17-49
17.5.5.2 File Stores ... 17-49
17.5.5.3 Rational ClearCase ... 17-50
17.5.5.4 Eventing ... 17-50
17.5.6 Import / Export and Introspection Settings ... 17-51
17.5.6.1 Import / Export ... 17-51
17.5.6.2 WSDL and XSD ... 17-52
17.6 Known Issues ... 17-52
A.1 Introduction to Oracle Enterprise Repository Workflow ... A-1
A.2 Introduction to Oracle Enterprise Repository Automated Workflows ... A-2
A.2.1 Software Components ... A-3
A.2.2 Automated Workflows ... A-3
A.2.3 Event Management Tools ... A-4
A.2.3.1 Web-based Process Administrator ... A-4
A.2.3.2 Log Viewer ... A-4
A.2.3.3 Email Notification Templates ... A-5
A.2.3.4 Workflow Configuration Tools ... A-5
A.2.3.5 Generating a New Config File ... A-5
A.2.3.6 Refreshing an Existing Config File ... A-5
A.2.3.7 Encrypting Config File Passwords ... A-5
A.3 Installing Oracle Enterprise Repository Workflow into Oracle BPM 10.3 ... A-5
A.3.1 Requirements ... A-6
A.3.2 Step 1: Install Oracle BPM 10.3.2 ... A-6
A.3.3 Step 2: Unzip the Oracle Enterprise Repository Workflow Installer ... A-7
A.3.4 Step 3: Configure the build.properties File ... A-7
A.3.5 Step 4: Configure the setenv File ... A-10
A.3.6 Step 5: Generate (or Regenerate) the workflow.xml File ... A-11
A.3.7 Step 6: Edit the workflow.xml File ... A-11
A.3.8 Step 7: Encrypt the workflow.xml File ... A-12
A.3.9 Step 8: Copy the JDBC jar(s) ... A-12
A.3.10 Step 9: Verify Permissions ... A-12
A.3.11 Step 10: Run the Setup Script ... A-12
A.3.12 Step 11: Verify the Setup Script ... A-13
A.4 Configuring the Oracle Enterprise Repository Event Manager ... A-14
A.4.1 What is the Oracle Enterprise Repository Event Manager? ... A-14
A.4.2 Configuring the Event Manager ... A-15
A.4.3 Triggering an Asset Event ... A-16

A.4.4 Configuring the Event Manager's System Settings A-17
A.4.4.1 Enabling the Event Manager A-17
A.4.4.2 Configuring Optional Event Manager Settings A-17
A.4.4.2.1 Eventing Manager Notifier Thread Sleep (seconds) A-18
A.4.4.2.2 Eventing Manager Store Thread Sleep (seconds) A-18
A.4.4.2.3 Eventing Manager Store Delivery Sleep (seconds) A-18
A.4.4.2.4 Batch Size for Event Manager Deliveries A-18
A.4.5 Configuring the Subscription Manager A-18
A.4.5.1 Configuring Web Service Endpoints A-19
A.4.5.2 Setting the Expression to Filter Events A-20
A.4.5.2.1 Delivering all Events to an Endpoint A-20
A.4.5.2.2 Delivering Events to an Endpoint Filtered by Event Type A-20
A.4.5.2.3 Delivering Events to an Endpoint Filtered Using a JMS Message Selector A-21
A.4.5.2.4 JMS Message Selector Examples A-21
A.4.6 Configuring Logging of Event Manager Events A-21
A.5 Administering Oracle Business Process Management Processes A-22
A.5.1 Overview A-22
A.5.2 Administering Oracle Business Process Management Web Applications A-22
A.5.2.1 Starting the Oracle Business Process Management Admin Center A-22
A.5.2.2 Starting the Oracle Business Process Management Process Engine A-23
A.5.2.3 Defining the Oracle Business Process Management Participants A-23
A.5.2.3.1 Oracle Business Process Management Administrators A-24
A.5.2.3.2 Advanced Registration Flow Participant A-24
A.5.3 Tuning the Oracle Business Process Management Automated Workflows Engine A-24
A.5.3.1 Advanced Properties A-25
A.5.3.2 Database Runtime Properties A-25
A.5.3.3 Memory and Execution Thread Properties A-25
A.5.4 Configuring a Standalone Process Engine for Failover A-26
A.5.5 Using the Oracle Business Process Management Log Viewer A-26
A.5.5.1 Filtering Event Log Messages for Oracle Enterprise Repository Flows A-26
A.6 Configuring Automated Workflows A-27
A.6.1 Overview of Automated Workflows A-28
A.6.2 Creating and Customizing a Workflow Configuration File A-28
A.6.2.1 Defining the Oracle Enterprise Repository Connection and Registrar A-29
A.6.2.2 Encrypting the Registrar User Password A-29
A.6.3 Wiring Asset Events to Flows A-29
A.6.4 Automatic Asset Registration Flows A-31
A.6.4.1 Configuring Community Flows A-31
A.6.4.1.1 Setting the Community for an Oracle Enterprise Repository Project A-32
A.6.4.1.2 Setting the Community for an Asset Type A-32
A.6.4.1.3 Setting the Community for an Asset using the Type Manager and Asset Editor A-33
A.6.4.1.4 Configuring a Community to Automatically Accept an Asset A-34
A.6.4.1.5 Configuring a Community to Assign Assets for Tab Approval A-34
A.6.4.1.6 Configuring a Community to Assign Assets for Tab Approval Using Multi-tier A-35
A.6.4.1.7 Configuring a Community to Automatically Register an Asset A-35

A.6.4.1.8 Configuring a Community to Have a Dedicated Registrar A-35
A.6.4.2 Configuring Automated Acceptance and Automated Registration Flows..... A-35
A.6.4.2.1 Asset Type A-35
A.6.4.2.2 Categorization Settings..... A-35
A.6.4.2.3 Submitter Role A-36
A.6.4.2.4 Conflict Resolution and Precedence A-36
A.6.5 Multi-tier Automatic Assignment Flows A-36
A.6.5.1 Use Cases A-36
A.6.5.2 Using an <alerid> for Tab Approvals..... A-37
A.6.5.3 Setting Up a Community for Multi-tier Tab Approval..... A-38
A.6.5.4 Setting Up an Asset Type for Multi-tier Tab Approval A-38
A.6.6 Metadata Change Flows A-39
A.6.6.1 Use Cases A-39
A.6.6.2 Configuring Metadata Change Flows A-40
A.6.6.2.1 Available Metadata Change Events/States..... A-40
A.6.6.2.2 Available Flows That Can Be Wired to Actions A-41
A.6.6.2.3 Example Metadata Change Configuration..... A-42
A.6.6.2.4 Example Metadata Change Configuration That Checks for Metadata Value..... A-42
A.6.6.2.5 ChangeClassification A-43
A.6.6.2.6 ChangeCAS A-43
A.6.6.2.7 ChangeAssetLifecycle..... A-43
A.6.6.2.8 ApproveTabAction A-43
A.6.6.2.9 UnapproveTabAction A-44
A.6.6.2.10 AutoApproveTabAction A-44
A.6.6.2.11 UnapproveChangeManagementTab A-44
A.6.6.2.12 ResetChangeManagementTab..... A-44
A.6.6.2.13 NotifyCustomUser A-45
A.6.6.2.14 Invoking Flows Based on Approval of Named Tabs A-45
A.6.7 Time-based Escalation Flows..... A-45
A.6.7.1 Tracking Unsubmitted Assets A-46
A.6.7.2 Tracking Unaccepted Assets..... A-46
A.6.7.3 Tracking Unapproved Assets A-47
A.6.7.4 Tracking Unregistered Assets..... A-47
A.6.8 Validation Expiration Flows A-47
A.6.8.1 Asset Expiration Warning Notification..... A-49
A.6.8.2 Unregister Assets After Expiration..... A-49
A.6.8.3 Inactivate After Expiration..... A-49
A.6.8.4 Delete Assets After Expiration A-49
A.6.9 Customizing Email Notification Templates..... A-49
A.6.10 Email Notification Use Cases..... A-50
A.6.11 Known Issues A-52
A.7 Configuring JMS Servers for Oracle Enterprise Repository A-52
A.7.1 Overview of JMS for the Event Manager A-52
A.7.2 Configuring Connectivity Properties for External JMS Servers A-53
A.7.2.1 Enabling and Configuring an External JMS Server..... A-53
A.7.2.2 Configuring JMS Message Header Properties A-54

A.7.2.3 Miscellaneous JMS Properties A-55
A.7.2.4 Configuring External JMS Jar Files A-55
A.7.3 Configuring the Embedded ActiveMQ JMS Server to Use a Database A-55
A.7.3.1 Configuring JMS Durable Subscribers for Web Service Endpoints A-56
A.7.4 Configuring JMS Servers in an Oracle Enterprise Repository Cluster A-56
A.7.4.1 Enabling JMS Clustering Mode A-56
A.7.4.2 Configuring Embedded JMS Servers for Clustering A-56
A.7.4.3 Configuring External JMS Servers for Clustering A-57
A.7.5 Configuring a JMS Provider In WebSphere 6.1.0.5 ..... A-57
A.8 Monitoring and Managing Events A-60
A.8.1 Overview A-60
A.8.2 Monitoring Events A-60
A.8.2.1 Prerequisites A-60
A.8.2.2 Usage A-61
A.8.3 Cleaning Up Stored Events A-61
A.8.3.1 Prerequisites A-61
A.8.3.2 Usage A-62
A.8.3.3 Sample Event Cleanup A-62
A.8.4 Configuring Workflow A-63
A.8.4.1 Refreshing the Workflow Config File A-63
A.8.5 Encrypting Your Passwords A-64
A.9 Extending the Event Manager for Web Service Endpoints A-65
A.9.1 Overview A-65
A.9.2 Developing a Web Service Endpoint A-65
A.9.3 Web Service Operations A-66
A.9.3.1 Available Web Service Operations A-66
A.9.3.2 Selecting a Web Service Operation A-67
A.9.3.3 Developing a Notifier Plug-in A-68
A.9.4 Developing an Endpoint with an Incompatible Contract A-69
A.9.5 Creating Custom Workflows in OBPM 11g A-69
A.10 Working with OER Workflow A-69
A.10.1 Automated Asset Registration Using Automated Workflows A-71
A.10.2 Accepting a Submission A-71
A.10.3 Registering an Asset A-72
A.10.4 Completing the Tab Approval Process A-78
A.10.5 Audit Log, Reviews, and Notes A-79
A.11 Getting Started with Automated Workflows A-80
A.11.1 Steps to Configure the Oracle Enterprise Repository Event Manager A-80
A.11.2 Use Cases A-81
A.11.3 Steps to Configure and Run the Oracle Business Process Management Process Engine . A-81
A.11.3.1 Use Cases A-81
A.11.3.2 Configuring the Automated Workflows to Process a Submission Event A-82
A.11.4 Triggering an Asset Submission Event A-83
A.12 "Community Flow" Use Case A-85
B.1 Customize the Oracle Enterprise Repository Help System B-1
B.1.1 Download the Help File B-1
B.1.2 Configure the Custom Help URL Property B-1

B.2 Creating and Configuring an Upload Submission Directory B-2
B.2.1 Asset Upload System Settings B-3
B.2.2 Creating and Configuring an Artifact Store B-5
B.2.3 Selecting a New Artifact Store B-7
B.3 Creating an Asset Audit Log B-7
B.4 Plug-ins and Adapters B-7
B.5 Oracle Enterprise Repository Default Artifact Store Integration B-8
B.6 Maintaining XSLT Templates for use with the Export-to-PDF Feature B-10
B.6.1 Maintaining XSL:FO Templates for use with Oracle Enterprise Repository B-10
B.7 Support for UTF-8/Unicode Characters B-12
B.8 Asset Type Inventory B-14
B.8.1 Base Data Pack B-14
B.8.2 Harvester Solution Pack B-15
B.9 Data Rules B-17
B.9.1 Flags on Asset Type Elements B-17
B.9.2 Grandfathered Data B-17
B.9.3 Element Storage B-18
B.9.4 Elements Required to Exist on Asset Types B-19
B.9.5 Elements Tied to Functionality B-19
B.9.6 Description of the Asset Type Catalog B-22
B.10 Harvester Properties B-22

Index

Preface

Oracle Fusion Middleware Configuration Guide for Oracle Enterprise Repository provides the information needed to configure various Oracle Enterprise Repository features to support your Governance initiative.

This guide covers the basic configuration information such as setting up departments, users, projects, roles, basic access settings, etc. It also covers more advanced topics such as configuring Oracle Enterprise Repository for external authorization, setting up advanced role-based access controls, configuring Oracle Enterprise Repository to harvest from design-time and runtime environments, tailoring Oracle Enterprise Repository's automated workflows and notifications, configuring Oracle Enterprise Repository to exchange metadata with the Oracle Service Registry, Prescriptive Reuse, assets in Progress, Automated Usage Detection, and Policy Management.

In addition, this guide provides an overview of all Oracle Enterprise Repository system settings, and a workbook that allows you to record the settings that are used by your organization.

Audience

This document is intended for individuals who are running the Governance initiative. These are the individuals who will determine the Governance practices and map those practices to the features offered in Oracle Enterprise Repository.

Documentation Accessibility

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

Access to Oracle Support

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

Related Documents

For more information, see the following documents in the Oracle Enterprise Repository 11g Release 1 documentation set:

- Oracle Enterprise Repository on OTN - The home page for Oracle Enterprise Repository on Oracle Technology Network (OTN) is:

  http://www.oracle.com/technetwork/middleware/repository/overview/index-100687.html

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

  http://blogs.oracle.com/governance

Conventions

The following text conventions are used in this document:

Convention	Meaning
boldface	Boldface type indicates graphical user interface elements associated with an action, or terms defined in text or the glossary.
italic	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 for Release 11.1.1.7

This guide has been updated for release 11.1.1.7. The following table lists the sections that have been added or changed.

For a list of known issues (release notes), see "Known Issues for Oracle SOA Products and Oracle AIA Foundation Pack" http://www.oracle.com/technetwork/middleware/docs/soa-aiafp-knownissuesindex-364630.html.

Content	Changes Made
Chapter 1	
Section 1.8	Added section for Types and Viewing Types.
Chapter 6	
Section 6.2.4.2	Added Registration Status and Categorizations
Section 6.2.6.1	Added Harvesting SOA Suite Composites in a Cluster
Section 6.3.4	Added Harvesting Oracle Service Bus 11g Assets
Chapter 9	Replaced chapter with Configuring Oracle Enterprise Repository Express Workflows. Moved Configuring Oracle Enterprise Repository Workflow to Appendix A.
Chapter 15	Added chapter for Contract Management Solution Pack
Chapter 16	
All sections	Removed 10g support information for OER/BI Reports.
Section 16.1.1	Added section for BI Publisher 11g support. Configure BI Publisher 11g with Oracle Enterprise Repository 11.1.1.x
Appendix A	Added Appendix for Configuring Oracle Enterprise Repository Workflow.

1 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, "Users"
- Section 1.4, "Projects"
- Section 1.5, "Roles"
- Section 1.6, "Access Settings"
- Section 1.7, "Sessions"
- Section 1.8, "Types"

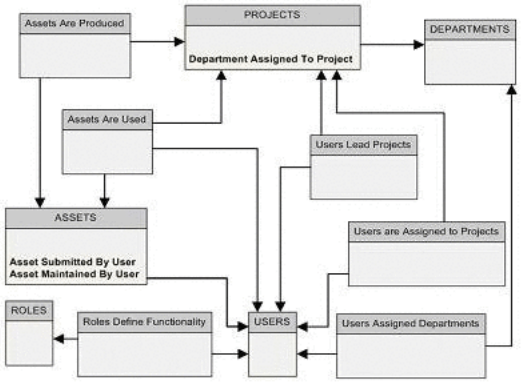
1.1 Overview

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



1.2 Departments

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

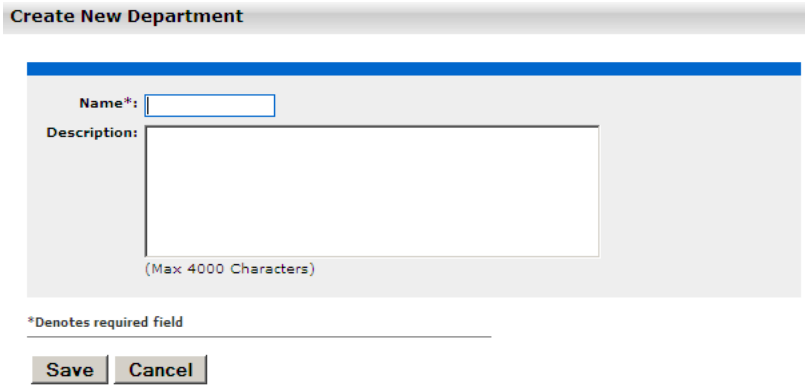
- Section 1.2.1, "Creating a New Department"
- Section 1.2.2, "Editing Department Information"

1.2.1 Creating a New Department

To create a new department, perform the following steps in 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 in each of the text boxes.

Figure 1–2 Create New Department Dialog



4. When finished, click **Save**.

1.2.2 Editing Department Information

To edit a department information, perform the following steps in 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 is displayed.
4. Click **Edit** in the department detail. The Edit Department dialog is displayed.
5. Update the text boxes as required.
6. When finished, click **Save**.

1.3 Users

This section describes the following topics:

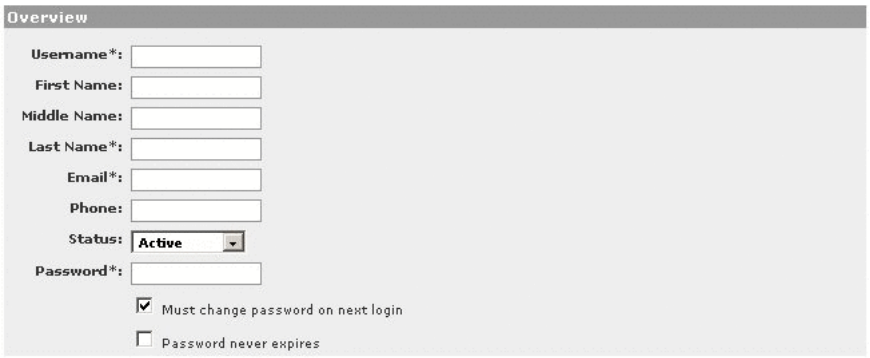
- Section 1.3.1, "Creating a New User"
- Section 1.3.2, "Viewing User Information"
- Section 1.3.3, "User Search"
- Section 1.3.4, "Cloning a User"
- Section 1.3.5, "Editing User Information"

1.3.1 Creating a New User

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

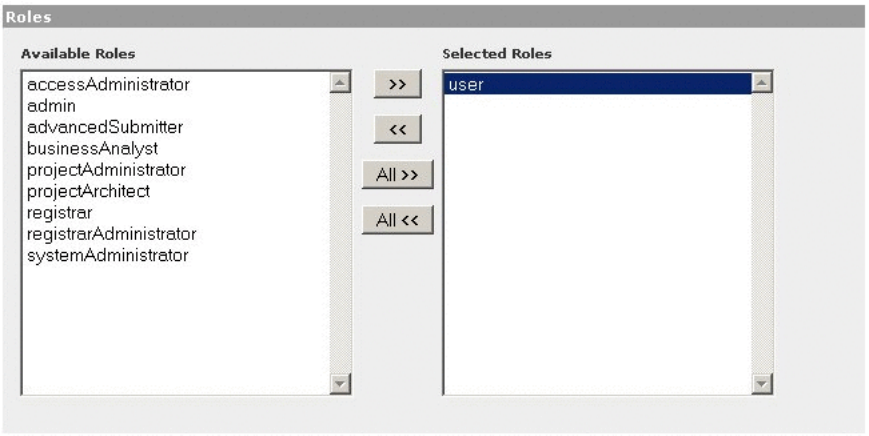
1. In the left panel, click **Users**.
2. Click **Create New** in the Users section. The Create New User dialog is displayed.
3. Enter the appropriate information in each of the text boxes in the Overview section, as shown in Figure 1–3.

Figure 1–3 Overview Section



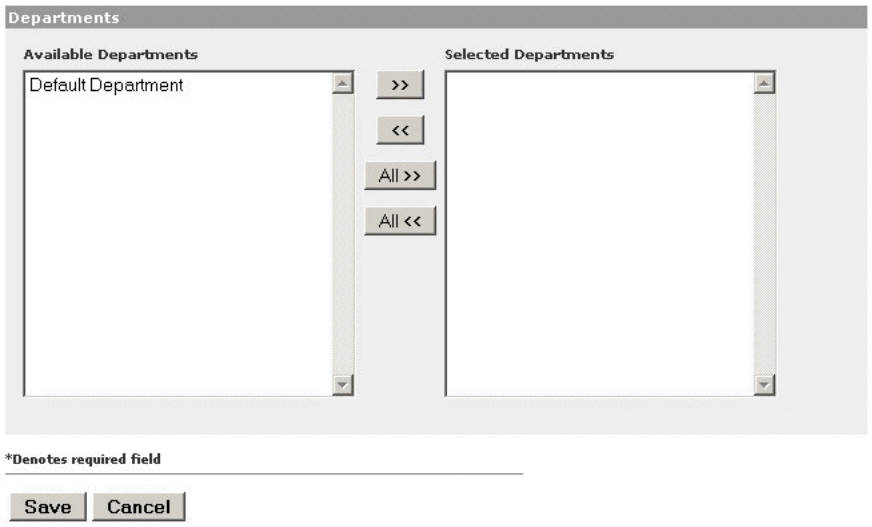
4. Select **Active** in the Status list.
5. In the Roles section, assign roles to the new user by using the >> and << buttons to move items from the Available Roles column to the Selected Roles column, as shown in Figure 1–4. (The User role is the default role for all new users.)

Figure 1–4 Roles Section



6. In the Departments section, assign the new user to departments by using the >> and << buttons to move items from the Available Departments column to the Selected Departments column, as shown in Figure 1–5.

Figure 1–5 Departments Section



7. When finished, click **Save**.

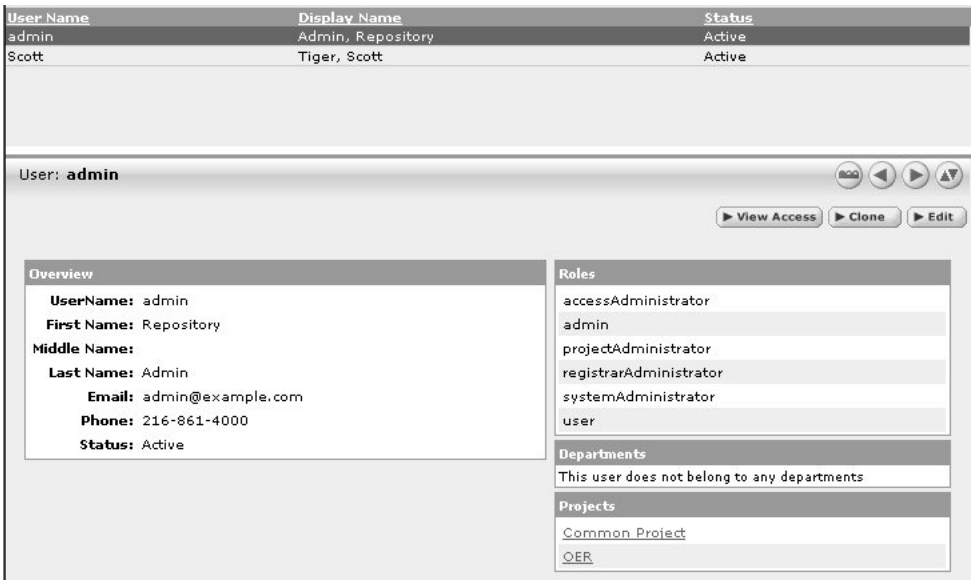
1.3.2 Viewing User Information

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

1. In the Users section, use **Search** or **List All** to locate the user(s) to be viewed.

   The list of users is displayed in the main pane, as shown in Figure 1–6.

Figure 1–6 List of Users



2. Select a user from the list. The user's information is displayed in the lower pane.
3. Click **Hide/Display Search Results** button to expand the User detail to fill the main pane.
4. Click the first button in Figure 1–6 to switch to the tabbed view of the User detail.

Figure 1–7 User Details Page

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

1.3.3 User Search

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

1. Enter appropriate text in the Name text box.
2. Use the Department, Role, and Status lists as appropriate to narrow the search.
3. Click the **Search** button. Search results appear in the list in the upper frame of the main pane.

1.3.4 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 Users section, use Search or List All to locate the user to be cloned.
2. 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.
3. Enter the appropriate information in each of the text boxes in the Overview section.
4. If necessary, edit the information in the in the Roles and Departments sections by using the >> and << buttons to move items between the Available and Selected columns.
5. When finished, click **Save**.

1.3.5 Editing User Information

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

1. In the Users section, use Search or List All to locate the user(s) to be edited. The list of users opens in the main pane.
2. Select a user from the list. The user's information is displayed in the lower pane.
3. 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).
4. Make changes as necessary in the Overview, Roles, and Departments sections.
5. When finished, click **Save**.

1.4 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 1.4.1, "Viewing Project Details"

- Section 1.4.2, "Creating a Project"
- Section 1.4.3, "Editing a Project"
- Section 1.4.4, "Reassigning Users/Usage"
- Section 1.4.5, "Locating a Project"
- Section 1.4.6, "Closing a Project"
- Section 1.4.7, "Adding a Relationship to a Project"

1.4.1 Viewing Project Details

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

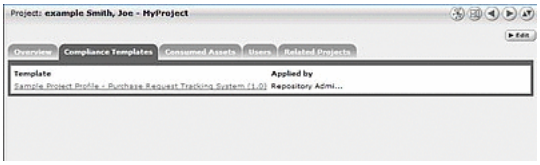
Overview

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 1–8 Compliance Templates Tab



Consumed Assets

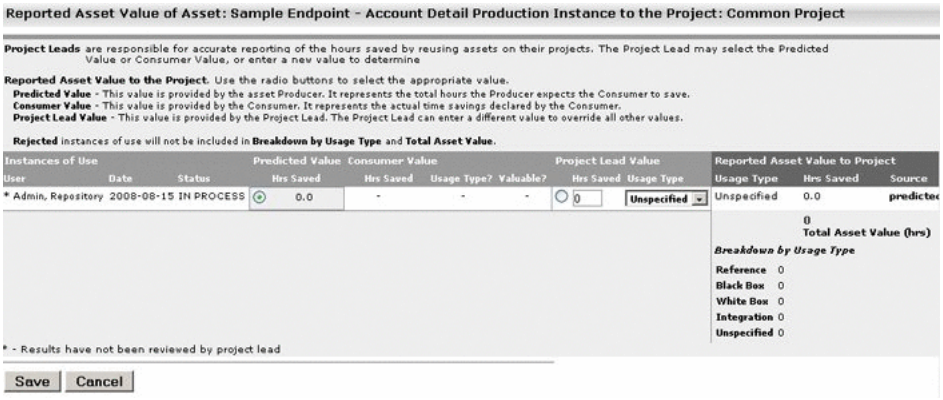
Lists any assets used in (or under consideration for use in) the project.

Figure 1–9 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 1–10 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 1–11 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 value to assets consumed by the project.

Figure 1–12 Users Tab

Related Projects

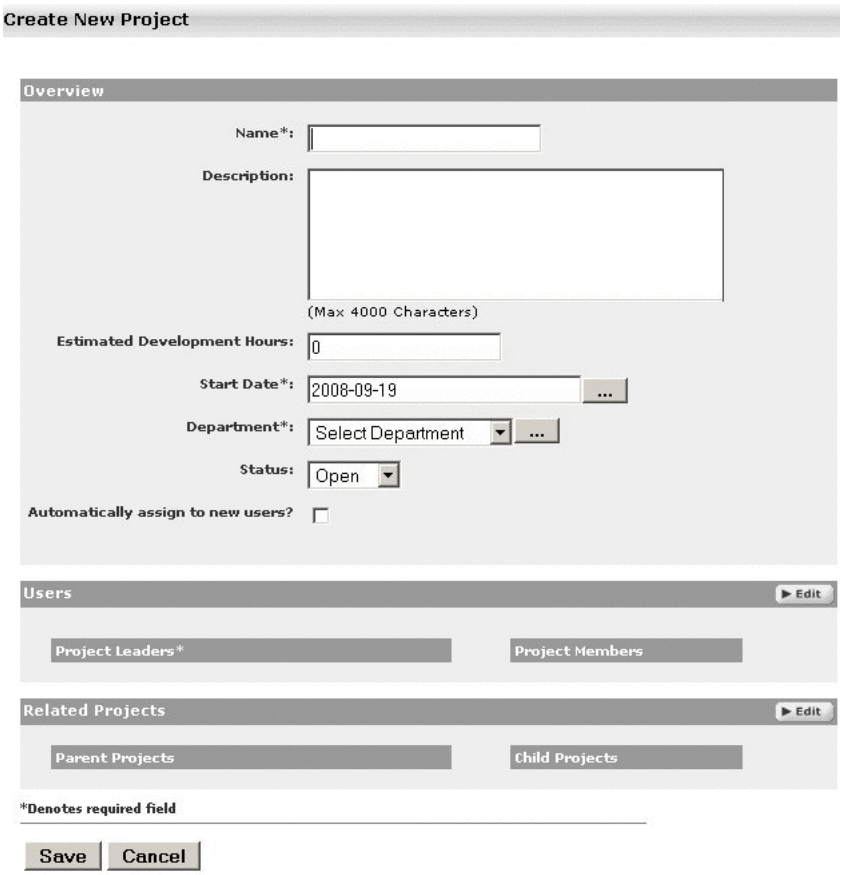
When enabled, lists any related projects, and defines the relationships in parent/child terms.

1.4.2 Creating a Project

This procedure is performed in the Projects screen.

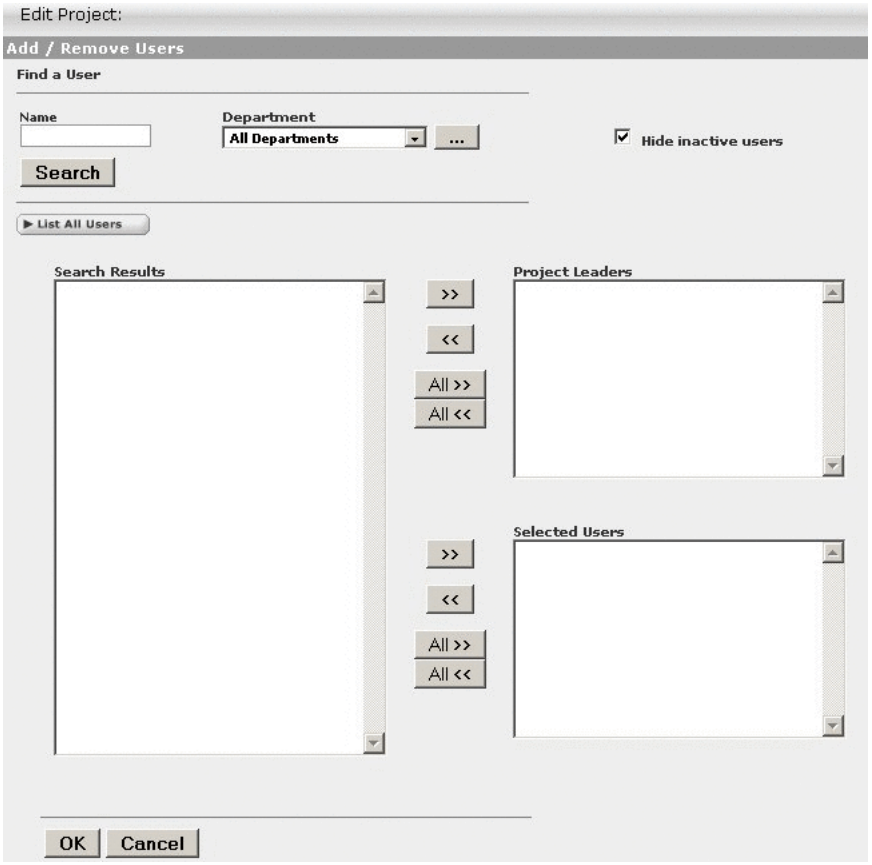
1. In the left panel, click Create **New**. The Create New Project dialog is displayed.

Figure 1–13 Create New Project



2. Enter the appropriate information in the Name, Description, and Estimated Development Hours text boxes.
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. There are two options that you can select from - Open and Closed. 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 1–14.

Figure 1–14 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 >> and << icons 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 1–15 User Dialog

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

Figure 1–16 Overview Tab

1.4.3 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 1.4.2, "Creating a Project").

1.4.4 Reassigning Users/Usage

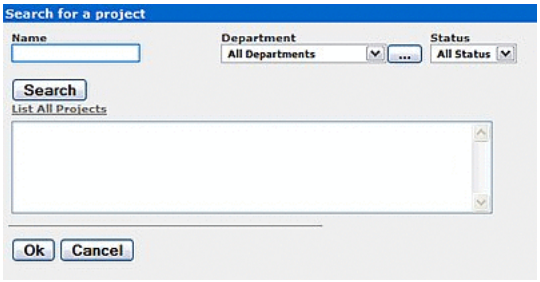
This procedure is performed in the Projects page.

1. Select the Users tab.

Figure 1–17 Users Tab

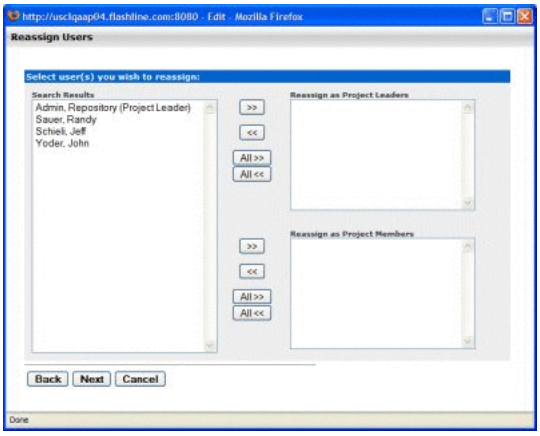
2. Click **Reassign Users/Usage**. The Reassign dialog is displayed.
3. Use the options to select the appropriate action.
4. Click **Choose a New Project**. The Search for a Project dialog is displayed.

Figure 1–18 Search for a Project



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 1–19 Reassign Users



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

1.4.5 Locating a Project

This procedure is performed on the Projects page.

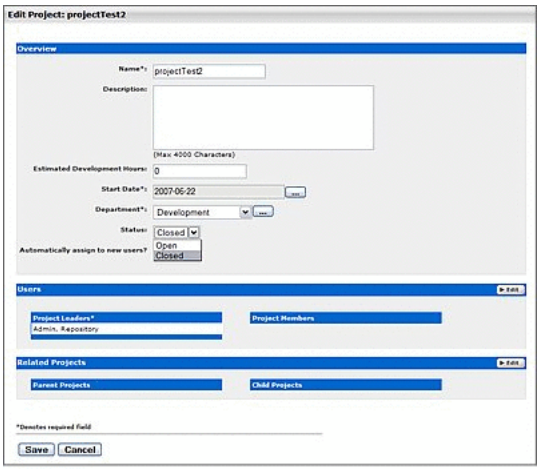
1. Enter a keyword or search term in the Name text box.
2. Use the Department and Status lists as necessary to filter the search.
3. Click **Search**. Search results are listed in the main pane.

1.4.6 Closing a Project

This procedure is performed on the Projects page.

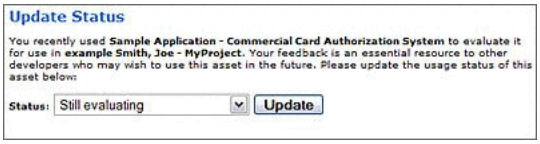
1. Select the project to be closed. The project detail is displayed in the bottom frame.
2. Click **Edit**. The Edit Project dialog is displayed.
3. Select **Closed** from the Status list in the Overview section of the Edit Project dialog, as shown in Figure 1–20.

Figure 1–20 Edit Project Dialog



4. Click **Save**. If the project has consumed assets, the Update Usage Status dialog is displayed, as shown in Figure 1–21. The consumed assets are listed in this dialog, each with a Status list.

Figure 1–21 Update Status



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.

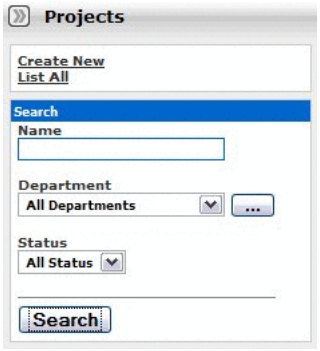
1.4.7 Adding a Relationship to a Project

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.

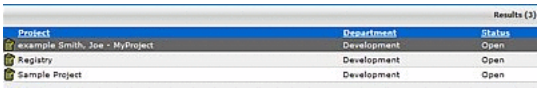
1. Click **List All** on the Projects page, as shown in Figure 1–22.

Figure 1–22 Projects Page



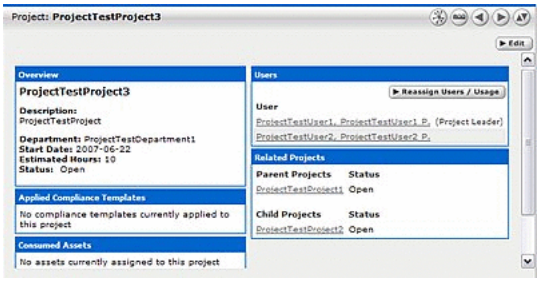
A list of projects appears in the top section of the main pane, as shown in Figure 1–23.

Figure 1–23 List of Projects



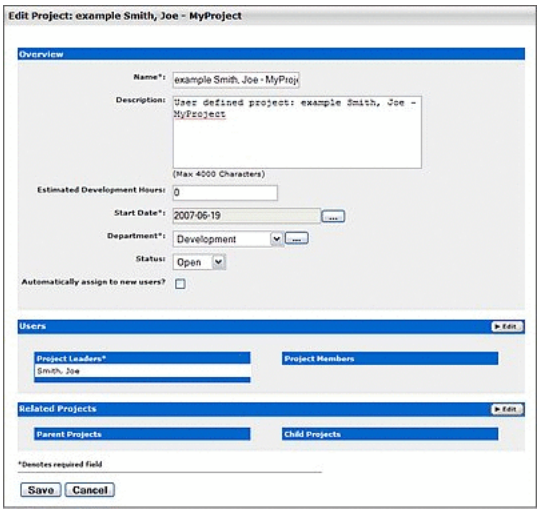
2. Select the project to be edited. The Project Detail pane is displayed in the bottom section of the main pane, as shown in Figure 1–24.

Figure 1–24 Project Detail Pane



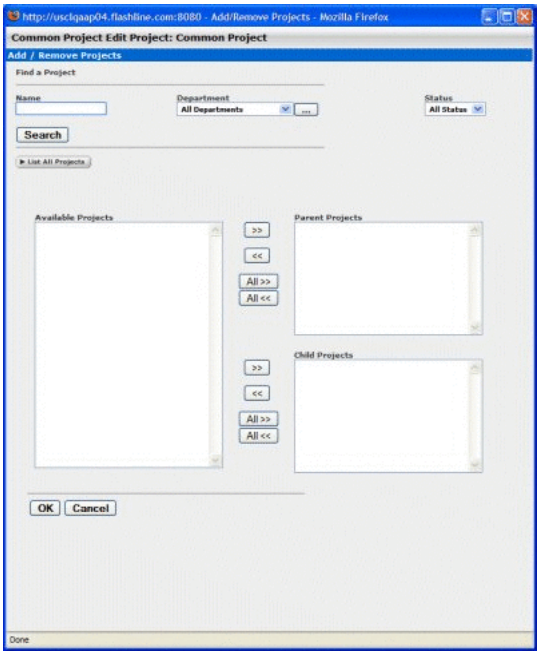
3. Click **Edit** in the Project Detail pane. The Edit Project dialog is displayed, as shown in Figure 1–25.

Figure 1–25 Edit Project Dialog



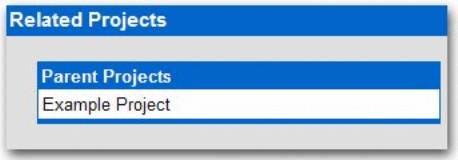
4. Click **Edit** in the Related Projects section of the Edit Projects dialog. The Add/Remove Projects dialog is displayed, as shown in Figure 1–26.

Figure 1–26 Add / Remove Projects



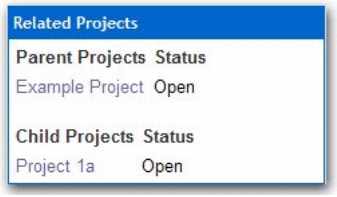
5. Click **List All Projects**. A list appears in the Available Projects window.
6. Select the project to be added and use the >> icon to move it to either the Parent Projects or the Child Projects category window, as appropriate. The project appears in the selected window.
7. Click **OK** to close the 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 1–27.

Figure 1–27 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 1–28.

Figure 1–28 Related Projects Section



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

1.5 Roles

Clicking the Roles link in the Oracle Enterprise Repository menu bar 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* 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 1.5.1, "User Role Descriptions"
- Section 1.5.2, "Creating a New Role"
- Section 1.5.3, "Editing a Role"
- Section 1.5.4, "Deleting a Role"

1.5.1 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

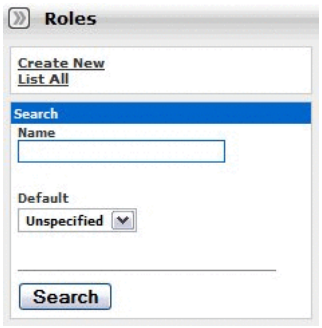
  (The system administrator role should not be assigned to general users.)

1.5.2 Creating a New Role

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

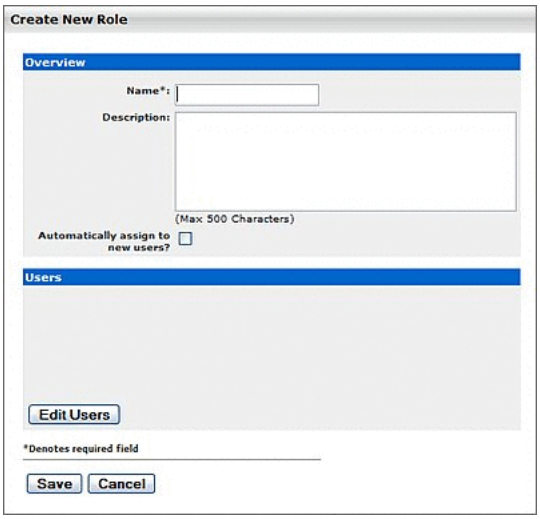
1. In the left panel, click **Roles**. The Roles section is displayed, as shown in Figure 1–29.

Figure 1–29 Roles Section



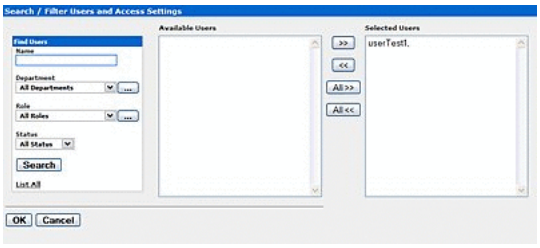
2. Click **Create New**. The Create New Role dialog is displayed, as shown in Figure 1–30.

Figure 1–30 Create New Role Dialog



3. In Figure 1–30, 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 1–31.

Figure 1–31 Add New Users Dialog



6. Use **Search** or click **List All** to locate users.
7. Use the >> and << buttons to move users between the Available Users and Selected Users columns.

8. When finished, click **Save**.

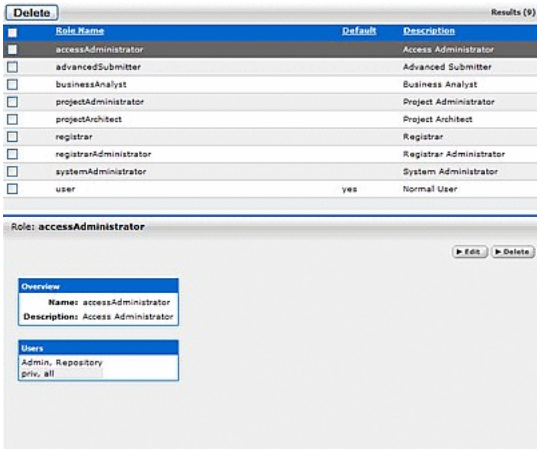
To add access settings to the role, see Section 1.6.1, "Granting Permissions to a Role".

1.5.3 Editing a Role

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

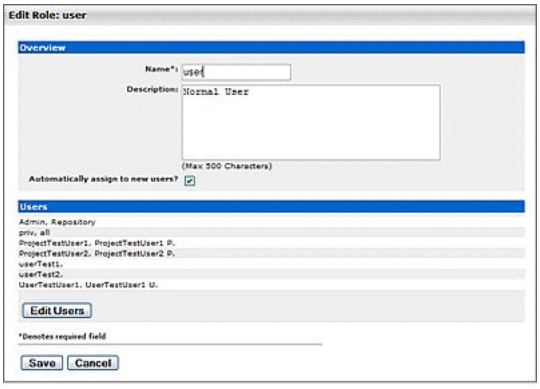
1. In the left panel, click **Roles**. A list of roles is displayed in the main pane, as shown in Figure 1–32.

Figure 1–32 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 1–33.

Figure 1–33 Edit Role: User Dialog



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

1.5.4 Deleting a Role

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

(Requires the Delete permission. See 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.

1.6 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.

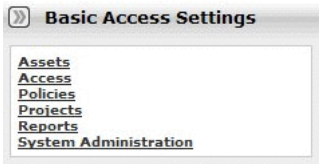
- Section 1.6.1, "Granting Permissions to a Role"
- Section 1.6.2, "Editing Permissions"
- Section 1.6.3, "Verifying User Permissions"
- Section 1.6.4, "Security Considerations"

1.6.1 Granting Permissions to a Role

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

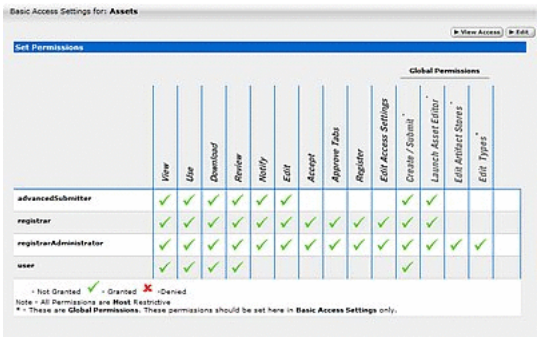
1. In the left panel, click **Basic Access Settings**.
2. Select the appropriate grouping from the list in the Basic Access Settings section.

Figure 1–34 Basic Access Settings



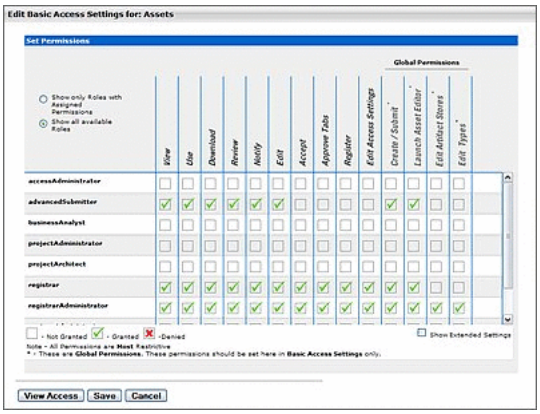
The selected grouping's access settings are displayed in the main pane.

Figure 1–35 Set Permissions



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

Figure 1–36 Set Permissions



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

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

Figure 1–37 Set Permissions

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**.

1.6.2 Editing Permissions

See: Section 1.6.1, "Granting Permissions to a Role"

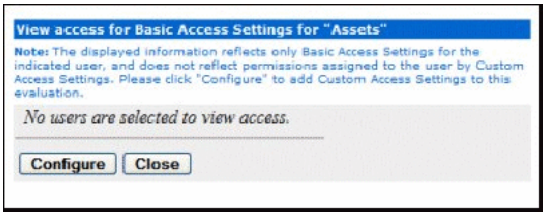
1.6.3 Verifying User Permissions

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.

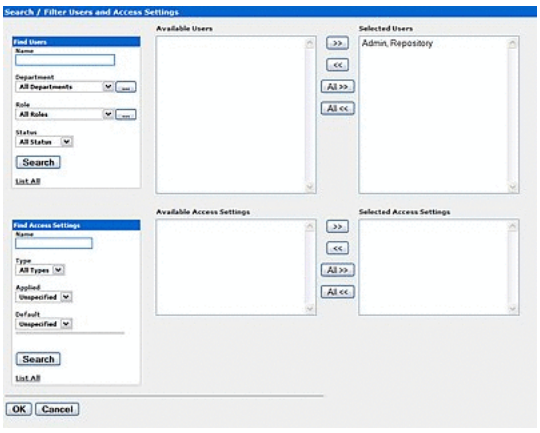
   The selected grouping's access settings are displayed in the main pane.
3. Click the **View Access** button (upper right corner of the main pane). A message dialog is displayed.

Figure 1–38 View Access for Basic Access Settings for Assets



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

Figure 1–39 Search / Filter Users and Access Settings



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 >> button 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 1–40 User Access Settings

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**.

1.6.4 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.

1.6.4.1 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.

1.6.4.2 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.
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 providers 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.

1.7 Sessions

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

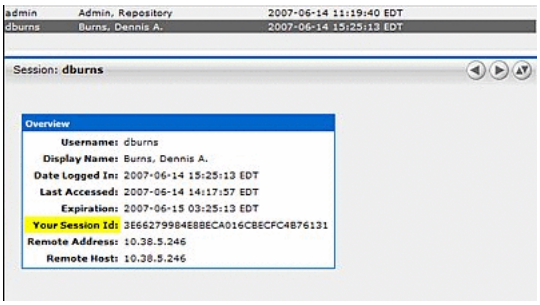
- Section 1.7.1, "Viewing Sessions"
- Section 1.7.2, "Deleting a Single Session"
- Section 1.7.3, "Deleting Multiple Session"

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 1–41.

Figure 1–41 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 Session

1. In the list of sessions in the main pane, select each session to be deleted. (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 session detail. A confirmation dialog is displayed.
3. Click **OK**. The session is deleted.

(Requires the Delete permission. See Section 1.6, "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 the *Oracle Fusion Middleware Upgrade Guide for Oracle Enterprise Repository*.

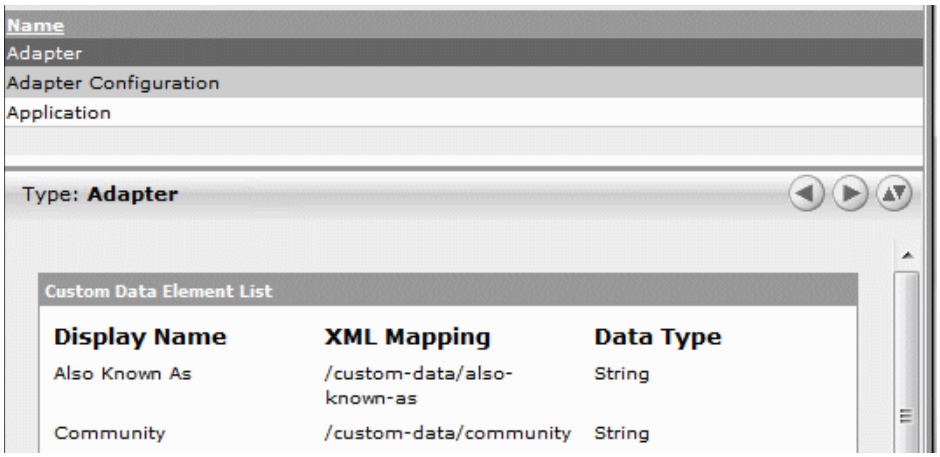
1.8.1 Viewing Types

Use the Types section for searching and viewing attributes of each individual Type. You can search by Name and Archetype. You can view Types from the Oracle Enterprise Repository Administration screen.

- In the Types section, use Search or List All to locate the type(s) you want to view.

  The list of users displays on the main pane, as shown in Figure 1–42.

Figure 1–42 Types Viewer



2

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 2.1, "Overview"
- Section 2.2, "Rebranding Oracle Enterprise Repository"
- Section 2.3, "Role/Department-based Homepage Display"
- Section 2.4, "Install Oracle Enterprise Repository Solution Packs"

2.1 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 2.1.1, "Accessing the Oracle Enterprise Repository Home Page"
- Section 2.1.2, "Customizing the Oracle Enterprise Repository Home Page"

2.1.1 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.

> **Note:** If your organization requires a user name and password each time you log into Oracle Enterprise Repository, ensure that the Enable Automatic Login box is unchecked.

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.

2.1.2 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.

For more information about the Oracle Enterprise Repository home page, see *Oracle Fusion Middleware User's Guide for Oracle Enterprise Repository*.

2.2 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 2.2.1, "Start with CSS"
- Section 2.2.2, "Customizing Skins"

2.2.1 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 2–1.

Figure 2–1 Cascading Style Sheets

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

2.2.2 Customizing Skins

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

- Section 2.2.2.1, "Choosing a Skin"
- Section 2.2.2.2, "Creating a Skin"
- Section 2.2.2.3, "Images"
- Section 2.2.2.4, "Activating Your Custom Skin"
- Section 2.2.2.5, "Installing a Oracle-supplied Custom Skin in Oracle Enterprise Repository"
- Section 2.2.2.6, "Using the Directory Structure"

2.2.2.1 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.

2.2.2.2 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 *<Oracle_home>/ user_projects\applications\base_domain\applications\oer_11.1.1.4.0\oer-web\skins* directory:

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

2.2.2.3 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 *<Oracle_home>\user_projects\applications\base_domain\applications\oer_11.1.1.4.0\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.

2.2.2.4 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.

2.2.2.5 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 *<Oracle_home>\user_projects\applications\base_domain\applications\oer_11.1.1.4.0\oer-app\WEB-INF\classes\skin.properties* file.
4. Copy the new `skin.properties` file from the custom skin ZIP archive to the Oracle Enterprise Repository *<Oracle_home>\user_projects\applications\base_domain\applications\oer_11.1.1.4.0\oer-app\WEB-INF\classes* directory.
5. Replace the existing Oracle Enterprise Repository *<Oracle_home>\user_projects\applications\base_domain\applications\oer_11.1.1.4.0\oer-app\WEB-INF\classes\skin.properties* file with new file.
6. Move the extracted custom skin directory containing the images and CSS files into the *<Oracle_home>/ user_projects\applications\base_domain\applications\oer_11.1.1.4.0\oer-web\skins* directory. The `skin.properties` file points to this directory to find the skin resources.
7. Restart the application server to apply the changes.

2.2.2.6 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.)

2.3 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 2.3.1, "Installation"

2.3.1 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 2–1.
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 2–1 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.

2.4 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 **11.1.1.x.x-OER-Harvester-Solution-Pack.zip**, which is bundled with the Oracle Enterprise Repository 11*g* 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 the Import/Export Tool chapter in *Oracle Fusion Middleware User's Guide for Oracle Enterprise Repository*.

Contract Management Solution Pack

The Contract Management Solution Pack is bundled with the Oracle Enterprise Repository 11*g* 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 the Import/Export Tool chapter in *Oracle Fusion Middleware User's Guide for Oracle Enterprise Repository*.

3

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, eTrust Single Sign-On, and Container Managed Setup.

This chapter contains the following sections:

- Section 3.1, "Overview"
- Section 3.2, "eTrust Single Sign-On"
- Section 3.3, "Container Managed Setup"

3.1 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 3.1.1, "LDAP/Active Directory"
- Section 3.1.2, "Enable LDAP Integration System Properties"
- Section 3.1.3, "Modify LDAP/Active Directory Properties"
- Section 3.1.4, "Security Considerations"
- Section 3.1.5, "LDAP Property Examples"

3.1.1 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

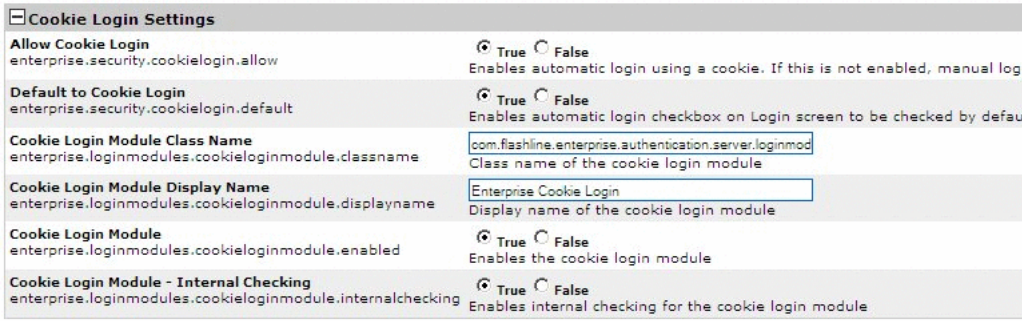
using LDAP/AD, that person has the ability to configure and administer the application properly. This user account should NOT be used for daily activities.

3.1.2 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. Locate the **Cookie Login Settings** group in the Enterprise Authentication section.

Figure 3–1 Cookie Login Settings



3. Ensure that the **Allow Cookie Login** setting is set to `False`.
4. Use the System Settings Search box to easily locate each of the following settings.
5. Enter `enterprise.authentication.ldap.enabled` into the Search box. Set the value to `True` and click **Save**.

   Change the settings as indicated below:

   - Default to Cookie Login
     - Set to `False`.
   - Unapproved User Login
     - Set to `True`.
   - Cookie Login Module
     - Set to `False`.
   - Cookie Login Module - Internal Checking
     - Set to `False`.
   - Plug-in Login Module Class Name
     - Enter `com.oer.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`.

6. Click **Save** when finished.

3.1.3 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.

3.1.4 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.

3.1.4.1 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.

- 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`

3.1.5 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 3–1 describes the LDAP /Active Directory properties with its examples.

Table 3–1 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=exa mple,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.givennameAttri b	givenname	ldap.givennameAttrib	givenName
ldap.surnameAttrib	sn	ldap.surnameAttrib	sn
ldap.telephoneAttrib	telephonenumber	ldap.telephoneAttrib	telephoneNumber
ldap.deptAttrib	department	ldap.deptAttrib	department

Table 3–2 describes the custom and common LDAP properties.

Table 3–2 Custom and Common LDAP Properties

Custom and Common Properties Regardless of implementation	
ldap.rbac.mapperclass	com.oer.enterprise.authentication.server.loginmodule.LDAPMapperImpl
ldap.deptAttrib	department
ldap.rbac.roleAttrib	roles
ldap.allow-user-creation	true
ldap.enable-synch-roles	true
ldap.enable-synch-depts	true

3.2 eTrust Single Sign-On

This section describes how to use eTrust Single Sign-On to authenticate the users in Oracle Enterprise Repository.

This section contains the following topics:

- Section 3.2.1, "Overview"
- Section 3.2.2, "Configure Oracle Enterprise Repository For Use With Single Sign-On Authentication"
- Section 3.2.3, "Advanced Options"

3.2.1 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 such as eTrust Single Sign-On.

3.2.2 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 3.2.2.1, "Enable Single Sign-On Integration System Properties"
- Section 3.2.2.2, "Using the Oracle Enterprise Repository SSO Integration with Basic Authentication"
- Section 3.2.2.3, "Modify Application Property Files Manually"

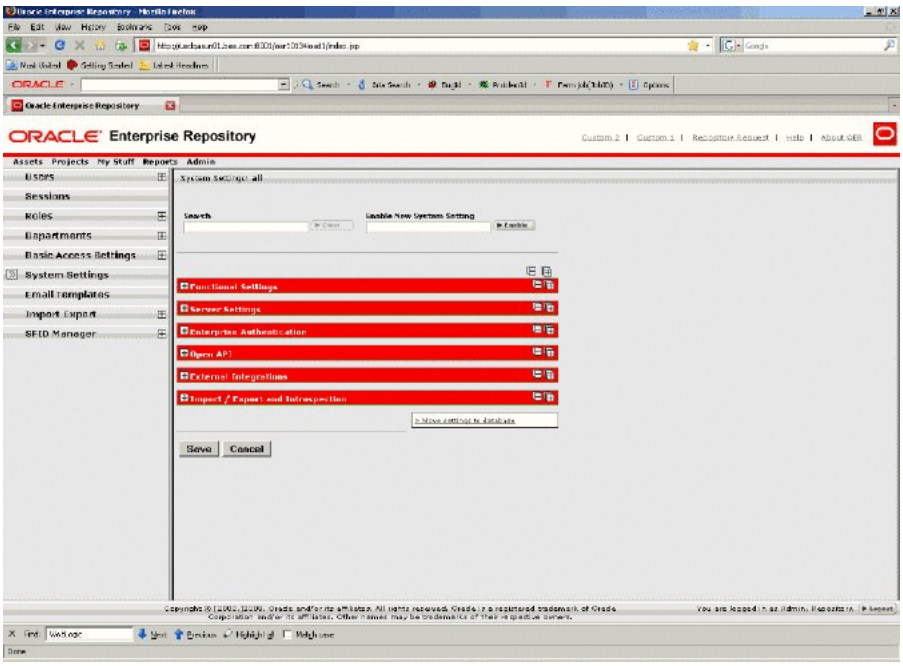
> **Note:** The Fallback authentication works only with REX API.

3.2.2.1 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 3–2.

Figure 3–2 Oracle Enterprise Repository System Settings Screen



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 3–3.

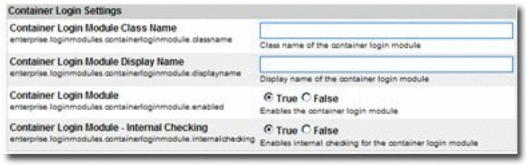
Figure 3–3 Enable New System Settings Dialog

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 3–4.

Figure 3–4 JWS Pass All Cookies

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 3–5.

Figure 3–5 Container Login Settings Section



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**.

3.2.2.2 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**.

3.2.2.3 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 synching 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. Most Single Sign-On web agent applications are deployed against an HTTP server that is separate from the Application Server. In this scenario, an AJP type connector (mod_jk/mod_jk2 for Apache HTTP Servers, mod_was_ap20_http for IBM HTTP Server, etc.) will link the HTTP server to the application server. Typically, the HTTP server runs on a separate machine for performance or resource pooling reasons. In this scenario it is necessary to modify the cmee.properties file to reflect the new name for your application.

   - Edit the `cmee.properties` file in WEB-INF/classes.
     - Original Configuration (Tomcat with Coyote)

       - **cmee.server.paths.image**=`http\://tomcat.example.com\:8080/oer-web/images`

       - **cmee.server.paths.jsp**=`http\://tomcat.example.com\:8080/oer`

       - **cmee.server.paths.servlet**=`http\://tomcat.example.com\:8080/oer`

       - **cmee.server.paths.jnlp-tool**=`http\://tomcat.example.com\:8080/oer-web/webstart`

       - **cmee.server.paths.resource**=`http\://tomcat.example.com\:8080/oer-web`

       - **cmee.enterprisetab.homepage**=`http\://tomcat.example.com\:8080/oer/custom/home.jsp`

       - **cmee.assettab.asset-detail-page**=`http\://tomcat.example.com\:8080/oer/cmee/index.jsp`

     - New configuration (Apache HTTP with mod_jk2 to Tomcat)

       - **cmee.server.paths.image**=`http\://apache.example.com/oer-web/images`

       - **cmee.server.paths.jsp**=`http\://apache.example.com/oer`

       - **cmee.server.paths.servlet**=`http\://apache.example.com/oer`

       -**cmee.server.paths.jnlp-tool**=`http\://apache.exampe.com/oer-web/webstart`

       - **cmee.server.paths.resource**=`http\://apache.example.com/oer-web`

       - **cmee.enterprisetab.homepage**=`http\://apache.example.com/oer/custom/home.jsp`

       - **cmee.assettab.asset-detail-page**=`http\://apache.example.com/oer/cmee/index.jsp`

     - In this example, the new URL to connect to the Repository is: `http://apache.example.com/oer/index.jsp`

3. Restart the Oracle Enterprise Repository application.

3.2.3 Advanced Options

The following options add functionality for assigning default roles, new user creation/notification, syncing departments, and syncing roles:

- Section 3.2.3.1, "Creating/Assigning Default Roles for New Users"
- Section 3.2.3.2, "Create New Users/Allow Unapproved Users"
- Section 3.2.3.3, "Enable Unapproved/New User Login"

- Section 3.2.3.4, "New User Notification"
- Section 3.2.3.5, "Syncing Departments"
- Section 3.2.3.6, "Syncing Roles"
- Section 3.2.3.7, "Enable Debug Logging"

3.2.3.1 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**.

3.2.3.2 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.

3.2.3.3 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`

3.2.3.4 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`

3.2.3.5 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
  ```

3.2.3.6 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
```

3.2.3.7 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
```

3.3 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 3.3.1, "Overview"
- Section 3.3.2, "Configure Oracle Enterprise Repository for Container Managed Authentication"

- Section 3.3.3, "Modify the Web Application's Web.xml File to Allow for Container Authentication"

3.3.1 Overview

The container is configured appropriately with a Realm or Authenticator back-end prior to enabling the values within the Oracle Enterprise Repository application.

3.3.2 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 3–5.
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**.

3.3.3 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.

4

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 4.1, "Overview"
- Section 4.2, "Security Considerations"
- Section 4.3, "Basic Concepts"
- Section 4.4, "Access Definitions"
- Section 4.5, "Process Overview"
- Section 4.6, "Granting and Denying Permissions"
- Section 4.7, "Configuring Access Settings for Existing Roles"
- Section 4.8, "Role-based Access Control Use Cases"

4.1 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.

4.2 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.

4.2.1 Access Settings and the Asset Editor

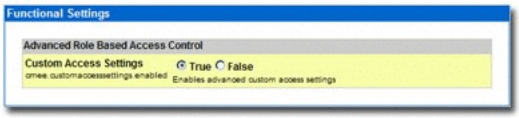
Depending on the specific Access Settings in use, users with access to the Asset Editor may be able to see relationships to assets that are otherwise invisible to them in Oracle Enterprise Repository. In this situation, the name of the hidden asset is visible, as is the detail of the relationship between 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.

4.2.2 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 4–1.

Figure 4–1 Functional Settings



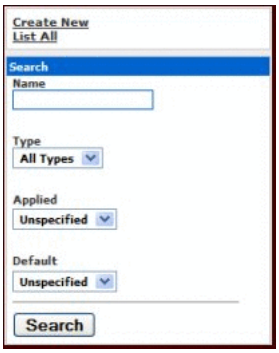
4. Repeat the process to enable each of the following properties:
   - `cmee.customaccesssettings.file`
   - `cmee.customaccesssettings.asset`

   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 4–2.

Figure 4–2 Custom Access Settings



4.3 Basic Concepts

This section describes the basic concepts of Role-based Access Control. This section describes the following topics:

- Section 4.3.1, "Roles"
- Section 4.3.2, "Access Settings"

4.3.1 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.

4.3.2 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.

4.4 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).

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
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

Permission	Description	BAS	CAS
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

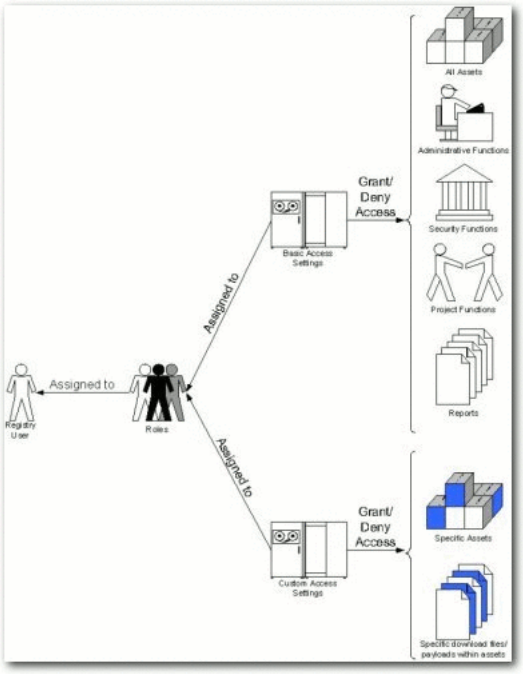
4.5 Process Overview

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 4–3 describes the process overview.

Figure 4–3 Process Overview



4.6 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 4.6.1, "Grant/Deny Access to Specific Assets"
- Section 4.6.2, "Grant/Deny Access to Specific Download Files Within an Asset"

4.6.1 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.

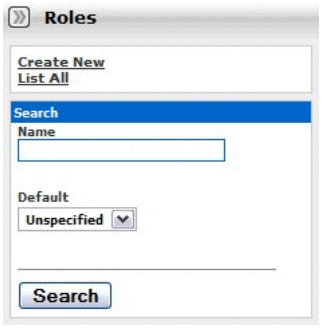
4.6.1.1 Prerequisites

- 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.

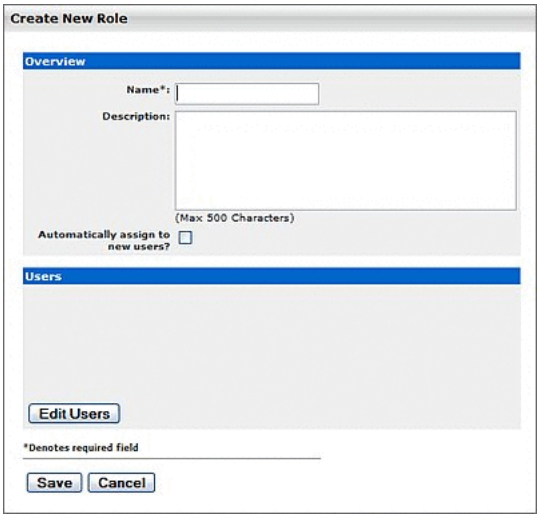
1. Click **Roles** in the left sidebar. The Roles section is displayed, as shown in Figure 4–4.

Figure 4–4 Roles Section



2. Click **Create New**. The Create New Role dialog is displayed, as shown in Figure 4–5.

Figure 4–5 Create New Role Dialog



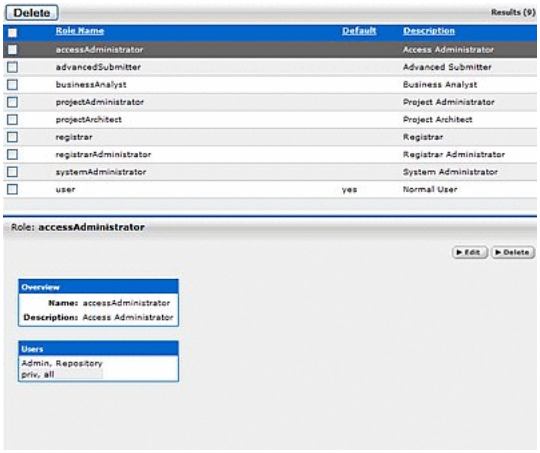
3. Enter the appropriate information in the Name and Description fields.
4. Click **Save**.

4.6.1.2 Create the Asset CAS

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

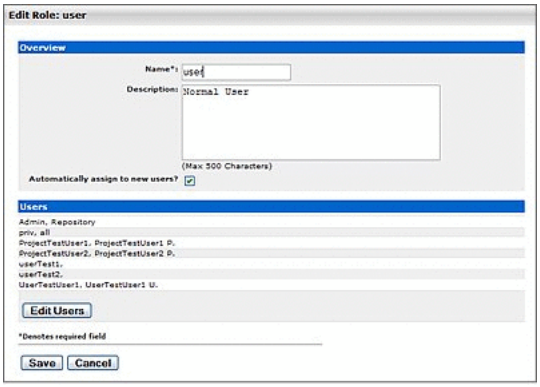
1. Click **Roles** in the left sidebar. A list of available Roles is displayed in the main pane, as shown in Figure 4–6.

Figure 4–6 List of Roles



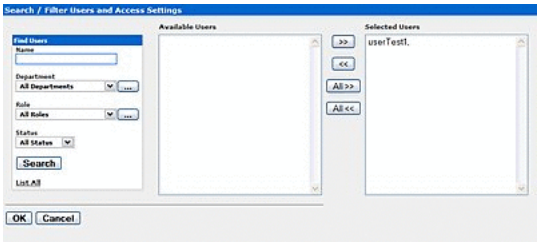
2. From the list, select the role to be edited. The role detail is displayed in the bottom frame of the main pane.
3. Click the **Edit** button in the role detail. The Edit Role dialog is displayed, as shown in Figure 4–7.

Figure 4–7 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 4–8.

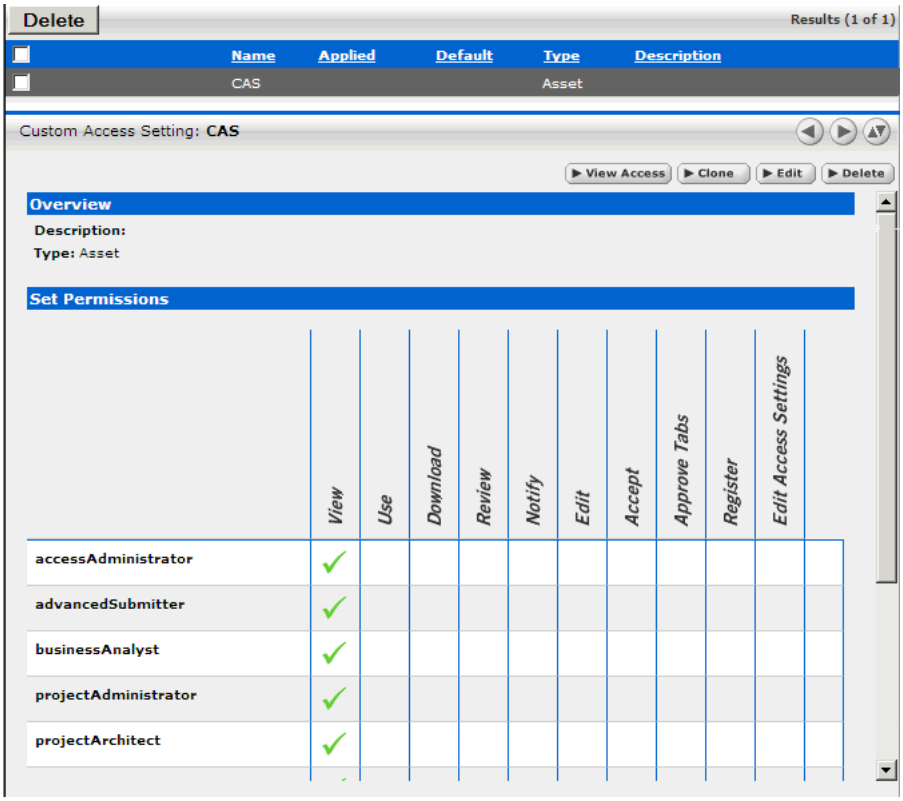
Figure 4–8 Search / Filter Users and Access Settings



5. Use Search or click **List All** to populate the Available Users column.
6. Use the << and >> buttons to move users from the Available Users to the Selected Users column.

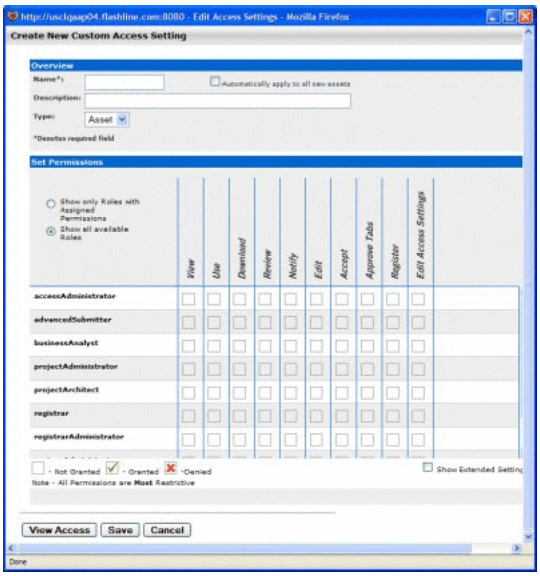
7. When finished, click **OK**. The Add Users dialog closes, and the selected users are listed in the Users section of the Edit Role dialog.
8. Click **OK**. The Edit Role dialog closes.
9. Click **Custom Access Settings** in the sidebar on the Admin screen. The Custom Access Settings section is displayed, as shown in Figure 4–9.

Figure 4–9 Custom Access Settings Section



10. Click **Create New**. The Create New Custom Access Setting dialog is displayed, as shown in Figure 4–10.

Figure 4–10 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 4–11.

Figure 4–11 Right Mark

    - A second click denies permission, as shown in Figure 4–12.

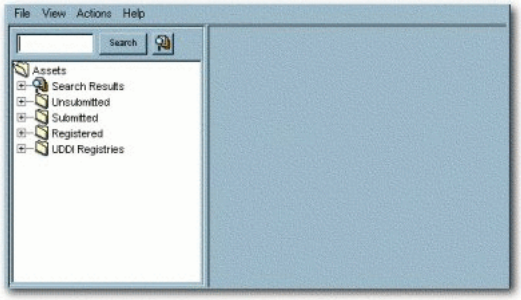
Figure 4–12 Cross Mark

    - A third click clears the option (permission Not Granted):

Figure 4–13 Blank

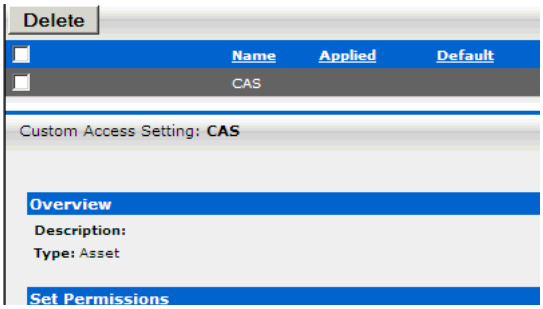
14. When finished, click **Save**. The Create New Custom Access Setting dialog is displayed.
15. Click **Assets** in the Oracle Enterprise Repository menu bar.
16. Click **Edit / Manage Assets**. The Asset Editor is displayed, as shown in Figure 4–14.

Figure 4–14 Asset Editor Window



17. Locate and open (or create, if necessary) the asset to which the new Custom Access Settings are applied.
18. Click the asset's Administration tab, as shown in Figure 4–15.

Figure 4–15 Administration Tab



19. Use the << and >> 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.

4.6.2 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

4.6.2.1 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**.

4.6.2.2 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.

4.7 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.

4.7.1 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)

4.7.2 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.

4.7.2.1 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

     ---

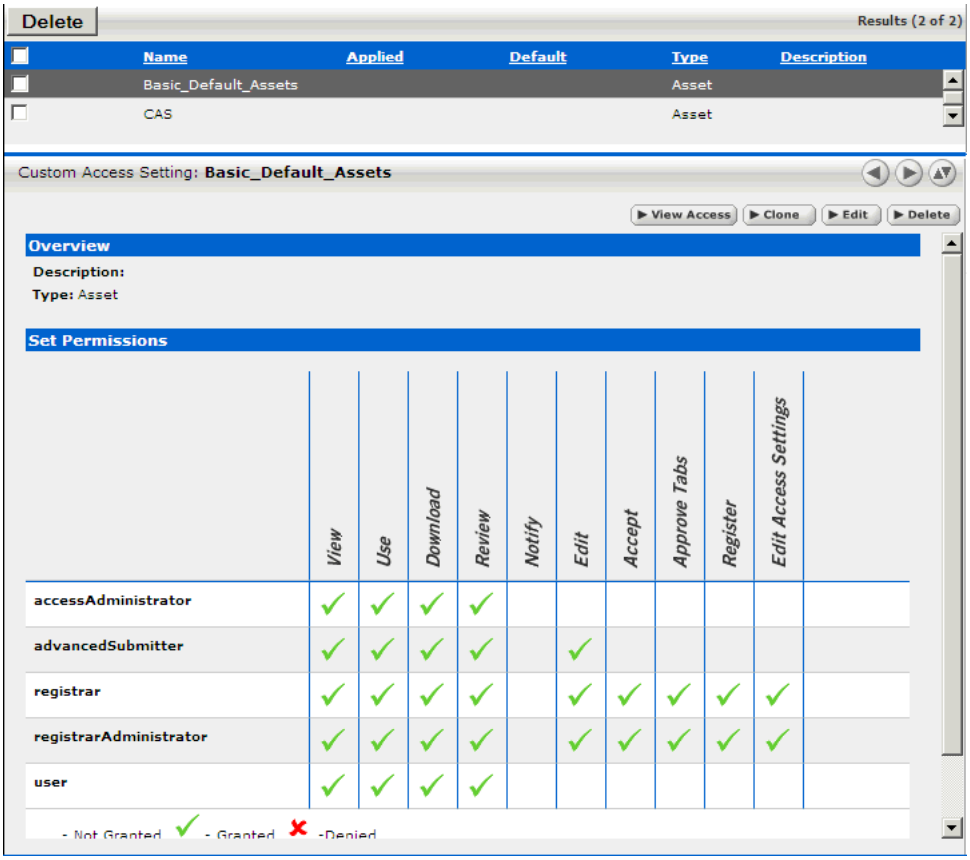
     Note: Under this configuration, Registrars can view and change Access Settings for all assets. To restrict this privilege, leave the Edit Access Settings option blank for the Registrar role.

   - **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 4–16 illustrates the resulting Custom Access Setting.

Figure 4–16 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 4–1 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.

4.7.2.2 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

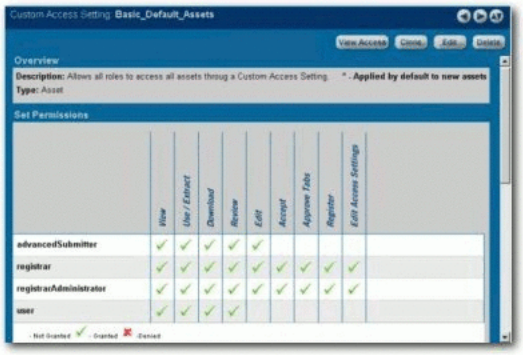
1. Create a CAS for assets called Basic_Default_Assets. Add the following roles and associated permissions:
   - **Role**: User
   - **Permissions**: View, Use, Review
   - **Role**: Advanced Submitter
   - **Permissions**: View, Use, Review, Edit
   - **Role**: Registrar
   - **Permissions**: View, 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 4–17.

Figure 4–17 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 4–2 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.

4.8 Role-based Access Control Use Cases

This section describes the role-based access control use cases.

This section contains the following topics:

- Section 4.8.1, "Use Case 1: Expose Web Services to Customers and Trading Partners"
- Section 4.8.2, "Use Case 2: Manage Intellectual Property in an Outsourced Environment"
- Section 4.8.3, "Use Case 3: Establish a Federated Repository"
- Section 4.8.4, "Use Case 4: Manage the Asset Lifecycle"
- Section 4.8.5, "Use Case 5: Limit Access to Source Code Files to Asset Production Teams"
- Section 4.8.6, "Use Case 6: Grant Browse-only Repository Access to Specific Groups"

4.8.1 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 4.8.1.1, "Benefit"
- Section 4.8.1.2, "Overview"
- Section 4.8.1.3, "Solution"

4.8.1.1 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.

4.8.1.2 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.

4.8.1.3 Solution

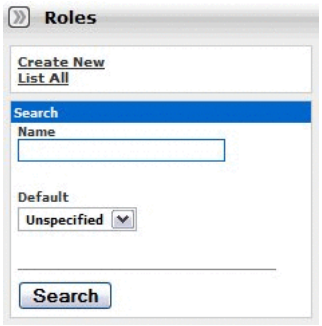
This section describes the solution to this use case.

4.8.1.3.1 Prerequisite Verify that Customer and Trading Partner users exist in Oracle Enterprise Repository.

4.8.1.3.2 Create and Assign the Roles This procedure is performed on the Oracle Enterprise Repository Admin screen.

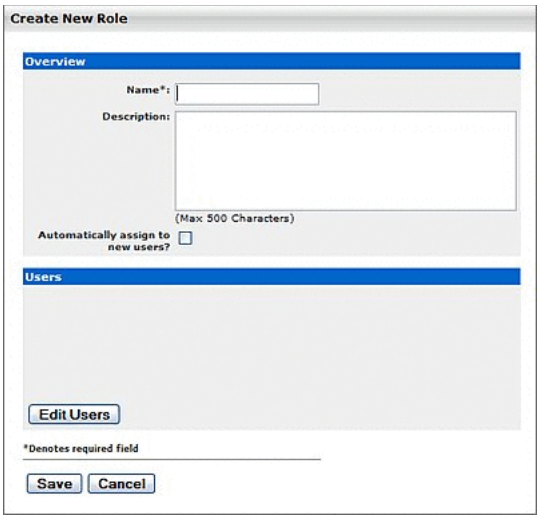
1. Click **Roles**. The Roles section is displayed, as shown in Figure 4–18.

Figure 4–18 Roles Section



2. Click **Create New**. The Create New Role dialog is displayed, as shown in Figure 4–19.

Figure 4–19 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.

4.8.1.3.3 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.

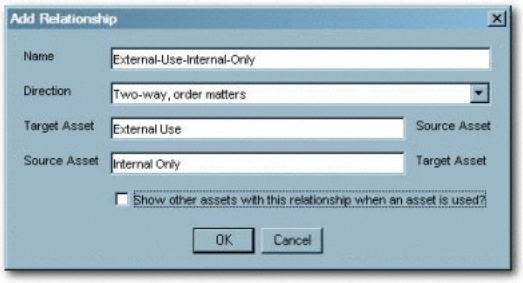
8. Delete all non-restricted fields (the first metadata grouping, as mentioned above) from the new Service - Internal Only asset type.

4.8.1.3.4 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 4–20.

Figure 4–20 Add Relationship Dialog



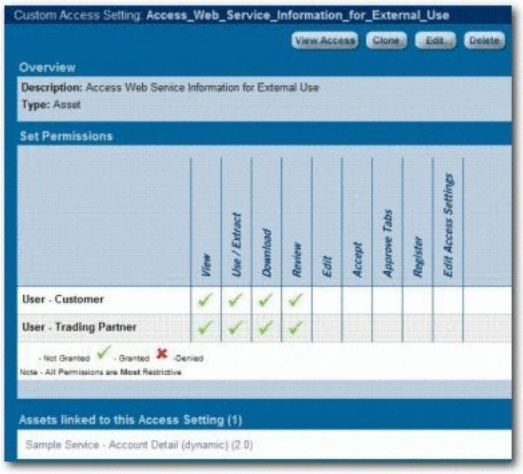
4.8.1.3.5 Apply the Relationship

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 4–21.

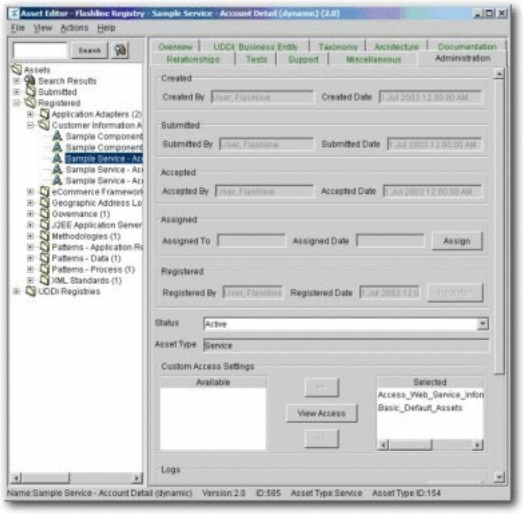
Figure 4–21 Custom Access Setting: CAS Access_Web_Service_Information_for_External_Use



- 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 4–22.

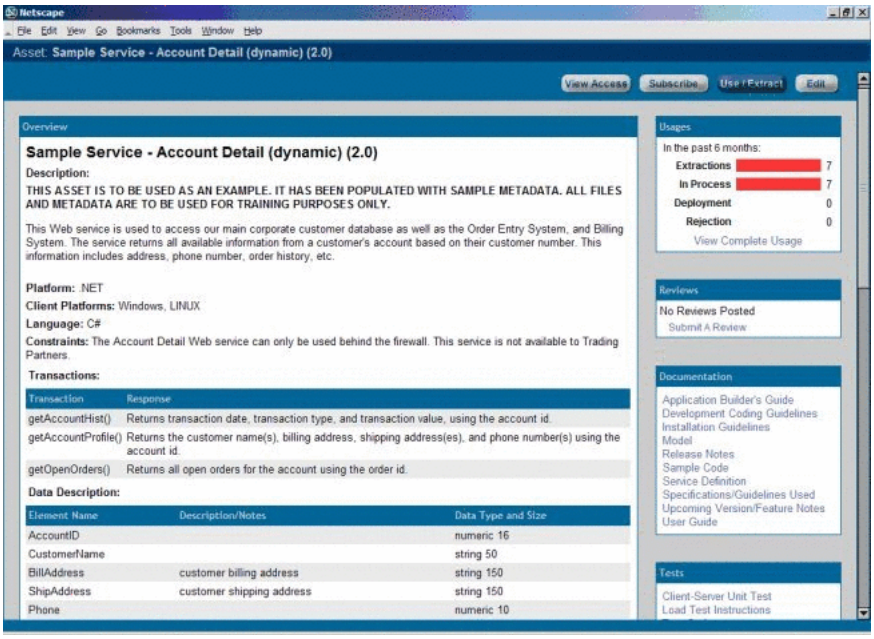
Figure 4–22 Asset Editor Screen



9. Repeat the process with each of the Web Service assets.

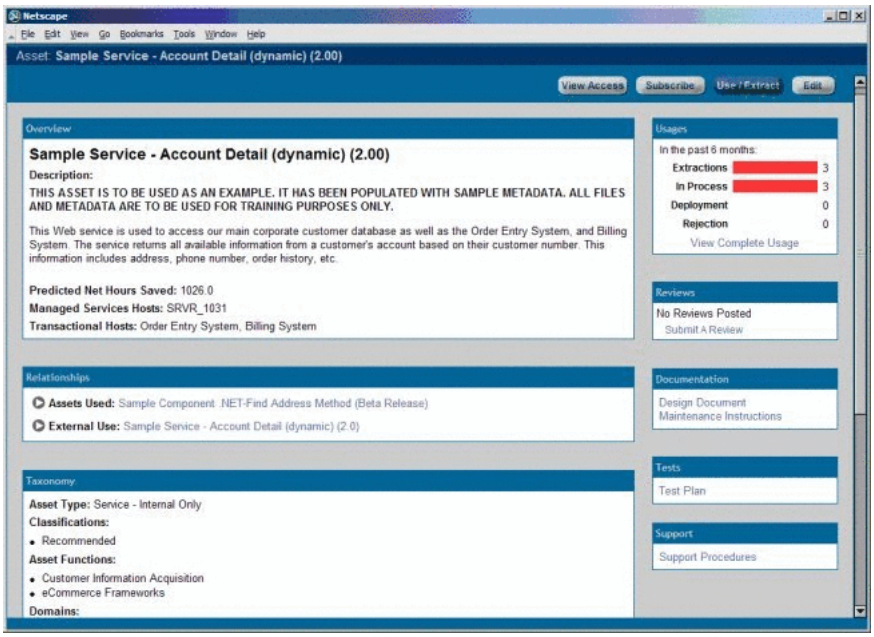
4.8.1.3.6 Results ■Service Asset Type

Figure 4–23 Service Asset Type



- Service - Internal Only Asset Type

Figure 4–24 Service - Internal Only Asset Type



Note: Note that considerably less information is provided in this view.

4.8.1.3.7 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.

4.8.2 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 4.8.2.1, "Benefits"
- Section 4.8.2.2, "Overview"
- Section 4.8.2.3, "Solution"

4.8.2.1 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.

4.8.2.2 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.

4.8.2.3 Solution

This section describes the solution to this use case. This section contains the following topics:

- Section 4.8.2.3.1, "Solution for Outsourced Development Teams"
- Section 4.8.2.3.2, "Solution for Export Controls"

4.8.2.3.1 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.

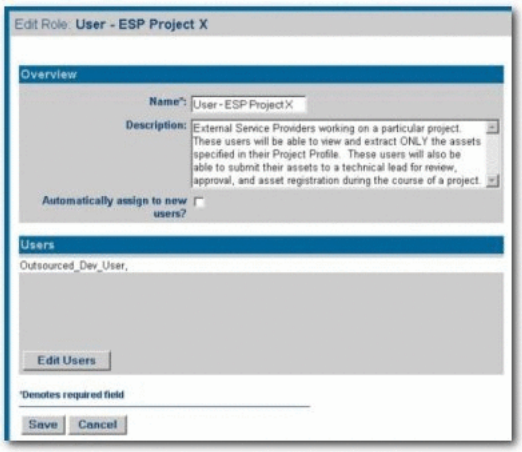
Assign the Roles

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.

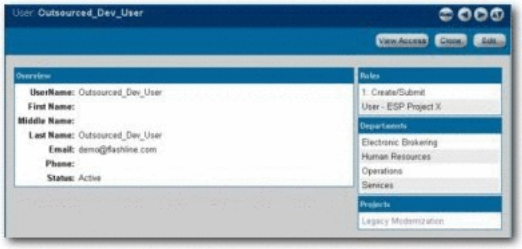
Click **Roles** to assign users to roles, as shown in Figure 4–25.

Figure 4–25 Edit Role Dialog



Click **Users** to assign roles to users, as shown in Figure 4–26.

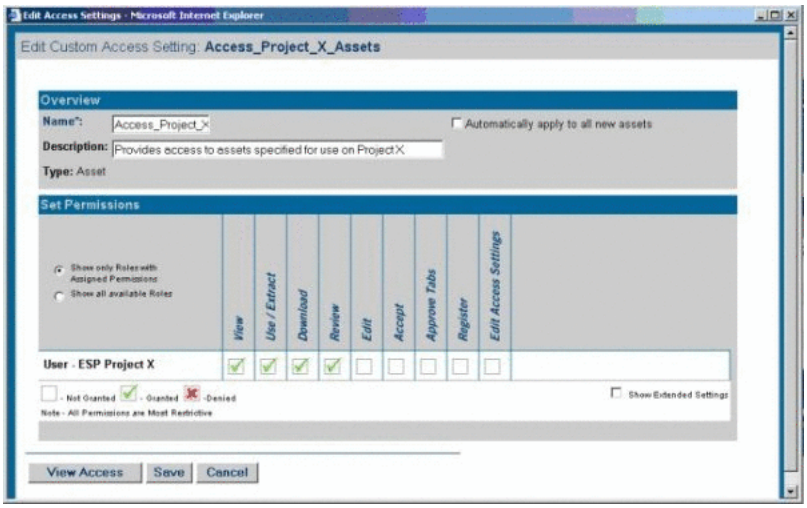
Figure 4–26 Users



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 4–27.
   - **Role**: User - ESP Project X
   - **Permissions**: View, Use, Download, Review

Figure 4–27 Edit Custom Access Settings Dialog



5. Identify the assets specified for use on Project X through the Project Profile, as shown in Figure 4–28.

Figure 4–28 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 4–29

Figure 4–29 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.

4.8.2.3.2 Solution for Export Controls The following procedures are performed on the Oracle Enterprise Repository Admin screen.

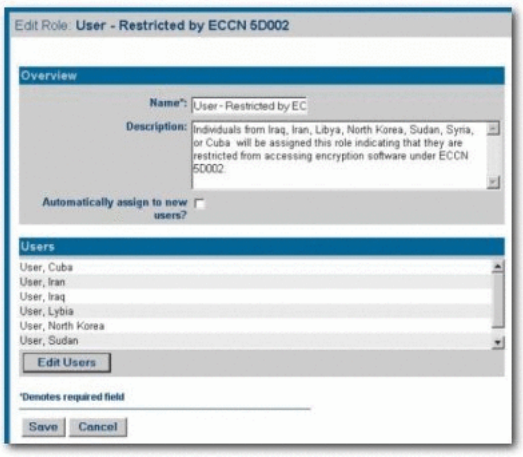
Prerequisites

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 4–30.

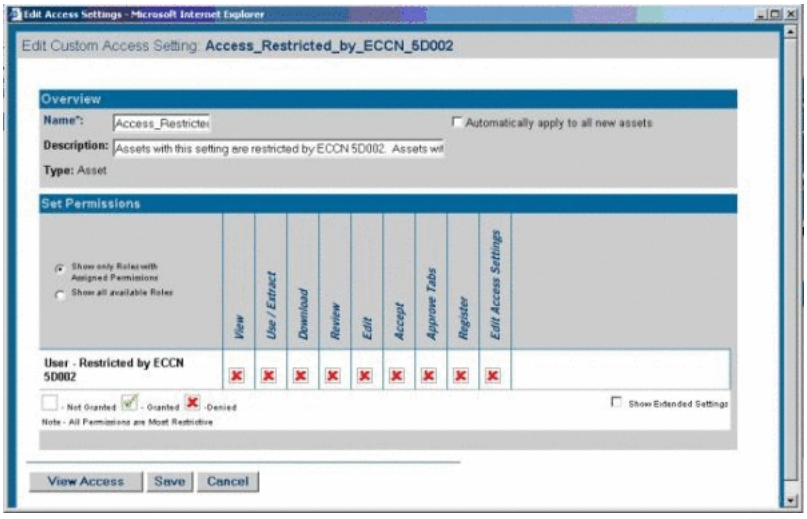
Figure 4–30 Edit Role



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 4–31.
   - **Role**: User - Restricted by ECCN 5D002
   - **Permissions**: DENY access to: View, Use, Download, Review, Accept, Approve Tabs, Register, Edit Access Settings

Figure 4–31 Edit Custom Access Setting Dialog



5. Add **Access_Restricted_by_ECCN_5D002 CAS** to all encryption software assets or assets restricted by export control under ECCN 5D002.

Figure 4–32 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.

4.8.3 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 4.8.3.1, "Benefit"
- Section 4.8.3.2, "Overview"
- Section 4.8.3.3, "Solution"

4.8.3.1 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.

4.8.3.2 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.

4.8.3.3 Solution

This section describes the solution to this use case.

4.8.3.3.1 Prerequisite Verify that enterprise and domain producers and all consumers have Oracle Enterprise Repository user accounts.

4.8.3.3.2 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.

4.8.3.3.3 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

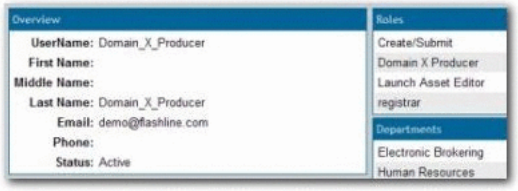
Figure 4–33 Overview Section

- Corresponding CAS

Figure 4–34 Set Permissions

- Role assigned to Domain_X_Producers

Figure 4–35 Overview Section



- Role assigned to Domain_X_Consumers

Figure 4–36 Overview Section

- Corresponding CAS

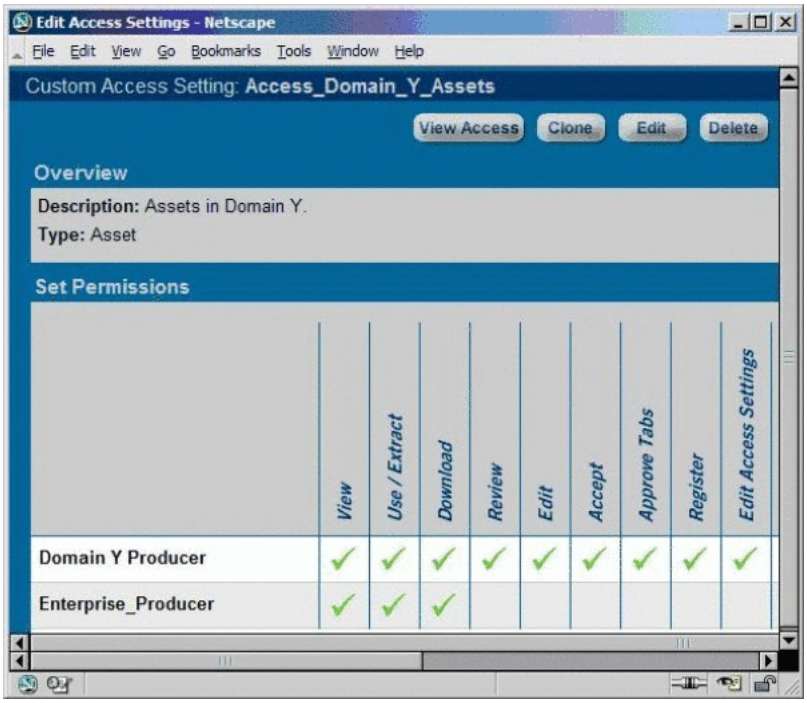
Figure 4–37 Set Permissions

- Roles assigned to Domain_Y_Producers

Figure 4–38 Overview Sections

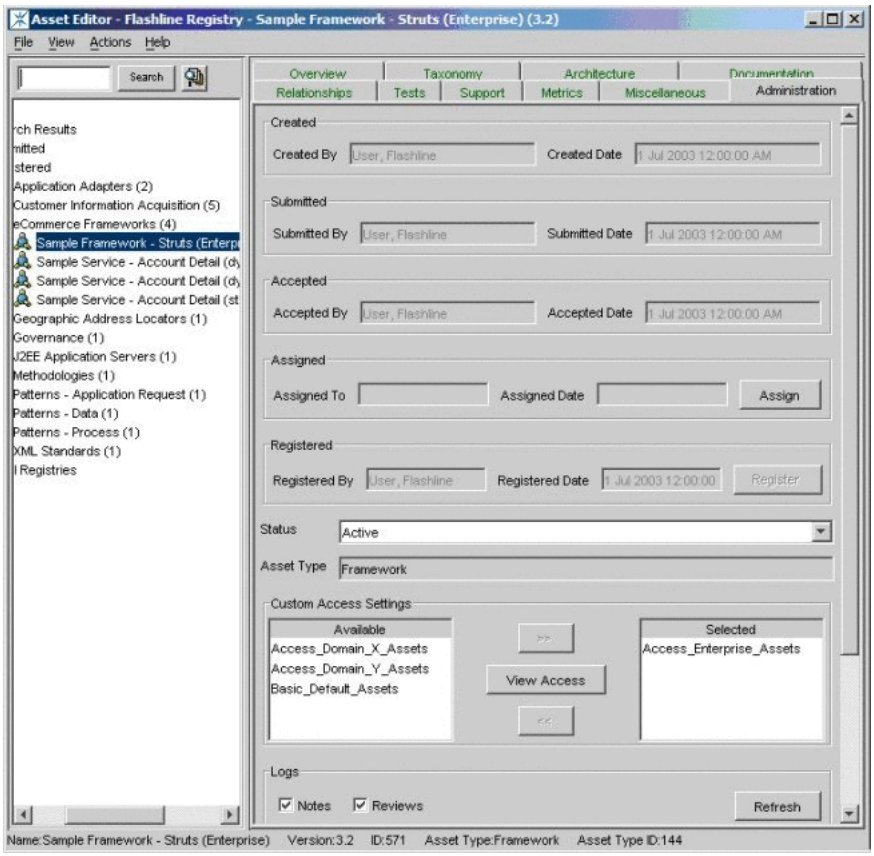
- Corresponding CAS

Figure 4–39 Edit Access Settings



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 4–40.

Figure 4–40 Asset Editor



4.8.3.3.4 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.

4.8.4 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 4.8.4.1, "Benefit"
- Section 4.8.4.2, "Overview"
- Section 4.8.4.3, "Configuring the Use Case Solutions"
- Section 4.8.4.4, "Asset Lifecycle Management Solution Validation Tests"

4.8.4.1 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.

4.8.4.2 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.

4.8.4.2.1 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.

4.8.4.2.2 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)

4.8.4.2.3 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

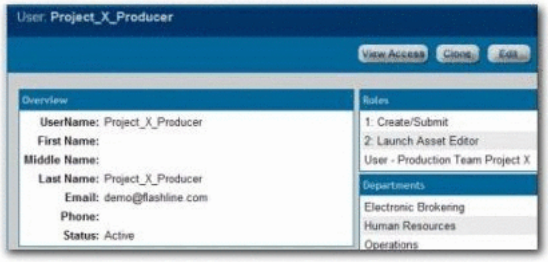
4.8.4.2.4 Create the Custom Access Settings

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.

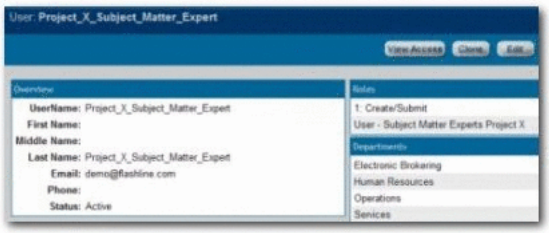
- Project_X_Producer: Roles

Figure 4–41 User: Project_X_Producer



- Project_X_Subject_Matter_Expert: Roles

Figure 4–42 User: Project_X_Subject_Matter_Expert

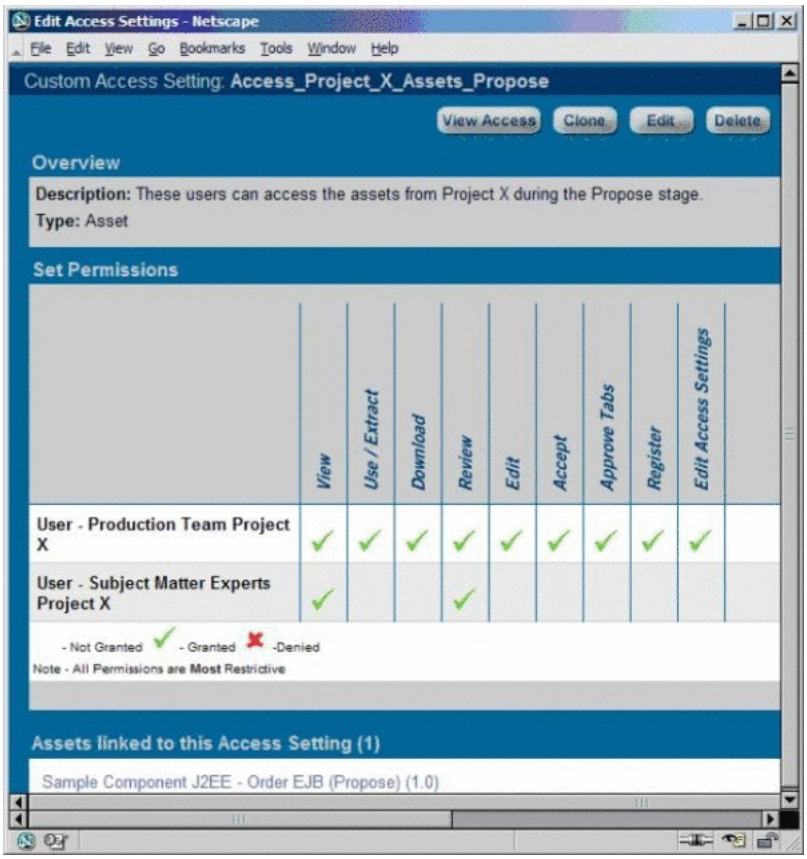


- User_Community: Roles

Figure 4–43 User: User_Community

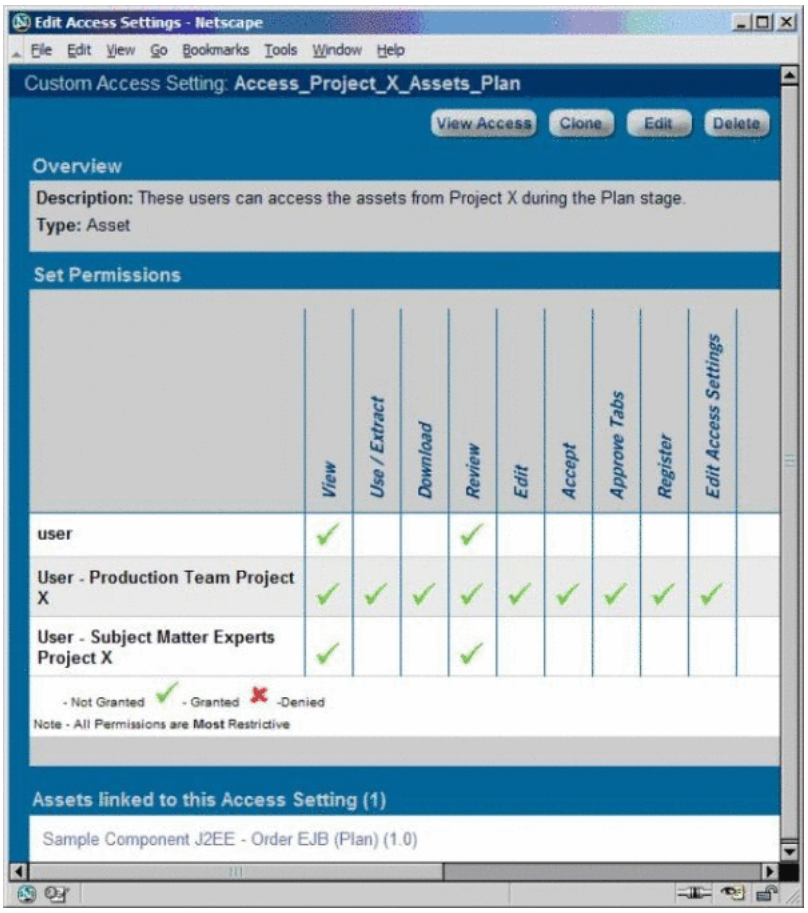
- Asset Lifecycle Stage - Propose
  - CAS: Access_Project_X_Assets_Propose

Figure 4–44 Edit Access Settings



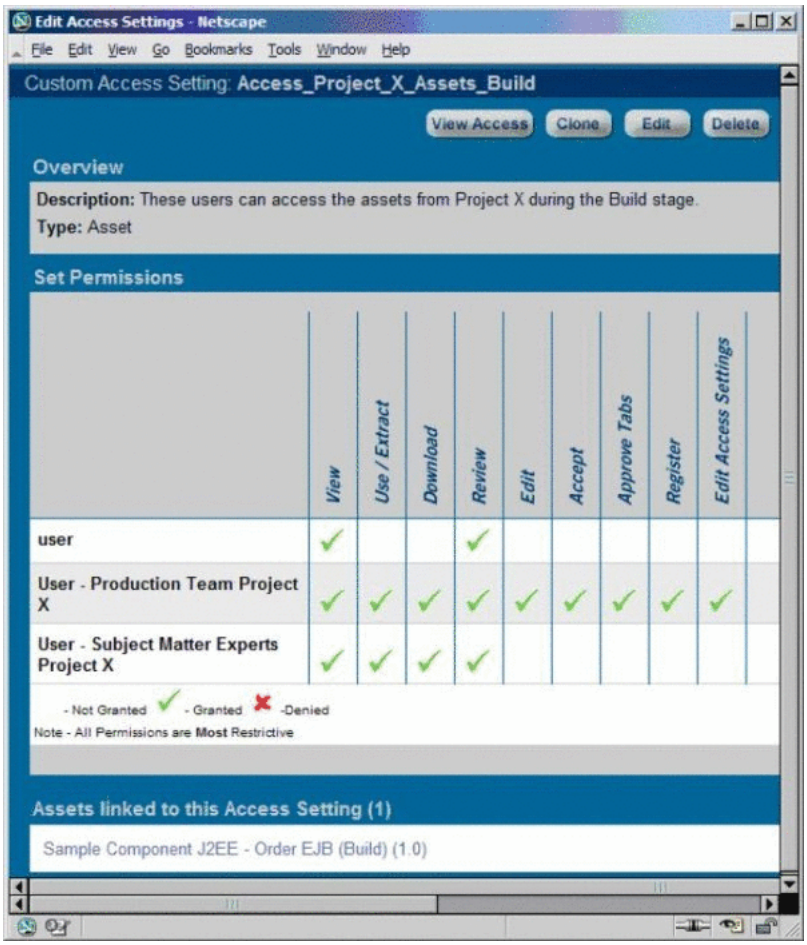
- Asset Lifecycle Stage - Plan
  - CAS: Access_Project_X_Assets_Plan

Figure 4–45 Edit Access Settings



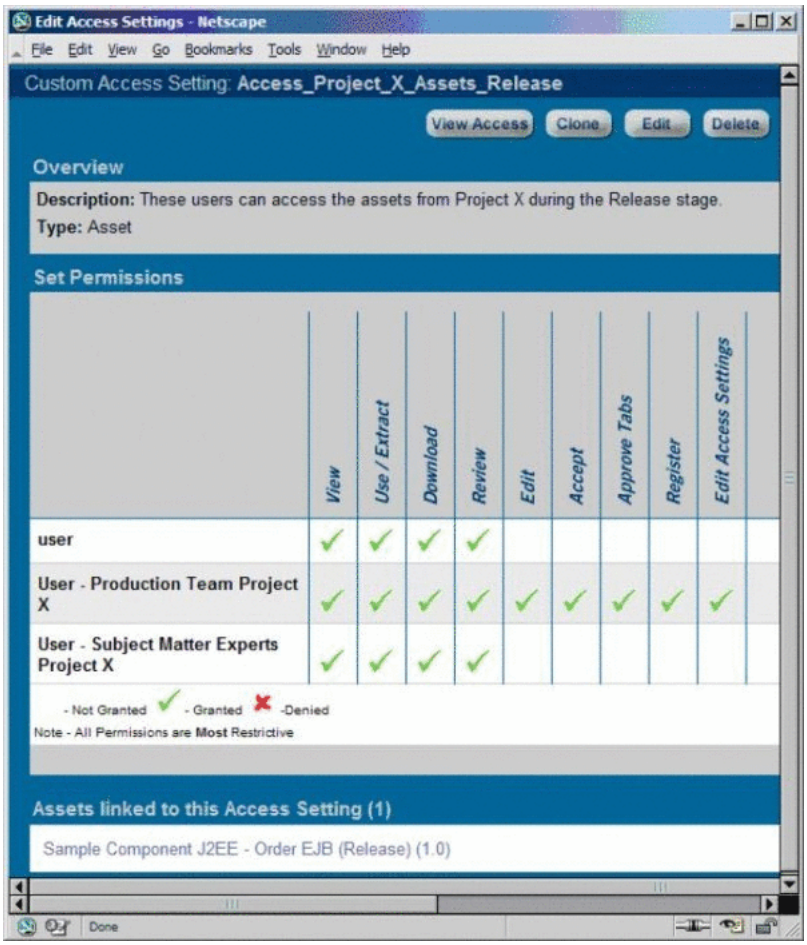
- Asset Lifecycle Stage - Build
  - CAS: Access_Project_X_Assets_Build

Figure 4–46 Edit Access Settings



- Asset Lifecycle Stage - Release
  - CAS: Access_Project_X_Assets_Release

Figure 4–47 Edit Access Settings



4.8.4.3 Configuring the Use Case Solutions

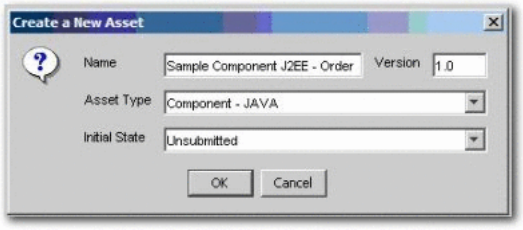
The following section outlines the necessary steps for each phase of the Asset Lifecycle Management use case.

4.8.4.3.1 Requirements Gathering Stage Solution

(Prerequisite: Verify that the Asset Lifecycle Categorization taxonomy is included as part of your selected asset type.)

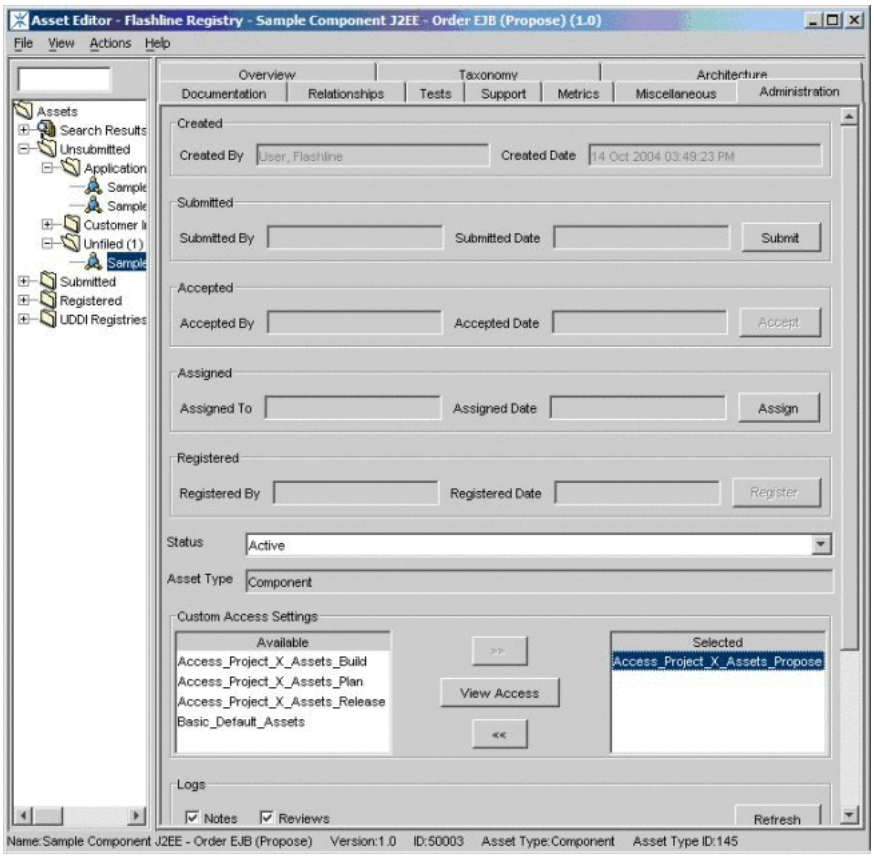
1. Click the **Assets** link in the Oracle Enterprise Repository menu bar.
2. Click **Edit/Manage Assets** to launch the Asset Editor.
3. Open the File menu in the Asset Editor.
4. Click **New**.
5. Create an asset with the initial state of Unsubmitted, as shown in Figure 4–48.

Figure 4–48 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 4–49.

Figure 4–49 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 4–50.

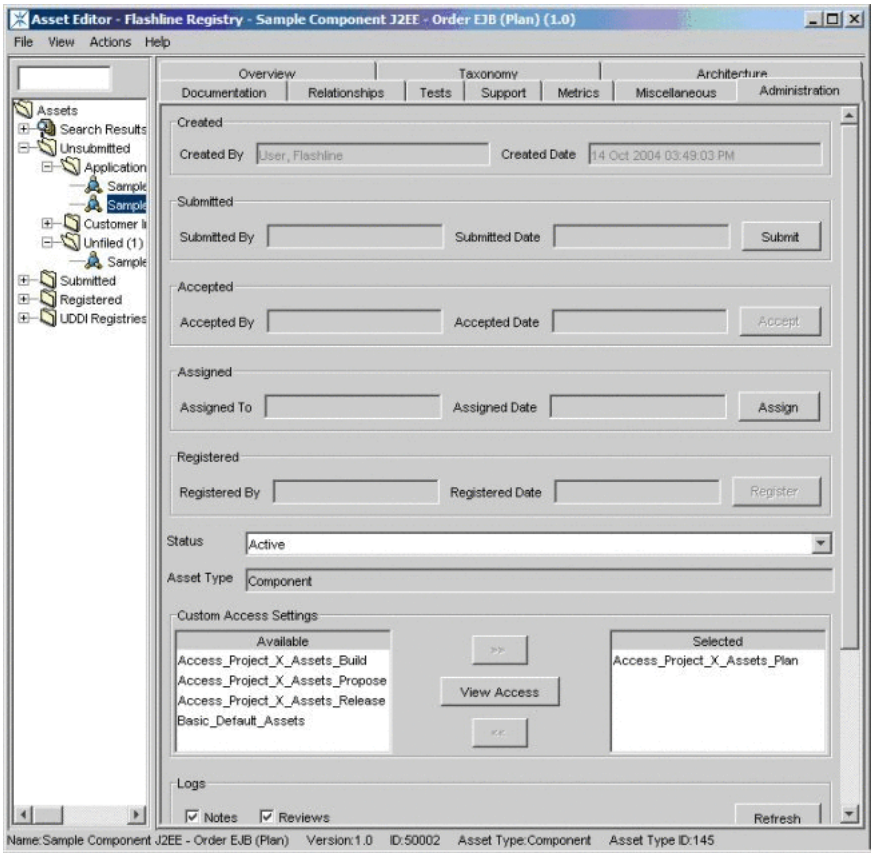
Figure 4–50 Browse Asset Tree

4.8.4.3.2 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 4–51 Asset Editor



4.8.4.3.3 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.

4.8.4.3.4 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.

4.8.4.3.5 Scheduled for Retirement Stage Solution 1.In the Asset Editor, change the status of the asset from Active to Inactive.

4.8.4.3.6 Retired Stage Solution 1.In the Asset Editor, change the status of the asset from Inactive to Retired.

4.8.4.4 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.

4.8.5 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 4.8.5.1, "Benefit"
- Section 4.8.5.2, "Overview"
- Section 4.8.5.3, "Solution"

4.8.5.1 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.

4.8.5.2 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.

4.8.5.3 Solution

This section describes the following use cases:

4.8.5.3.1 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

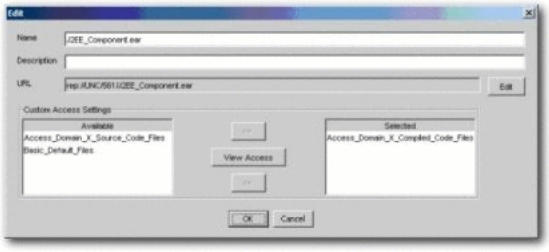
4.8.5.3.2 Create the Customer Access Settings

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.

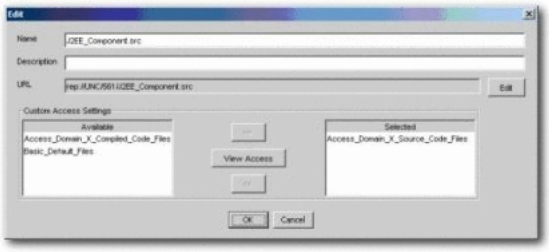
5. Remove the **Basic_Default_Files CAS** from the targeted assets. (This assumes that the Default File CAS has been set up.), as shown in Figure 4–52.

Figure 4–52 Edit Dialog



6. Assign the **Access_Domain_X_Compiled_Code_Files CAS** to all compiled code files, as shown in Figure 4–53.

Figure 4–53 Edit Dialog



7. Assign the **Access_Domain_X_Source_Code_Files** CAS to all source code files.

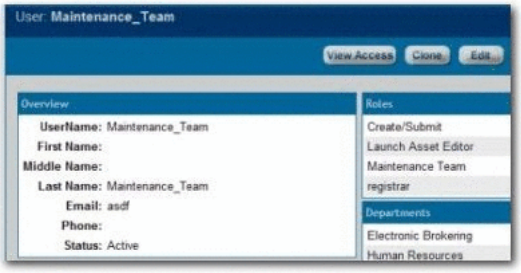
Figure 4–54, Figure 4–55, Figure 4–56, Figure 4–57, Figure 4–58, and Figure 4–59 illustrate the Production Team, Maintenance Team, and Consumer roles, and the relevant CAS settings respectively.

- Production_Team (all production users): Roles

Figure 4–54 User: Production_Team

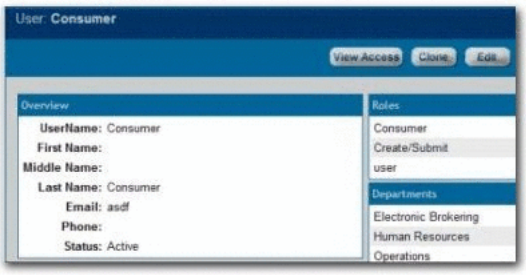
- Maintenance_Team (all maintenance users): Roles

Figure 4–55 User: Maintenance_Team



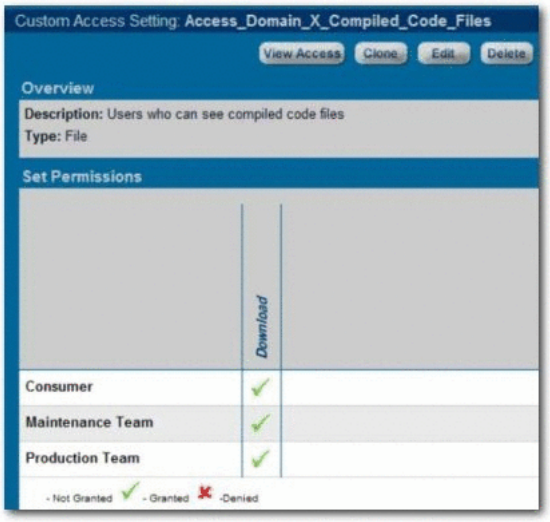
- Consumers (all consumers): Roles

Figure 4–56 User: Consumer



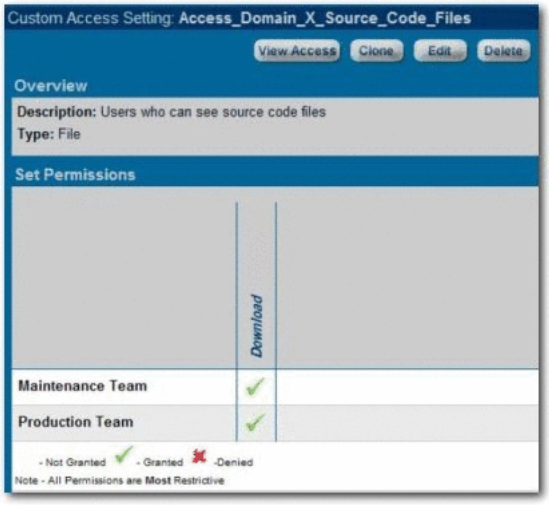
- Access_Domain_X_Compiled_Code_Files

Figure 4–57 Custom Access Setting



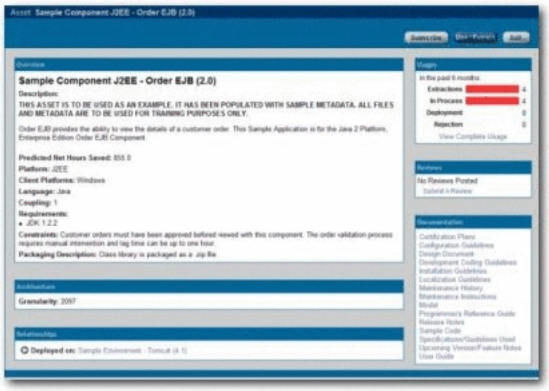
- Access_Domain_X_Source_Code_Files

Figure 4–58 Custom Access Setting



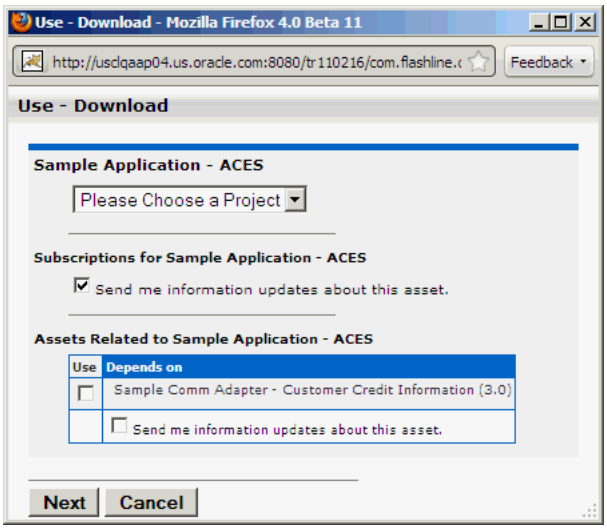
4.8.5.3.3 Results Production Team, Maintenance Team, and Consumer users see the same metadata information when viewing the asset in Oracle Enterprise Repository, as shown in Figure 4–59.

Figure 4–59 Asset: Sample Component



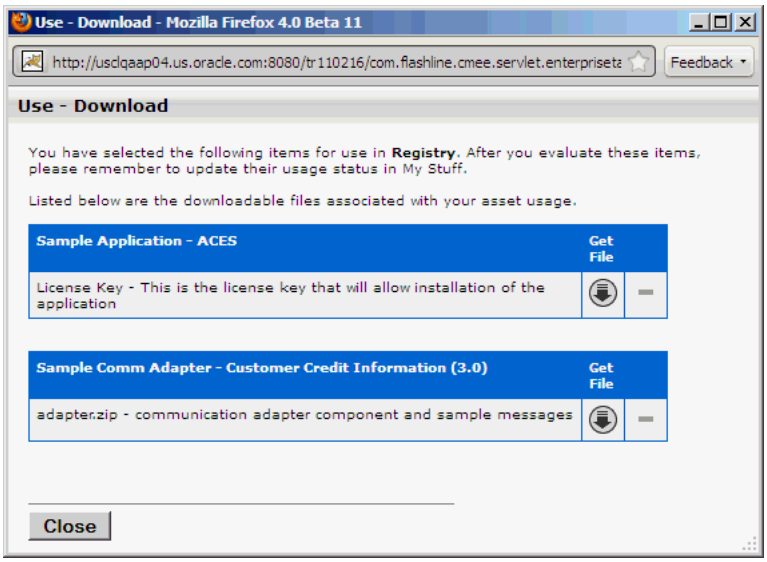
Production Team and Maintenance Team users see both the compiled and source code files during asset download, as shown in Figure 4–60.

Figure 4–60 Use / Extract Dialog



Consumers see only the complied code file during asset download, as shown in Figure 4–61.

Figure 4–61 Use / Extract Dialog



4.8.5.3.4 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.

4.8.6 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 4.8.6.1, "Benefit"
- Section 4.8.6.2, "Overview"
- Section 4.8.6.3, "Solution"
- Section 4.8.6.4, "Validation Test"

4.8.6.1 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.

4.8.6.2 Overview

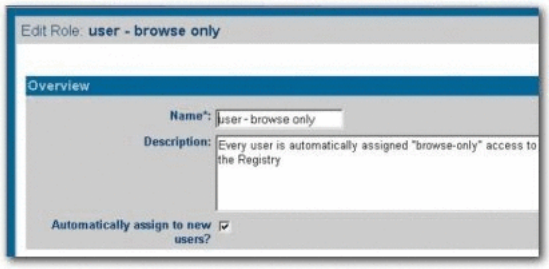
In this scenario, all Repository users are automatically assigned browse-only access. It is likely that users are authenticated against LDAP, and will gain access to the Repository using their standard organization-assigned login user name and password. As these users determine their need for additional functionality, they can be assigned additional roles by Repository administrators.

4.8.6.3 Solution

This section describes the solution for this use case.

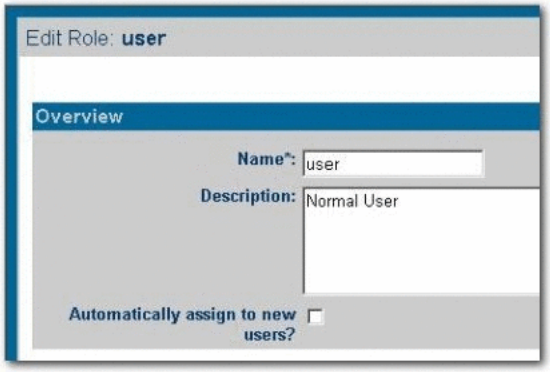
1. Create the following role:
   - user - browse only
2. Ensure that this role is automatically assigned to new users, as shown in Figure 4–62.

Figure 4–62 Edit Role: user-browse only



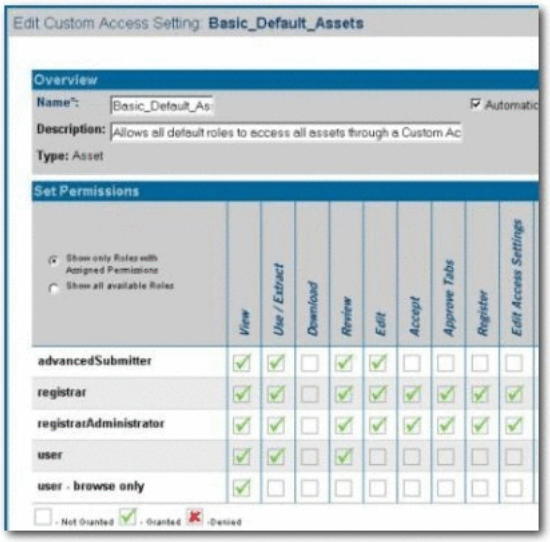
3. Edit the User role to ensure that it is not automatically assigned to new users, as shown in Figure 4–63.

Figure 4–63 Edit Role: user



4. Edit **CAS: Basic_Default_Assets**.

Figure 4–64 Edit Custom Access Setting: Basic_Default_Assets



- 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.

4.8.6.4 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

5

Diagnostics and Password Encryption

This chapter provides information about how you can access the Oracle Enterprise Repository Diagnostics page and encrypt passwords.

This chapter contains the following sections:

- Section 5.1, "Accessing Oracle Enterprise Repository Diagnostics Page"
- Section 5.2, "Generating Encrypted Passwords"
- Section 5.3, "Encrypting the Configuration File Passwords"

Password encryption is enabled by default within the Oracle Enterprise Repository, however, you may use the JVM startup parameter `cmee.passwordencryption=false` to disable password encryption.

5.1 Accessing Oracle Enterprise Repository Diagnostics Page

In Oracle Enterprise Repository 11*g* release, Oracle Enterprise Repository 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\base_domain\bin* to access the .sh and .cmd files.
2. Edit `setOERdomain.sh` or `setOERdomain.cmd` before starting the Oracle Enterprise Repository server.
3. Uncomment the following line that is commented, by default:

   ```
   OER_DIAG_OPTION="-DdiagPagesEnabled=true"
   ```

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.

5.2 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).

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 *<ORACLE_HOME>/repositoryXXX/core/tools/solutions/11.1.1.x.x-OER-PasswordTools.zip*. The **11.1.1.x.x-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 5.2.1, "Encrypted Passwords Used in Oracle Enterprise Repository Deployment"
- Section 5.2.2, "Other Passwords"

5.2.1 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.

5.2.2 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
- cmee.wsaa.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.

5.3 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 5.3.1, "Harvester Configuration File"
- Section 5.3.2, "Exchange Utility Configuration File"
- Section 5.3.3, "Workflow Configuration File"

5.3.1 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 distributed with Harvester 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.

5.3.2 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 5–1, 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 5–1 Encrypt Password Tool

Figure 5–2 describes a sample image of how the password field appears before the encryption.

Figure 5–2 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>
```

Figure 5–3 describes a sample image of how the password field appears after you run the password encryption tool.

Figure 5–3 Example Image of Password After Encryption

```
<!--Connection info to OER-->
<repository>
    <uri>http://localhost:8080/oer</uri>
    <credentials>
        <user>admin</user>
        <password>v2_1.qRhDTl1LdPo=</password><!--run encrypt.bat to encrypt this-->
    </credentials>
</repository>
```

5.3.3 Workflow Configuration File

For enhanced security, the password encryption tool (`encrypt.bat/encrypt.sh`), which resides in the `11.1.1.x.x-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/repositoryXXX/core/tools/solutions` directory.
2. Extract the `11.1.1.x.x-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

6

Configuring and Using Automated Harvesting in Design-time and Runtime Environments

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 can be harvested at design-time, directly from the development environment.

The files can also be harvested as they are deployed to the runtime environment. The Harvester can be used from the command line, within IDEs including Oracle JDeveloper, and within Ant and the Weblogic Scripting Tool (WLST). See Section 10.2, "Configuring the Oracle Registry Repository Exchange Utility," for more information.

This chapter contains the following sections:

- Section 6.1, "Getting Started with Harvester"
- Section 6.2, "Configuring the Harvester"
- Section 6.3, "Harvesting of Artifacts"

6.1 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 6.1.1, "Prerequisites"
- Section 6.1.2, "Harvester Functionality"
- Section 6.1.3, "Artifacts/Products Version Matrix"

6.1.1 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

`repositoryxxx/core/tools/solutions/11.1.1.x.x-OER-Harvester-Solution-Pack.zip` file from the Oracle Enterprise Repository installation home.

For more information about importing the Oracle Enterprise Repository solution packs, see *Oracle Fusion Middleware Installation Guide for Oracle Enterprise Repository*.

For more information about asset type inventory for the Harvester Solution Pack, see Section B.8, "Asset Type Inventory".

- If you wish to harvest from Oracle Service Bus, install Oracle Service Bus server (version 10.3.1.0) on the same machine as the harvester, or a shared file system accessible to the harvester.

  Note: If you wish to harvest any later versions of Oracle Service Bus 11*g*, then you have to use the corresponding Oracle Service Bus harvester version that is specifically developed for later versions of Oracle Service Bus 11*g* and is available as part of Oracle Service Bus 11*g* product installation.

- 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 10.3.1 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.

6.1.2 Harvester Functionality

You can use the Harvester to:

- Publish standard SOA artifacts such as SCA Composites, BPEL, WSDL, XSD, and XSLT, BPMN 2.0 and WS-Policy (from Oracle and other vendors) to Oracle Enterprise Repository.
- 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*

6.1.3 Artifacts/Products Version Matrix

Table 6–1 describes the version matrix of the artifacts/products that is supported with Harvester.

Table 6–1 Artifacts/Products Version Matrix

Artifact/Product	Version
BPEL	1.1 and 2.0
WSDL	1.1
SCA	0.95
JCA Adapter	1.5
XSD	1.0
XSLT	1.0
Oracle BPEL PM	10.1.3.4
JDeveloper	10.1.3 and 11.1.1
Oracle SOA Suite	10.1.3 and 11.1.1
WS-Policy	1.2
Oracle Service Bus	10.3 and 11.1.1
WebLogic Web Services	10.3
BPMN	2.0

6.2 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 6.2.1, "Obtaining the Harvester"
- Section 6.2.2, "Configuring the Harvester for the Command Line"
- Section 6.2.3, "Using Harvester within an Integrated Development Environment"
- Section 6.2.4, "Invoking the Harvester Using the Repository.Submit Ant Task"
- Section 6.2.5, "Invoking Harvester from WLST"
- Section 6.2.6, "Runtime Harvesting Details"

- Section 6.2.7, "Performing Optional Harvester Configuration"
- Section 6.2.8, "Configuring Logging for the Harvester"
- Section 6.2.9, "Transaction Handling in Harvester"
- Section 6.2.10, "Advanced Configuration"

6.2.1 Obtaining the Harvester

The Harvester is available in `11.1.1.x.x-OER-Harvester.zip`, which is bundled with the Oracle Enterprise Repository 11*g* installation, at the following location:

<ORACLE_HOME>/repositoryXXX/core/tools/solutions/11.1.1.x.x-OER-Harvester.zip

This manual refers to the directory that you extract the `11.1.1.x.x-OER-Harvester.zip` file to as the `<Harvester Home>` directory.

Table 6–2 describes the relationship matrix for the product and the corresponding version.

Table 6–2 Product and Version Matrix for Harvester

Product and Version	Harvest		IDE Search, Browse, and Consume
	Command Line	**IDE**	
11*g* OSB	OSB 11*g*	Yes	Yes
10.3.1 OSB	OER 11*g*	Yes	Add Oracle Enterprise Repository 11*g* Harvester to Eclipse IDE. For more information, see *Oracle Fusion Middleware Integration Guide for Oracle Enterprise Repository*
SOA Suite 11*g*	OER 11*g*	Yes	Yes
SOA Suite 10*g*	OER 11*g*	Yes	No
AIA Foundation Pack 11g	AIA	No	No
Standards based artifacts (WSDL 1.1, XSD, BPEL, XSLT, BPMN 2.0)	OER 11*g*	Varies	Varies

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 "The parameter.properties File" section in the *Oracle Fusion Middleware User's Guide for Oracle Enterprise Repository*.

6.2.2 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 6.2.2.1, "Setting Repository Connection Information for the Command Line"
- Section 6.2.2.2, "Selecting the Artifacts to Harvest for the Command Line"

6.2.2.1 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), which is located in
<ORACLE_HOME>/repositoryXXX/core/tools/solutions/11.1.1.x.x-OER-PasswordTools.zip,
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.

Alternatively, the repository connection information can also be passed as parameters to the command line utility as follows:

```
C:\test\harvester> harvest -url http://localhost:7101/oer -user admin
-password <password> -file c:\test\samples
```

6.2.2.2 Selecting the Artifacts to Harvest for the Command Line

The Harvester can be run from the command line using the `harvest.bat` utility.

Before running harvest.bat or harvest.sh, ensure that the environment variables mentioned in Table 6–3 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 6–3 Command Line Script

Environment Variable	Description
JAVA_HOME	Ensure that the JAVA_HOME environment variable points to an installed java runtime (JRE) or SDK. For Oracle Service Bus introspection, this must be Java version 6 or higher.
BEA_HOME	Ensure that the BEA_HOME 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, C:\bea.
JAVA_OPTS	Optionally, set your JAVA_OPTS 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 -Dhttp.proxyHost=www-proxy.yourhost.com -Dhttp.nonProxyHosts= "*.yourhost.com |localhost" See Also: *http://java.sun.com/javase/6/docs/technotes/guides/net/proxies.html*

Table 6–4 shows the options that can be specified using the Harvester command line utility:

Table 6–4 Command Line Options for the Harvester

Harvester Options	Description
-settings<file>	Specifies the configuration settings XML file.
-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.
-url <URL>	Specifies the URL of Oracle Enterprise Repository.
-user <OER User Name>	Specifies the user name of the Oracle Enterprise Repository user.
-password <OER 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: *http://<host>:<port>/<domain>/diag/encryptstrings.jsp* See Also: For more information about password encryption, see Chapter 5, "Diagnostics and Password Encryption".
-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 Section 6.2.4.2, "Specifying Parameters as Nested Elements".
-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.

Table 6–4 (Cont.) Command Line Options for the Harvester

Harvester Options	Description
`-remote_password <password>`	Specifies the password to connect to the remote server. To ensure security, the password must be encrypted. The Oracle Enterprise Repository Web console has a tool to encrypt passwords: *http://<host>:<port>/<domain>/diag/encryptstrings.jsp*
`-remote_server_type <type>`	Specifies the type of remote server. Remote server could include either of the following: SOASuite, OSB, WLS.
`-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.
`-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.
`-soa_partition <type>`	Specifies the name of partition for Oracle SOA Suite and by default, it uses the soa "default" partition.
`-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.
`-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_password ****** -remote_server_type SOASuite -remote_project
MyComposite_rev1.0 -soa_partition department1
```

To ensure security, the password must be encrypted. The Oracle Enterprise Repository Web console has a tool to encrypt passwords:

http://<host>:<port>/<domain>/diag/encryptstrings.jsp

Figure 6–1 shows the command line utility options and online help displayed by the `harvest.bat -help` command.

Figure 6–1 Harvest Command Line Utility Options

```
harvest (-settings |-url -user -password | (other params))
-settings (file)                  : Configuration Settings XML file
-url (URL)                        : OER URL
-user (OER User Name)             : OER User Name
-password (OER Password)          : OER Password
-artifact_store (store)           : 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)                 : Running server from which to harvest the
                                    remote project, instead of from a file.
-remote_username (username)       : Username to connect to the remote server
-remote_password (password)       : Password to connect to the remote server
-remote_server_type (type)        : Type of remote server.
                                    One of: (SOASuite, OSB, WLS)
-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.
-mds_settings (file)              : 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)       : 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)      : Will apply all assets created from the harvest to this producing
                                     The project must already exist in OER.
-help                             : Harvester Help.
```

None of the command line options are required. They can be omitted.

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>
```

6.2.3 Using Harvester within an Integrated Development Environment

Harvester can be integrated with integrated development environments (IDEs) such as Oracle JDeveloper, Eclipse, and Visual Studio .NET.

Note: 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.

For more information about integrating the harvester with the IDEs, see *Oracle Fusion Middleware Integration Guide for Oracle Enterprise Repository*.

6.2.4 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 `11.1.1.x.x-OER-Harvester.zip`).

This section contains the following topics:

- Section 6.2.4.1, "Specifying Parameters for the repository.submit Ant Task"
- Section 6.2.4.2, "Specifying Parameters as Nested Elements"
- Section 6.2.4.3, "Running the Harvester from Ant"
- Section 6.2.4.4, "Using the Third-Party Tasks"

6.2.4.1 Specifying Parameters for the repository.submit Ant Task

Table 6–5 shows parameters that can be specified for the `repository.submit` Ant task in the `harvest-tasks.xml` file:

Table 6–5 Parameters for the repository.submit Ant Task

Attribute	Description	Required
`repositoryURL`	Repository instance to connect to.	Yes, unless specified by a property.
`repositoryUsername`	Username to log into Oracle Enterprise Repository.	Yes, unless specified by a property.
`repositoryPassword`	Password to log into Oracle Enterprise Repository. To ensure security, the password must be encrypted. The Oracle Enterprise Repository Web console has a tool to encrypt passwords: *http://<host>:<port>/<domain>/diag/encryptstrings.jsp*	Yes, unless specified by a property.
`timeout`	Number of seconds before calls to Oracle Enterprise Repository will time out.	No. Defaults to 300 (5 minutes).
`failOnError`	Fails the entire build script if the Oracle Enterprise Repository operation results in an error.	No. Defaults to "true."
`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.

Table 6–5 (Cont.) Parameters for the repository.submit Ant Task

Attribute	Description	Required
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 Section B.10, "Harvester Properties".	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.

6.2.4.2 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 6–1, 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 6–1 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. Acceptable values are:

- **Unsubmitted**
- **Submitted - Pending Review**
- **Submitted - Under Review**
- **Registered**

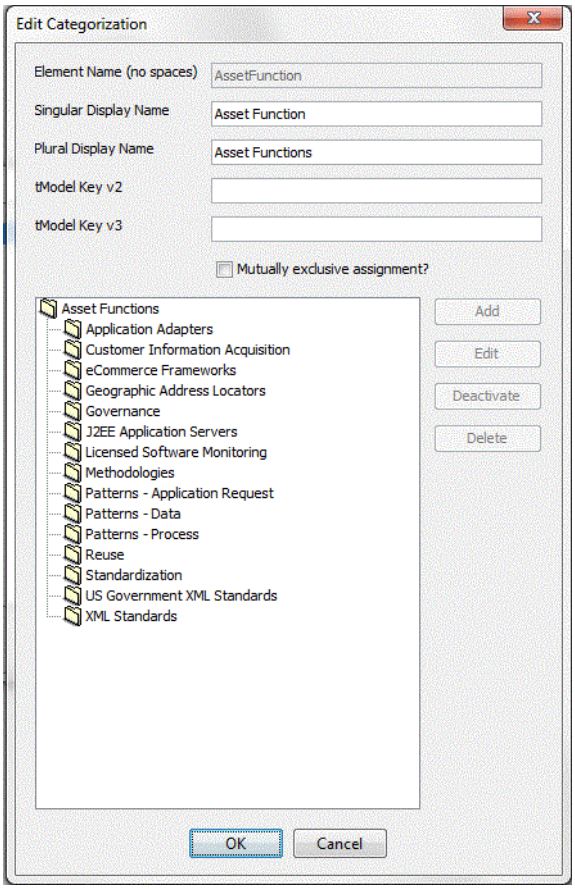
By default, a registration status is applied to newly created assets. You can also specify the registration status for assets that are created from a previous harvest of the same artifact. To enable, set `applyRegistrationStatusToExistingAssets` to **true**, as shown in Example 6–1.

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 6–2.

Figure 6–2 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`.

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 6–2 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.

6.2.4.3 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 `11.1.1.x.x-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 6–6 for the list of environment variables that you need to set.

Table 6–6 Command Line Script

Environment Variable	Description
JAVA_HOME	Ensure that the JAVA_HOME 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 ANT_HOME 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 BEA_HOME environment variable points to the installation directory containing Oracle Service Bus server, if you plan to harvest projects from Oracle Service Bus. For example, C:\bea.
JAVA_OPTS	Optionally, set your JAVA_OPTS 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 -Dhttp.proxyHost=www-proxy.yourhost.com -Dhttp.nonProxyHosts= "*.yourhost.com |localhost" See Also: *http://java.sun.com/javase/6/docs/technotes/guides/net/proxies.html*

6.2.4.4 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 URIStreamHandlers 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"/>
```

6.2.5 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 10*g* and SOA Suite 11*g*.

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_10.3\common\wlst\lib* directory.
2. In the command window, enter the following commands:
   - Initialize the WLST classpath: run *<BEA_HOME>\wlserver_10.3\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 6.2.2.2, "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://mywlsserver: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.

6.2.6 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 6.2.6.1, "Harvesting from Oracle SOA Suite Server"
- Section 6.2.6.2, "Harvesting Web Services from WebLogic Server"
- Section 6.2.6.3, "Harvesting Web Services from Oracle Service Bus"

6.2.6.1 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 server is as follows:

```
harvest.bat -remote_url mysoasuiteserver:8001 -remote_username weblogic
-remote_password <*****> -remote_server_type SOASuite -remote_project
MyComposite_rev1.0 -soa_partition default
```

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. For more information about password encryption, see Chapter 5, "Diagnostics and 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:** Partitions should be used only with SOA servers of version 11*g* 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
    WLLogFile /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.

6.2.6.2 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.

6.2.6.3 Harvesting Web Services from Oracle Service Bus

You need the Admin WLS security role when harvesting Web Services from Oracle Service Bus.

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:

```
harvest.bat -remote_url myosbserver:7001 -remote_username weblogic
-remote_password (******) -remote_server_type OSB -remote_project
MyOSBProject
```

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. For more information about password encryption, see Chapter 5, "Diagnostics and Password Encryption".

6.2.7 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 B.10, "Harvester Properties".

6.2.7.1 Oracle Metadata Service URLs

Harvester supports reading files from Oracle's MDS (Metadata Service). MDS is used in SOA Suite 11*g* 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 11*g* 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*.

6.2.7.2 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`.

6.2.8 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.

6.2.9 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:

- In 11*g* R1, the Oracle Enterprise Repository server 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.

6.2.10 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 `11.1.1.x.x-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 B.10, "Harvester Properties".

6.3 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 6.3.1, "Harvesting Artifacts"
- Section 6.3.2, "Harvesting Oracle Business Process Analysis Projects"
- Section 6.3.3, "Harvesting Oracle Business Process Management 11g Projects"
- Section 6.3.5, "Harvesting Oracle Service Bus 10g Assets"
- Section 6.3.6, "Harvesting a SOA Project from JDeveloper"
- Section 6.3.7, "Harvesting Deployed Composites from JDeveloper"
- Section 6.3.10, "Detecting Duplicate Artifacts"
- Section 6.3.11, "Downloading Harvested Artifacts"
- Section 6.3.12, "Searching Harvested Assets"
- Section 6.3.13, "Previewing of the Created Assets in Harvester"
- Section 6.3.14, "Best Practices"
- Section 6.3.15, "Known Issues"

6.3.1 Harvesting Artifacts

The metadata can be submitted to Oracle Enterprise Repository either from the command line, from Oracle JDeveloper, 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 6.3.1.1, "BPEL"
- Section 6.3.1.2, "WSDL"
- Section 6.3.1.3, "XSD"
- Section 6.3.1.4, "XSL"
- Section 6.3.1.5, "Harvesting Oracle SOA Suite"
- Section 6.3.1.6, "JCA Adapters"
- Section 6.3.1.7, "Applications via Oracle BSE (WSDL/SOAP)"
- Section 6.3.1.8, "Proxy Service or Business Service"
- Section 6.3.1.9, "Oracle Service Bus (OSB) Project"

- Section 6.3.1.10, "BPMN"

The Harvester creates entities for these artifacts in Oracle Enterprise Repository and creates the relationships between them.

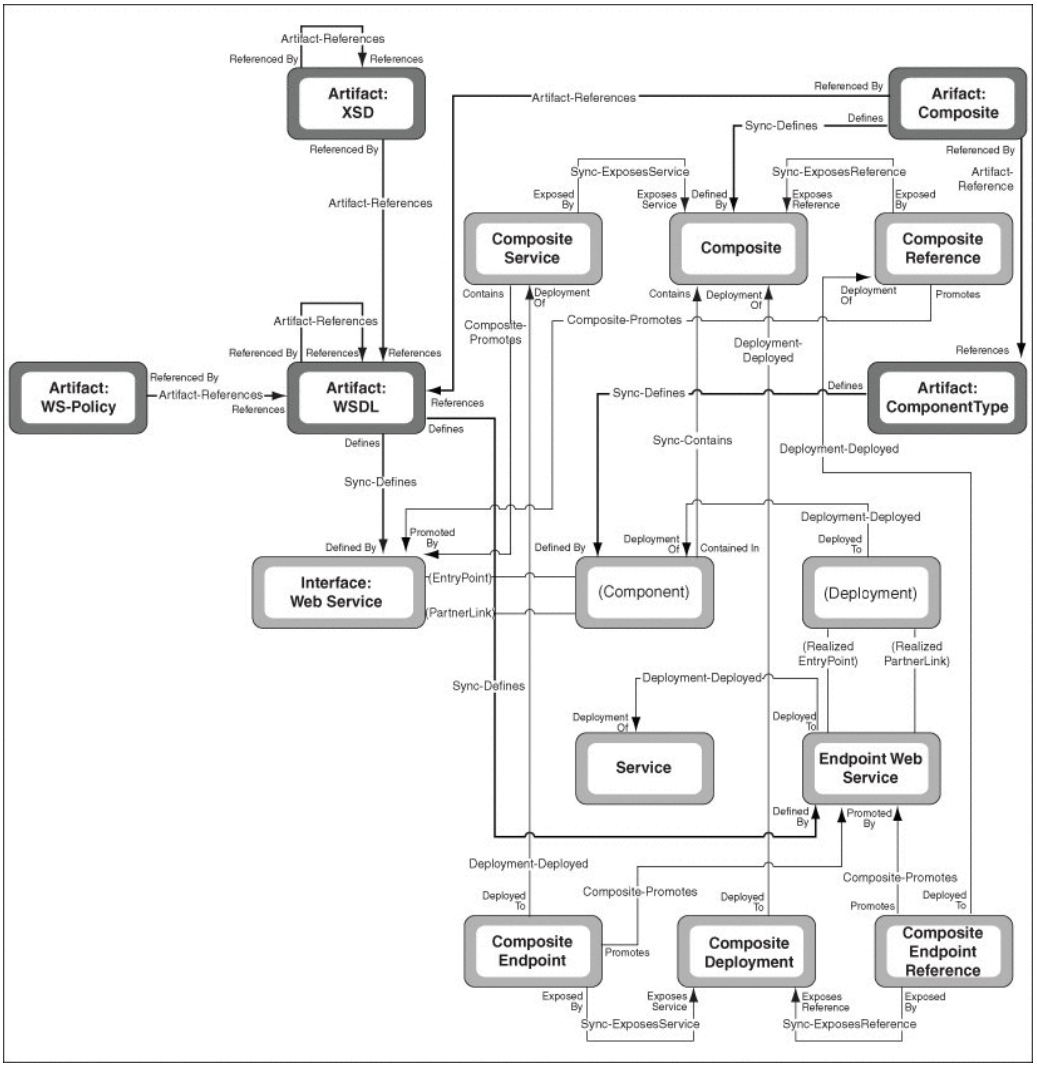
The artifacts created in Oracle Enterprise Repository are stored in the created asset's File Information. The locally stored harvested artifacts store the physical file in Oracle Enterprise Repository whereas for the remotely stored harvested artifacts (http://), Oracle Enterprise Repository points to 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 6–3 shows the asset types created by the Harvester and the relationships between them and Figure 6–4 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 6–3 Asset Types Created by the Harvester

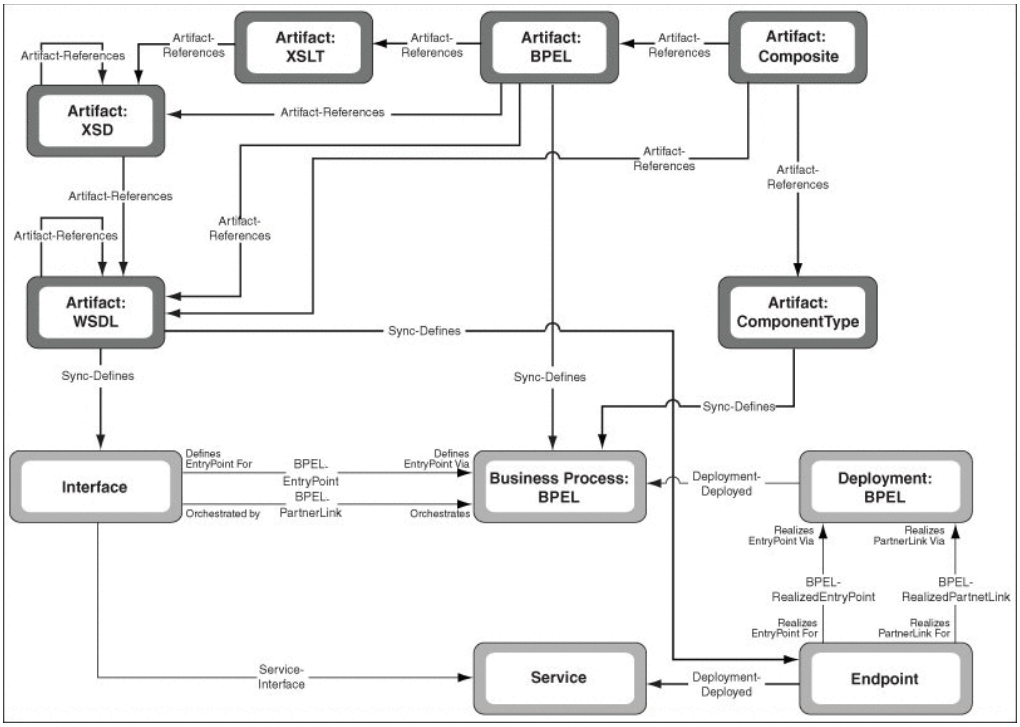


6.3.1.1 BPEL

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 6–4 Asset Model



6.3.1.2 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.

6.3.1.3 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.

6.3.1.4 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.

6.3.1.5 Harvesting Oracle SOA Suite

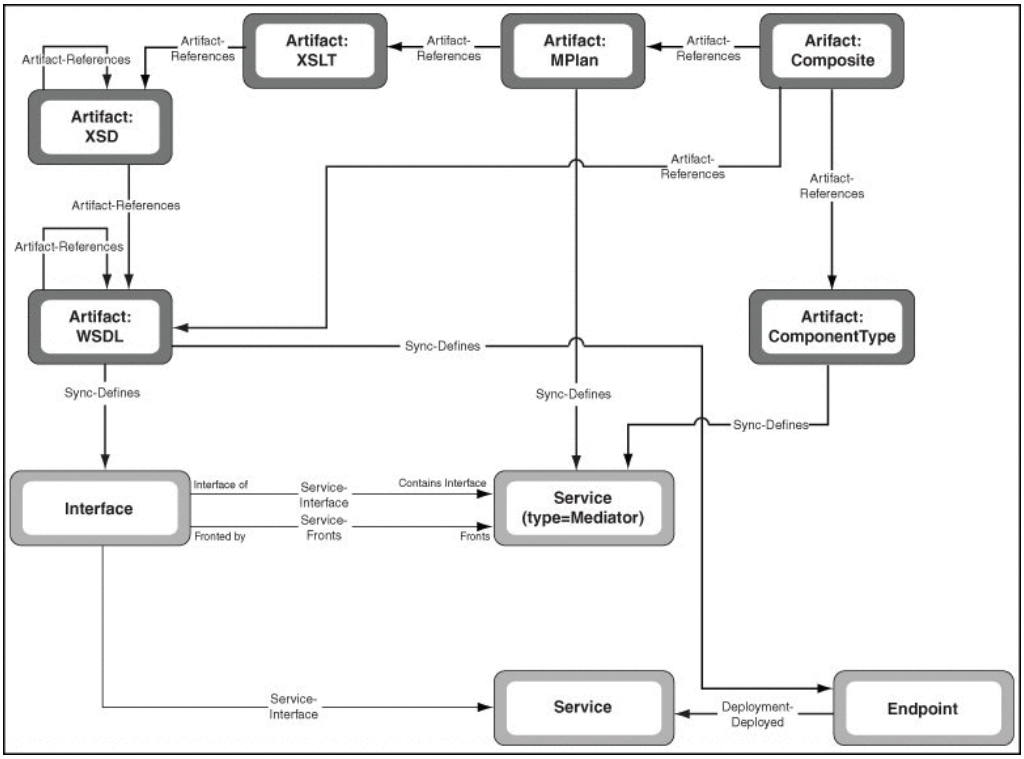
The section describes harvesting an Oracle SOA Suite 11*g* project.

- Composite: When an Oracle SOA Suite 11*g* 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 11*g* 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

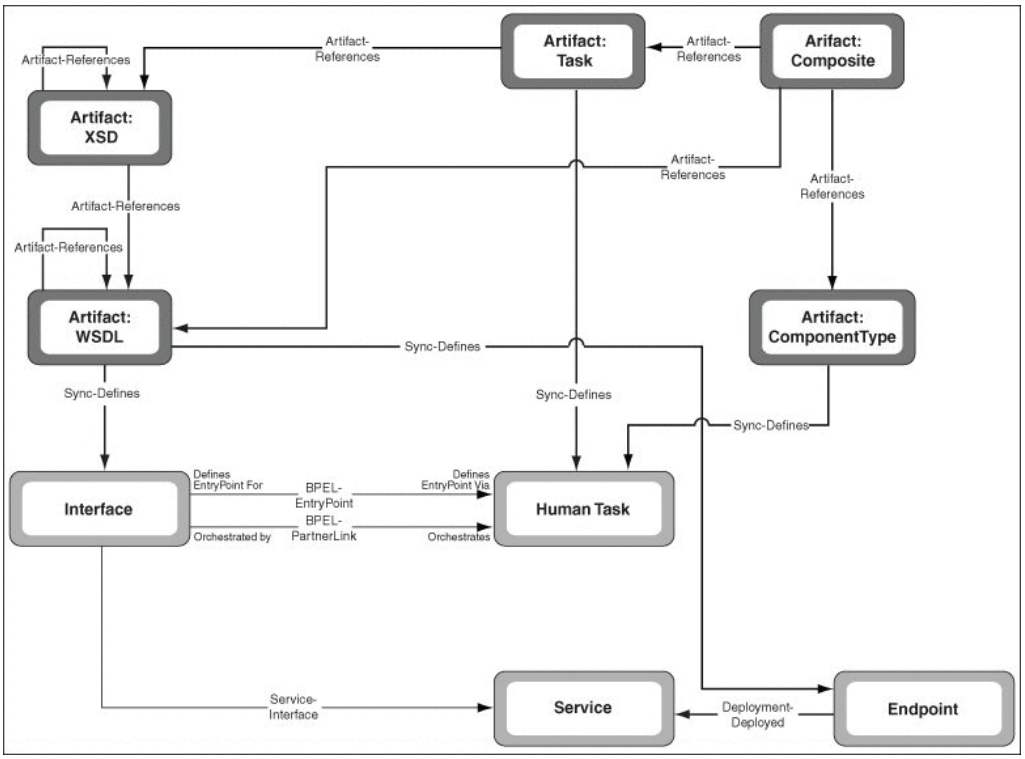
related to the Composite Deployment by the Sync-ExposesReference relationship.

  - The Composite Deployment are related to the Composite by the Deployment-Deploys relationship.

- Component-level Artifacts: The following components of the composite should be created and linked to the Composite by the Sync-Contains relationship:
  - Any BPEL process results in a `Business Process:BPEL` artifact.
    * It relates to an `Artifact:BPEL` artifact by the Sync-Defines relationship. This relates to any dependent artifacts, as described in Section 6.3.1.1, "BPEL".
  - Any Mediator should result in a Service asset, with the Service Type attribute set to `MEDIATOR`.
    * It relates to an `Artifact:MPlan` artifact by the Sync-Defines relationship. This relates to any dependent `Artifact:XSLT` artifacts by the Artifact-References relationship.

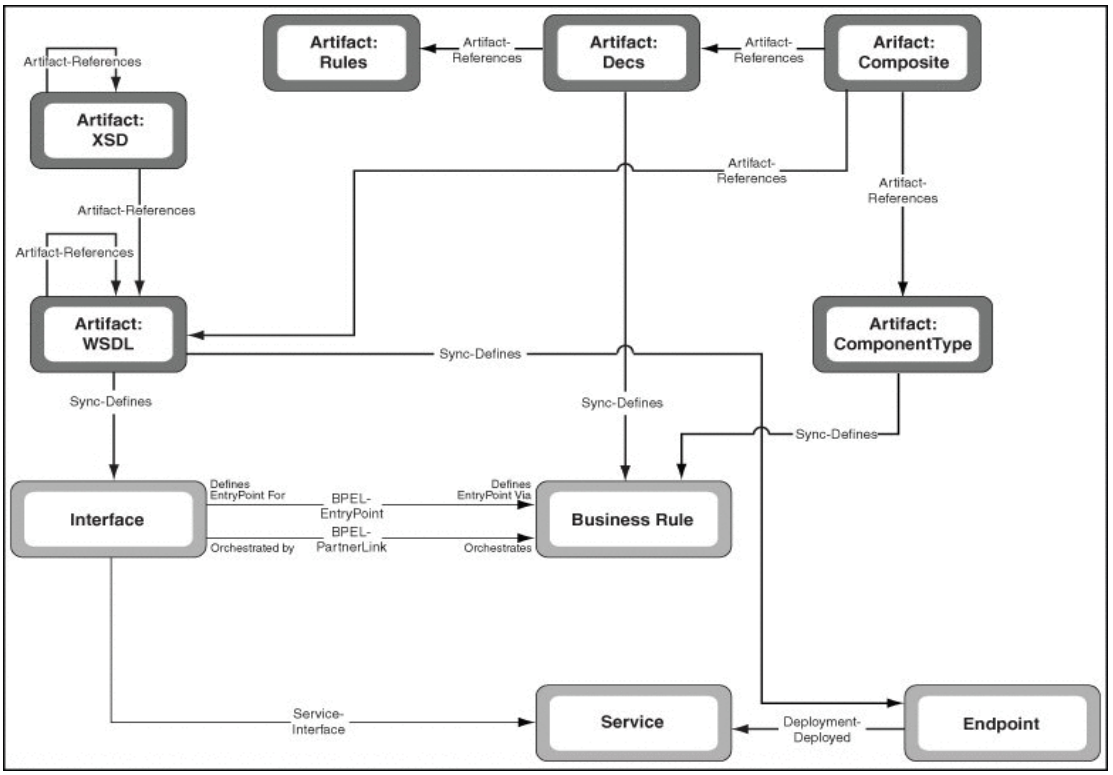


  - Any Human Task should result in a Human Task asset.
    * It relates to an `Artifact:Task` artifact by the Sync-Defines relationship. This relates to any dependent `Artifact:XSD` artifacts by the Artifact-References relationship.



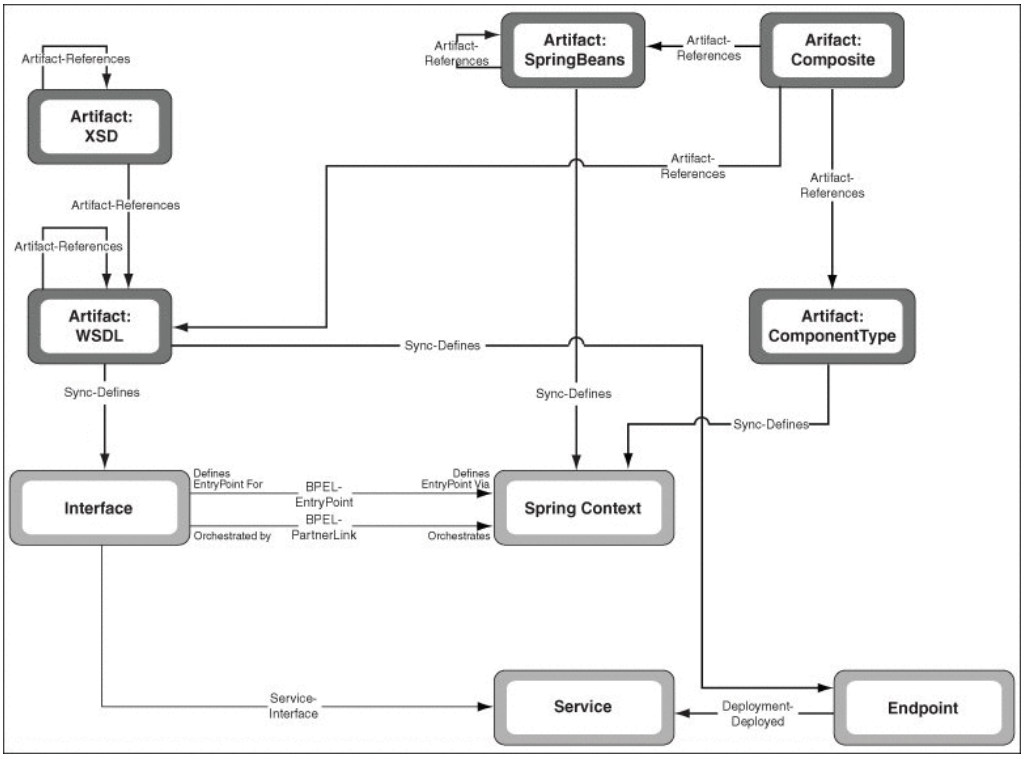
– Any Business Rule should result in a Business Rule asset.

  * 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.



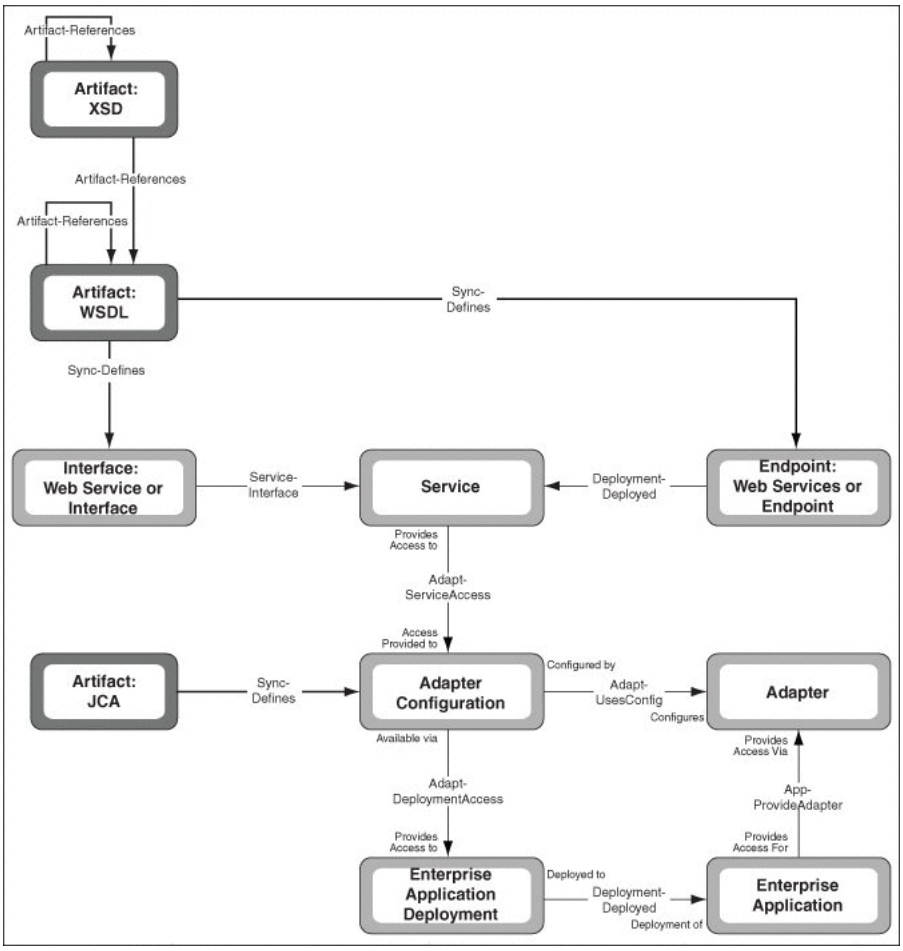
- Component-level Relationships:
  - Any component services offered by a component results in an Interface asset. This is related to the component-level asset by the BPEL-Entry point relationship (except for Mediators, in which case it is related by the Service-Interface relationship).
    * In the case 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.

6.3.1.6 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:

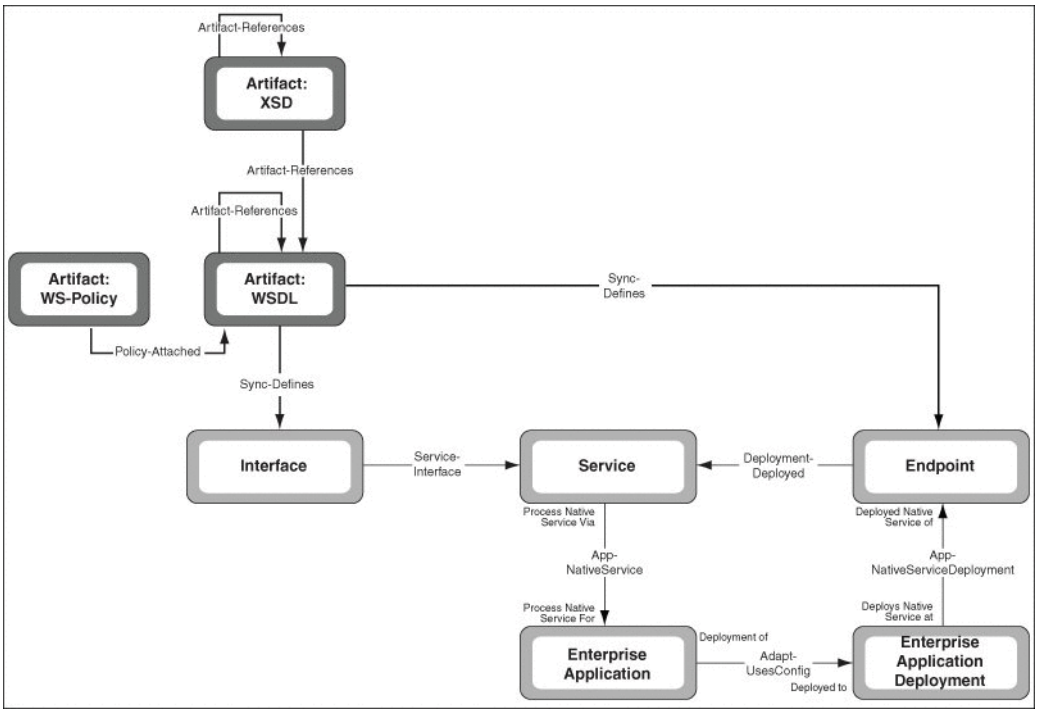
- Harvester attempts to locate an Adapter asset in Oracle Enterprise Repository, as configured in the Harvester's `adapters.xml` file.
- An Adapter Configuration asset is created that contains the binding and connectivity information used by a specific service.
  - It is related to the Adapter asset.
  - 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.



6.3.1.7 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.



6.3.1.8 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 B.10, "Harvester Properties".
  - The Service asset is related to the Endpoint asset using the relationship "`Deployed to`".

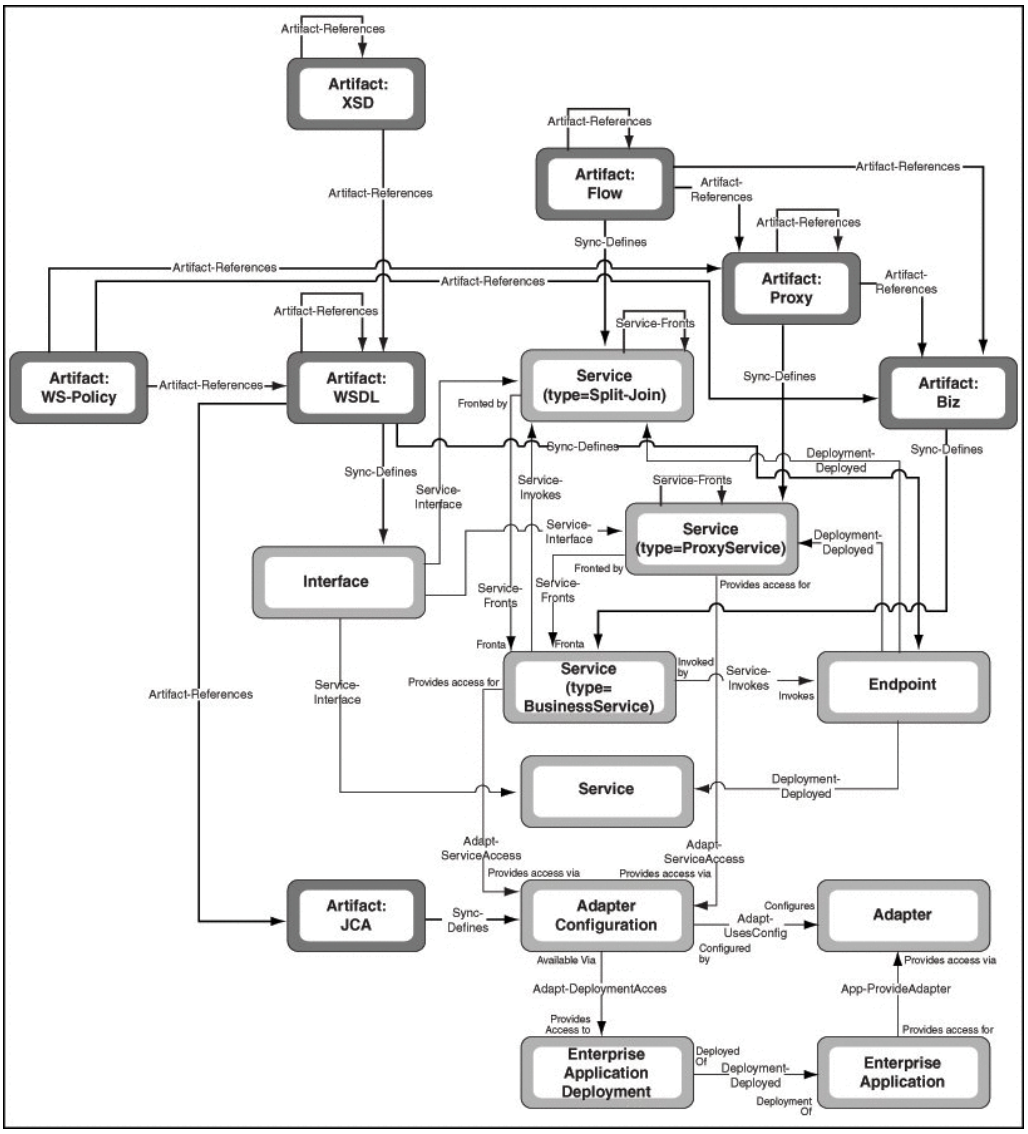
  - There is no WSDL associated with this service. So it will not be available to integrate with UDDI using the RR-XU utility. And the WSDL can not be consumed via the Oracle Enterprise Repository Web UI's "`Use|Download`" feature, or the "`Service Consumption Framework`" feature in Oracle Eclipse Workshop.

- 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.

6.3.1.9 Oracle Service Bus (OSB) Project

An Oracle Service Bus project can be submitted to Oracle Enterprise Repository from the command line. The project must be exported from Oracle Service Bus Workshop, using the command `Export|Oracle Service Bus: Configuration jar`.

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 instead of the harvester supplied with the Oracle Enterprise Repository installation. Similarly, for harvesting AIA, use the AIA harvester.

6.3.1.10 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.

6.3.2 Harvesting Oracle Business Process Analysis Projects

Oracle Enterprise Repository Harvester 11*g* supports harvesting of SOA Oracle Business Process Analysis (BPA) Projects if the BPA Projects are converted to SOA composites. The steps to perform are as follows:

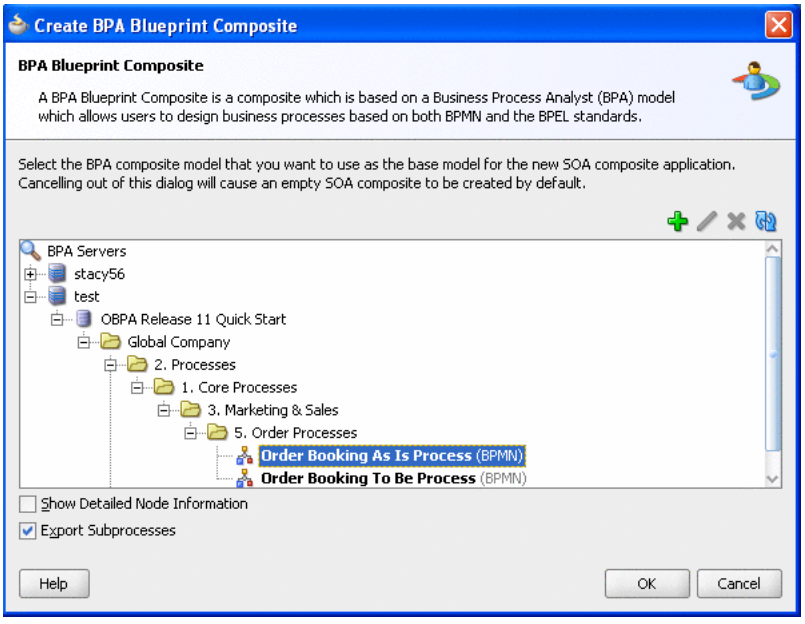
> **Note:** Oracle JDeveloper 11*g* Release 1 (11.1.1.2.x) does not support importing of BPEL process from BPA server. You need to use 11*g* Release 1 (11.1.1.7) JDeveloper to import.

Step 1: Convert the BPA Project to SOA Composite

1. Design a BPA process using the BPA Architect.
2. Right-click the designer window, select **SOA**, and **Transform Business Process into BPEL Process**. Right-click the designer window, and select **SOA**, **Share Blueprint with IT**.
3. Open JDeveloper and create BPA server connection, if it is not already done.
4. Create a new application and then create a new SOA project.
5. In the Configure SOA Settings dialog, select **Composite From Oracle BPA Blueprint**, and then click **Finish**.

   For more information about creating a new application and a SOA project, see *Oracle Fusion Middleware Developer's Guide for Oracle SOA Suite*.
6. Expand BPA Server and choose the required BPMN process, as shown in Figure 6–5. Click **OK** to create SOA project. For more information about BPA Server, see the BPA documentation at *http://www.oracle.com/technology/software/products/bpa/index.html*.

Figure 6–5 Create BPA Blueprint Composite Dialog



7. Open BPEL Process and fix all compilation errors (if any). Close the Application.

Step 2: Harvest the Converted BPA Project

1. Harvest the Oracle SOA Suite project. For more information, see Section 6.3.14.9, "Harvesting SOA Projects".
2. Open JDeveloper 11g Release 1 (11.1.2). Open the above application and right-click and select **Harvest Project** option to harvest into Oracle Enterprise Repository.

6.3.3 Harvesting Oracle Business Process Management 11*g* Projects

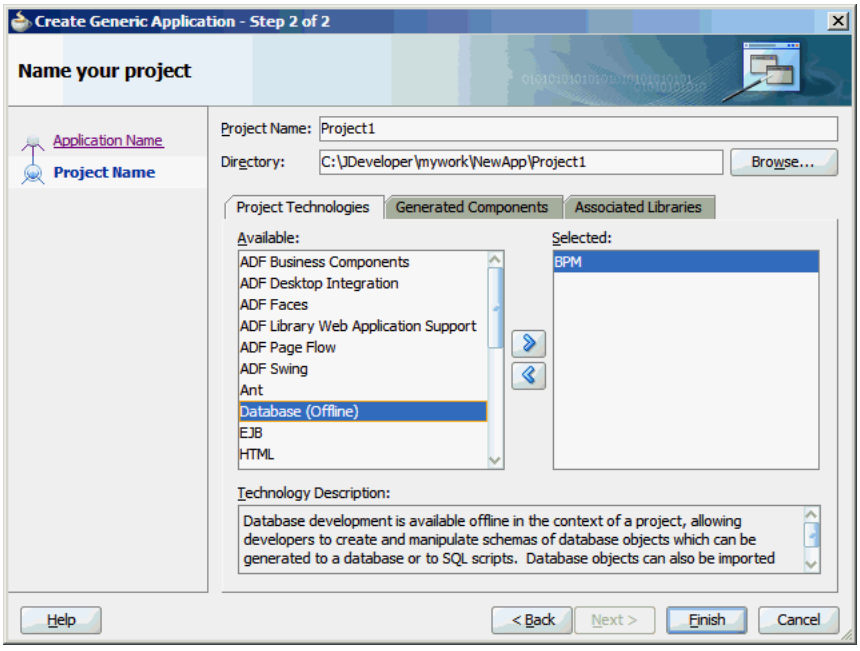
Oracle Enterprise Repository Harvester 11*g* supports harvesting of Oracle Business Process Management (Oracle BPM) projects. As part of the Oracle SOA 11*g*, Oracle JDeveloper supports the design and development of Oracle BPM projects with the BPMN process. You can develop a project as a composite with the BPMN component, and then harvest the BPM project to Oracle Enterprise Repository. The steps to perform are as follows:

Step 1: Create BPM Project in JDeveloper

To create a BPM project:

1. In the Application Navigator of JDeveloper, click **New Application**. The Create Generic Application - Name your application page is displayed.
2. Enter a name in the Application field and click **Next**. The Create Generic Application - Name your project page is displayed.
3. Enter a name for the BPM project in the Project Name field.
4. In the Available list under the Project Technologies tab, double-click BPM to move it to the Selected list, as shown in Figure 6–6.

Figure 6–6 The Create Generic Application - Name your project Page



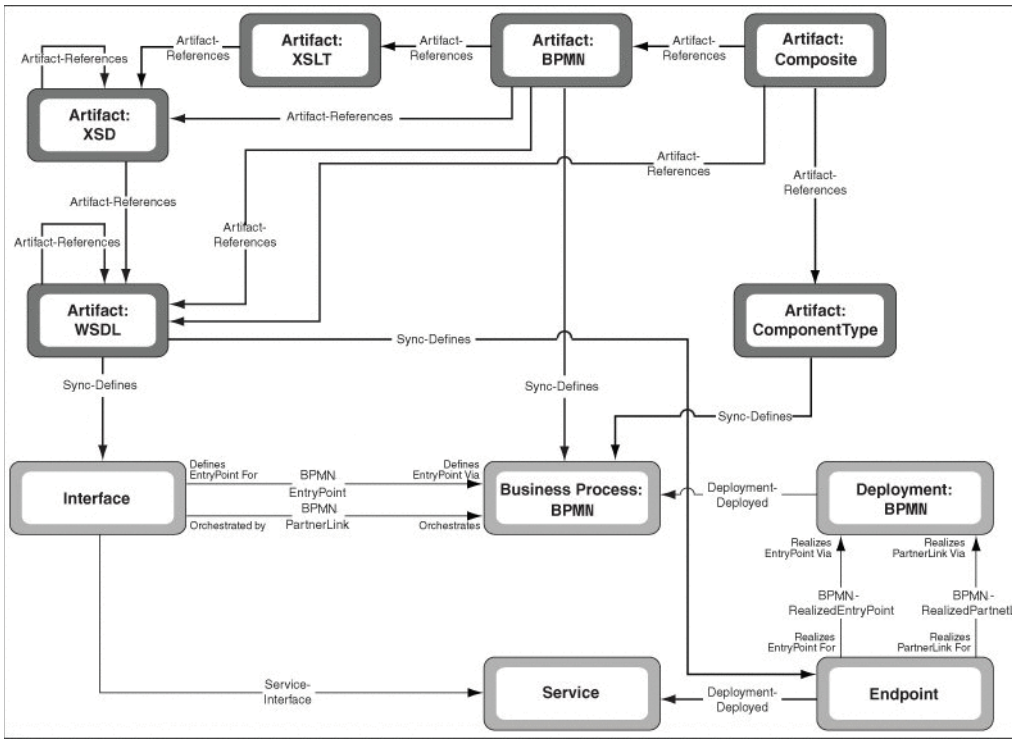
5. Click **Finish**.

Step 2: Design a BPM Process and Deploy to SOA Server

For more information on how to design a BPM process and deploy to SOA server, see "Creating a BPM Project from a BPA Project" in the *Oracle Fusion Middleware Modeling and Implementation Guide for Oracle Business Process Management*.

Step 3: Harvest the BPM Project to Oracle Enterprise Repository from JDeveloper and Deploy to Oracle SOA Server

- Open JDeveloper 11g Release 1 (11.1.2).
- Right-click and select **Harvest Project** option to harvest into Oracle Enterprise Repository.



6.3.4 Harvesting Oracle Service Bus 11*g* Assets

For more information about Harvesting Oracle Service Bus 11g assets see *Oracle Fusion Middleware Developer's Guide for Oracle Service Bus*.

6.3.5 Harvesting Oracle Service Bus 10*g* Assets

You can harvest Oracle Service Bus 10*g* assets into Oracle Enterprise Repository 11*g*. The following instructions cover Workspace Studio and Oracle Service Bus Eclipse, and work for Oracle Service Bus 10.3 and Oracle Service Bus 10.3.1.

The following steps assume that Oracle Service Bus 10.3.1 is installed in `BEA_HOME` and Oracle Enterprise Repository harvester is installed in `HARVESTER_HOME`:

1. Download `com.bea.alsb.harvester.connector_1.0.200.zip`, the Harvester Plugin, and unzip to the *<BEA_HOME>\osb_10.3\eclipse\plugins* directory.
2. Download and unzip `11.1.1.x.x-OER-Harvester.zip` to the <dir> directory. The `HARVESTER_HOME` will be `<dir>/harvester`.
3. Edit `<BEA_HOME>\workshop_10.3\workshop.ini`. Add the following

```
-Dharvester.home=HARVESTER_HOME
```

- To run from command line, use **osb10harvest.sh** for Linux and **osb10harvest**.bat for Windows.
- To run from ant, use **osb10runant.sh** for Linux and **osb10runant.bat** for Windows.

To Harvest from Oracle Service Bus Eclipse

1. Launch Oracle Workshop for WebLogic.
2. Import the Oracle Service Bus configuration jar file.
3. Right-click the Oracle Service Bus project, and then click **Submit to Oracle Enterprise Repository**. The Submit Assets to Oracle Enterprise Repository dialog is displayed.
4. Fill in the connection parameters and then click **Submit**.

See Section 6.3.1.9, "Oracle Service Bus (OSB) Project"for more information.

6.3.6 Harvesting a SOA Project from JDeveloper

You can harvest a SOA project to Oracle Enterprise Repository from Oracle JDeveloper. To harvest a SOA project, perform the following steps:

1. Open Oracle JDeveloper and create an Oracle Enterprise Repository Connection.
   a. Click the **Resource Palette** tab.
   b. Click **New**, **New Connection**, **Oracle Enterprise Repository**. The Create Oracle Enterprise Repository Connection dialog is displayed.
   c. Select **IDE Connections**.
   d. Enter a name for the Oracle Enterprise Repository connection in the Connection Name field.
   e. Enter the repository URL, for example, http://localhost:port/oer.
   f. Enter a valid username/password for the repository connection and click Test Connection. After a successful status is displayed in the Status pane, the Oracle Enterprise Repository connection appears under the IDE Connections tab.
2. In the Application Navigator, right-click the SOA project node and select **Harvest Project**. The Harvest SOA Project dialog is displayed.
3. Select the repository connection you created in Step 1 in the Repository Connection list.
4. Select **Unsubmitted** in the Registration Status list.
5. Enter a description for the harvesting in the Harvester Description field.
6. Click OK. This runs an ant script to harvest SOA composite to Oracle Enterprise Repository.
7. Similarly, you can also harvest a XSD or WSDL to Oracle Enterprise Repository. In the Application Navigator, right-click the appropriate node and select **Harvest File**. This runs an ant script to harvest the schema/wsdl to Oracle Enterprise Repository.

6.3.7 Harvesting Deployed Composites from JDeveloper

You can also harvest a deployed SOA composite to Oracle Enterprise Repository from Oracle JDeveloper. To harvest a deployed SOA composite from Oracle JDeveloper, perform the following steps:

1. Open JDeveloper that contains your deployed SOA composites.
2. Right-click the SOA composite node in the Application Server navigator and select **Submit to Enterprise Repository**. The Submit to Enterprise Repository dialog is displayed.
3. Enter the following details:
   - Repository Connection: Select the repository connection to which you want to submit the deployed composite.
   - Registration Status: Select the appropriate registration status from the list.
   - Composite Name and Revision: The name of the composite that you want to deploy is displayed in this field.
   - Partition Name: The default value is default.
   - Remote Server URL: Enter the URL of the remote server.
   - Show Debug Information: Select this option if you want to view the debug information.

You can also consume services, schemas, XSLTs, and events from Oracle Enterprise Repository. You must associate a JDeveloper application with Oracle Enterprise Repository project to enable usage tracking.

Note: If you want to harvest individual assets such as WSDL/XSDs, then use the harvester from command line.

6.3.8 Associate JDeveloper Application with Oracle Enterprise Repository Project

1. Click **Application Menu** that appears to the right of the application in the Application Navigator.
2. Select **Application Properties**. The Application Properties dialog is displayed.
3. Click Repository in the left pane. The Repository page is displayed in the right.
4. In the Repository Connection list, select the Oracle Enterprise Repository connection that you want to use for usage tracking.
5. In the Repository Project list, select the Oracle Enterprise Repository project that you want to use for usage tracking.
6. Click **OK**.

6.3.9 Consume WSDL Services in Oracle Enterprise Repository from Jdeveloper

1. In JDeveloper, open the composite editor.
2. Drag and drop the Web Service component from the Component Palette to External References swim lane. The Create Web Service dialog is displayed.
3. Enter a reference name in the Name field.
4. Click the Find Existing WSDL icon at appears at the extreme right of the WSDL URL field. The SOA Resource Browser dialog is displayed.

5. Select Resource Palette from the list.
6. Click **IDE Connections**, **Oracle Enterprise Repository**, <*Connection Name*>, **Asset Types**, **Service**, and then select the service you want to invoke or consume.

   If the service has only one WSDL or UDDI key associated with it, then the same WSDL or UDDI key is used to create the reference. If service has more than one WSDL and/or UDDI key associated with it, then the Choose WSDL dialog is displayed. You need to select one of the URLs/UDDI keys to consume.

   For resolving UDDI keys in JDeveloper, you have to create a UDDI connection prior to creating the reference, without which you cannot select UDDI keys.

   Note: Follow the same procedure to browse for schemas (XSDs), events (EDL), transformations (XSLT), except that you will select the respective Artifacts instead of Service.

6.3.10 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*), 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.

6.3.11 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 *Oracle Fusion Middleware User's Guide for Oracle Enterprise Repository*.

6.3.12 Searching Harvested Assets

You can search for harvested assets using either of the following two methods:

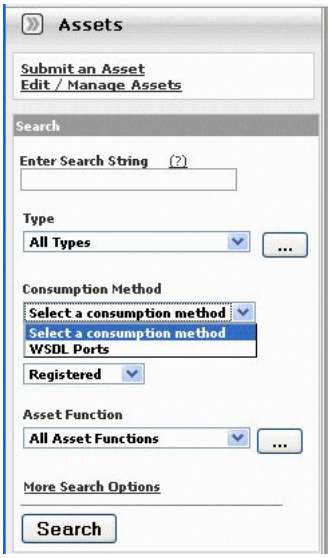
- Section 6.3.12.1, "Using Consumption Method"

- Section 6.3.12.2, "Using Metadata"

6.3.12.1 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 6–7, 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 6–7 Assets Pane - Consumption Method



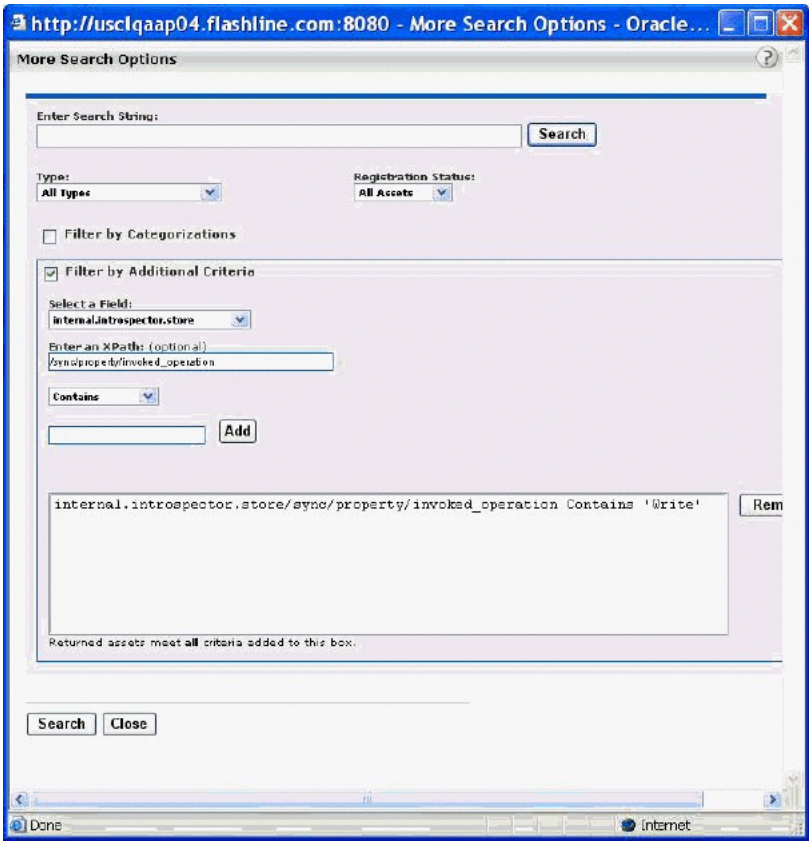
6.3.12.2 Using Metadata

The Harvester tags each asset with properties that can be used for querying.

Figure 6–8 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 6–8 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

6.3.13 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>
```

6.3.14 Best Practices

This section describes best practices for the Harvester. It contains the following topics:

- Section 6.3.14.1, "Recommended Privileges for Harvesting"
- Section 6.3.14.2, "Do Not Override the Namespace Parameter"
- Section 6.3.14.3, "Namespaces in WSDL Files"
- Section 6.3.14.4, "Harvesting Completed Work"
- Section 6.3.14.5, "Harvesting and Maintenance Releases of XSD"
- Section 6.3.14.6, "Harvesting Metadata throughout the Asset Lifecycle"
- Section 6.3.14.7, "Downloading WSDL Artifacts"
- Section 6.3.14.8, "Harvesting Retired Endpoints"
- Section 6.3.14.9, "Harvesting SOA Projects"
- Section 6.3.14.10, "Harvesting from Oracle JDeveloper 11g"
- Section 6.3.14.11, "Harvesting a File in the Harvester Folder"
- Section 6.3.14.12, "Running Harvester and Exchange Utility on the Same Computer"
- Section 6.3.14.13, "Harvesting Composite Service WSDL"

6.3.14.1 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.

6.3.14.2 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 (JDev and runtime). Changing or overriding the namespace leads to duplicate assets in OER.

6.3.14.3 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 6–7 shows the correlation of Oracle Enterprise Repository assets with WSDL structure:

Table 6–7 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`

6.3.14.4 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.

6.3.14.5 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.

6.3.14.6 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. Design/Development time

   After implementation, assets should be published to Oracle Enterprise Repository. Any policies associated with the assets will also be published to Oracle Enterprise Repository. The Oracle Enterprise Repository Governance process provides a means of verifying that the implementation aligns with the organization's policies before it is made available for reuse to the general consumer community.

   Recommended approach: Invoke the harvester from the IDE (JDeveloper, Oracle Service Bus Workshop, Eclipse or VS.NET).

2. 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.

3. 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. For more information about Amberpoint integration, see "Chapter 5, Integration with Amberpoint" in *Oracle Fusion Middleware Integration Guide for Oracle Enterprise Repository*.

6.3.14.7 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.

6.3.14.8 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.

6.3.14.9 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.

6.3.14.10 Harvesting from Oracle JDeveloper 11*g*

It is not recommended to change namespace when harvesting from Oracle JDeveloper 11*g*.

Also, it is recommended not to harvest same composite from JDeveloper 11*g* using two different status, for example, registered and submitted.

6.3.14.11 Harvesting a File in the Harvester Folder

It is recommended not to harvest a file that is located within the unzipped Harvester folder, `11.1.1.x.x-OER-Harvester.zip`. Currently, this is not supported.

6.3.14.12 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.

6.3.14.13 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.

6.3.15 Known Issues

This section describes the following known Harvester issues:

- Section 6.3.15.1, "Asset Types Must be Present in the System"
- Section 6.3.15.2, "Two Versions of an Asset Type"
- Section 6.3.15.3, "Do Not Delete the Harvester-Specific Metadata Entries in a in a Harvested Asset"
- Section 6.3.15.4, "Supported Remote Server Types"
- Section 6.3.15.5, "Using Incorrect Encrypted Password"
- Section 6.3.15.6, "View in Repository Option"

6.3.15.1 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 `11.1.1.x.x-OER-Harvester-Solution-Pack.zip`.

6.3.15.2 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 `11.1.1.x.x-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.

6.3.15.3 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.

6.3.15.4 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 10.x.

6.3.15.5 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.Ill
egalArgumentException: 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(DefaultMetada
taLogger.java:98)
        at
@ com.oracle.oer.sync.plugin.writer.oer.OERConnectionCache.getAuthToken(OERCon
nectionCache.java:138)
        at
@ com.oracle.oer.sync.plugin.writer.oer.OERAssetQueries.getToken(OERAssetQueri
es.java:82)
        at
@ com.oracle.oer.sync.plugin.writer.oer.OERAssetQueries.assetTypeQueryByUUID(AL
ERAssetQueries.java:159)
        at
@ com.oracle.oer.sync.framework.MetadataManager.putAssetType(MetadataManager.jav
a:204)
        at
@ com.oracle.oer.sync.framework.impl.DefaultPluginManager.processIntrospector(De
faultPluginManager.java:104)
        ... 4 more
```

6.3.15.6 View in Repository Option

From Oracle JDeveloper 11g, 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 will not work.

As a workaround, you need to login to Oracle Enterprise Repository console and click links for further information.

7

Configuring Metrics

This chapter describes how to configure metrics in your organization.

This chapter contains the following sections:

- Section 7.1, "Overview"
- Section 7.2, "Validate the Elements on the Asset Type"
- Section 7.3, "Running Queries Against the Database"

7.1 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.

7.1.1 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

7.1.2 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:

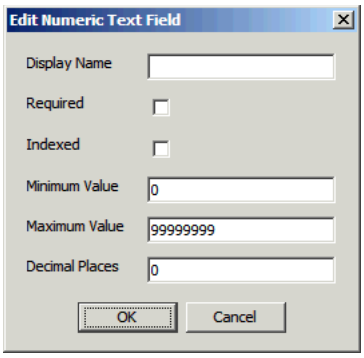
- Total development hours (TDH)
- Production investment (Pinv)
- Consumption Factor (Cfac)
- Predicted number of annual reuse opportunities (n)
- Hourly burden rate (B)

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 7–1.

Figure 7–1 Edit Numeric Text Field Dialog



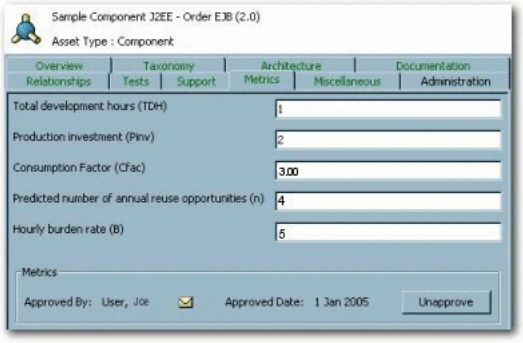
9. Click **OK**.
10. Repeat steps 4 - 9 for each metric.
11. When all metrics have been added to the Type, click **Save**.
    - Do not add the elements to the Viewer tab at this time.
12. Repeat the process for each of the base asset types.

7.2 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 7–2.

Figure 7–2 Administration Tab



3. Determine the asset name, version and asset type name from the status bar on the edited asset.
4. Save the asset.

7.3 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
```

7.3.1 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.

8

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 8.1, "Overview"
- Section 8.2, "Setting Up E-Mail Notification"
- Section 8.3, "Managing E-mail Distribution Lists"
- Section 8.4, "Working with E-Mail Templates"

8.1 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.

8.1.1 Using E-Mail Notifications and E-Mail Distribution Lists

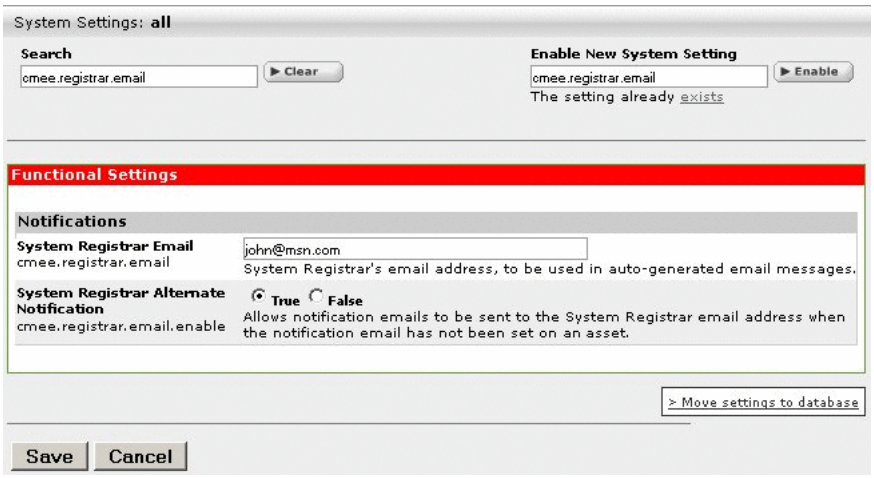
Users can create e-mail distribution lists so that the list members are automatically copied on certain e-mail notifications that are generated as part of the asset registration process. Distribution lists can help automate the asset registration workflow because members of the distribution list are 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.

8.2 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 8–1. Click **Save** to save the settings.

Figure 8–1 Functional Settings

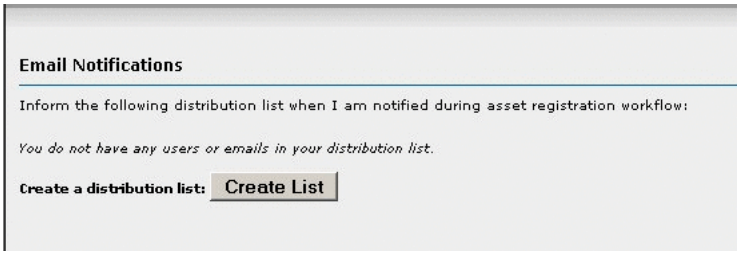


8.3 Managing E-mail Distribution Lists

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 8–2.

Figure 8–2 E-mail Notifications

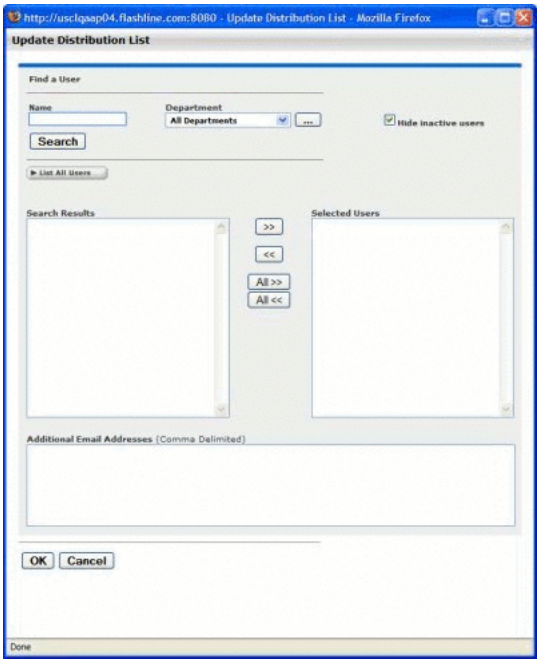


Creating a Distribution List

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 8–3.

Figure 8–3 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**.

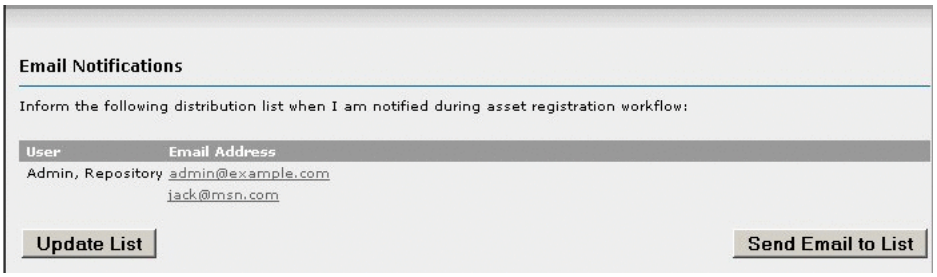
   Search results appear in the list in the Search Results column.
3. Add users to the distribution list by using the >> and << icons to move users from the Search Results column to the Selected Users column. To add or remove all users from either column, use the All >> and All << buttons.

   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@msn.com,jill@yahoo.com,jake@comcast.net
5. Click **OK** to save the distribution list.

Figure 8–4 E-mail Notifications



8.3.1 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 Creating a Distribution List.
3. Click **OK** to save the distribution list.

8.3.2 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

8.4 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.

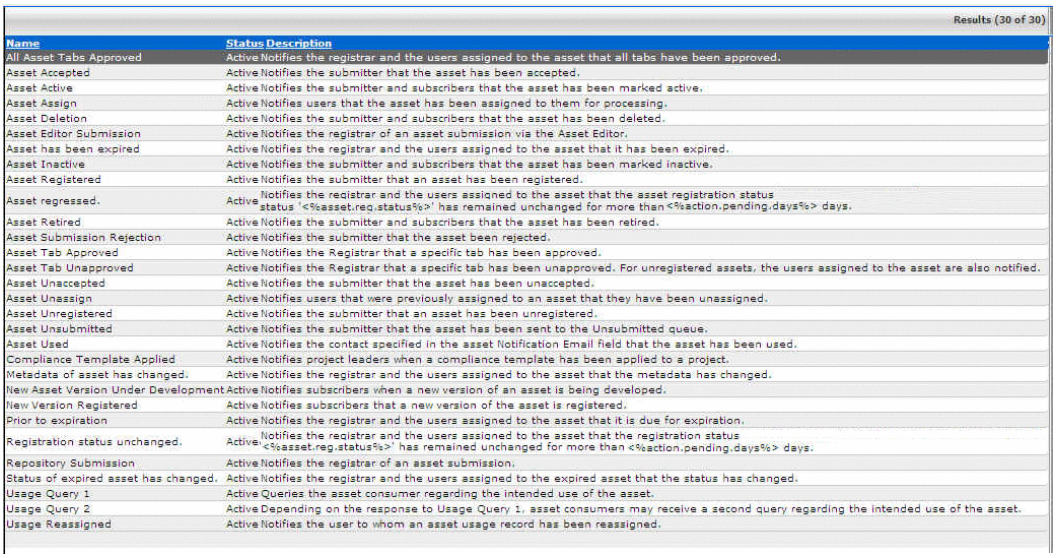
  This is a notification to remind the user to submit the survey regarding this asset. The survey can be found in the Web Tool under the My Stuff tab within the Consumed Assets section. In the detail of the given asset is a button called Update Survey Status. This button will only be available if the Consumer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

8.4.1 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 8–5.

Figure 8–5 List of E-mail Templates

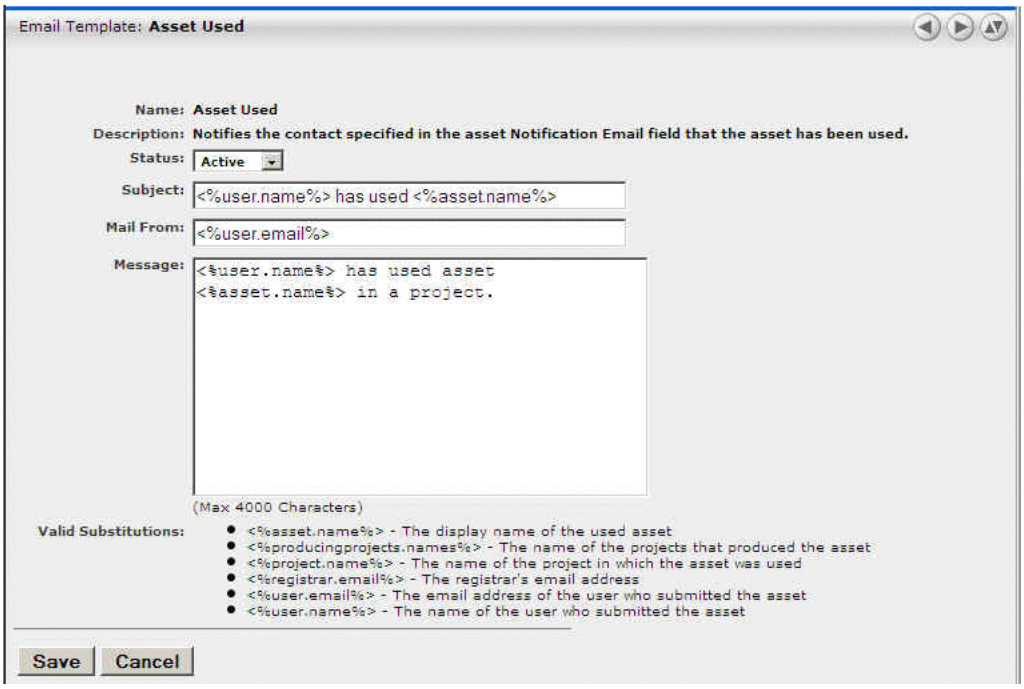


Results (30 of 30)

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

4. Click any template to display the details of that E-mail Template in the pane below. For example, click Asset Used. Figure 8–6 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 8–6 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

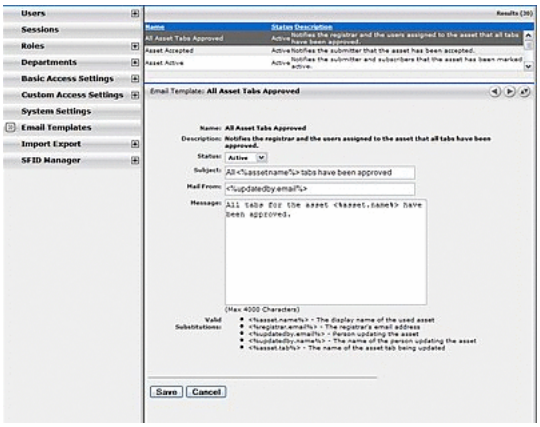
5. Click **Save**.
6. Similarly, you can view the details of the other E-mail Templates.

8.4.2 Editing E-Mail Templates

The procedure begins on the Oracle Enterprise Repository Admin screen.

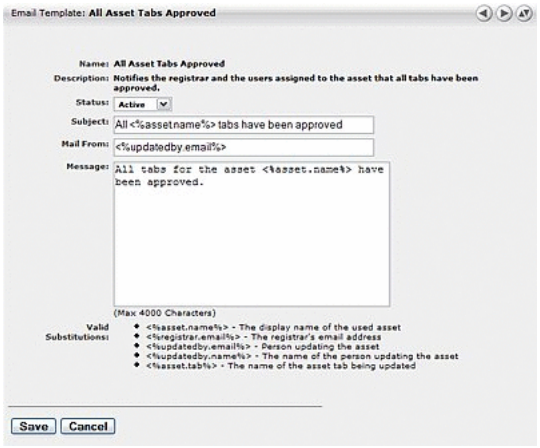
1. Click **E-mail Templates** in the Admin sidebar. The list of e-mail templates appears in the upper frame of the main pane.

Figure 8–7 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 8–8 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.

9

Configuring Oracle Enterprise Repository Express Workflows

This chapter describes how to set up and configure Oracle Enterprise Repository (OER) 11g Express Workflows.

This chapter contains the following sections:

- Section 9.1, "Introduction to Oracle Enterprise Repository Express Workflow"
- Section 9.2, "Configuring Express Workflows"
- Section 9.3, "Running the Sample Workflow"
- Section 9.4, "Modifying the Sample Workflow"

This chapter assumes that you are familiar with the following concepts:

- Assets and Asset Types, Asset submission, Asset Registration Status, Asset Categorizations (Oracle Fusion Middleware User's Guide for Oracle Enterprise Repository)
- Projects and Users (Oracle Fusion Middleware Configuration Guide for Oracle Enterprise Repository)
- Oracle SOA Suite application deployment (Oracle Fusion Middleware Developer's Guide for Oracle SOA Suite)

9.1 Introduction to Oracle Enterprise Repository Express Workflow

Express Workflows, as described in this section, require:

- OER PS6 (11.1.1.7), with Patch 17420982 applied
- Oracle SOA Suite + Oracle BPM Suite, both 11.1.1.7

> **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 Part IV of the Oracle® Fusion Middleware Integration 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.

9.2 Configuring Express Workflows

Express Workflows depends on these components:

- Express Workflows SOA composite application (SCA)
- OER Event Manager
- Express Workflows configuration file

Here are the steps involved for the configuration:

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 10.3.2. It contains the following topics:

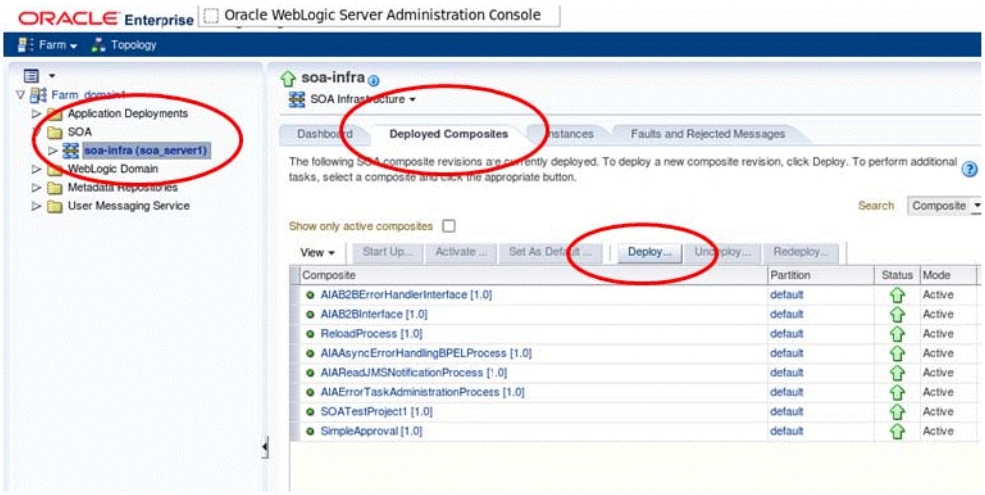
9.2.1 Step 1: Install the OER Express Workflows SOA Composite Application

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 11g R1+ with BPM Suite. No additional configuration is required in your SOA Suite and BPM Suite servers.

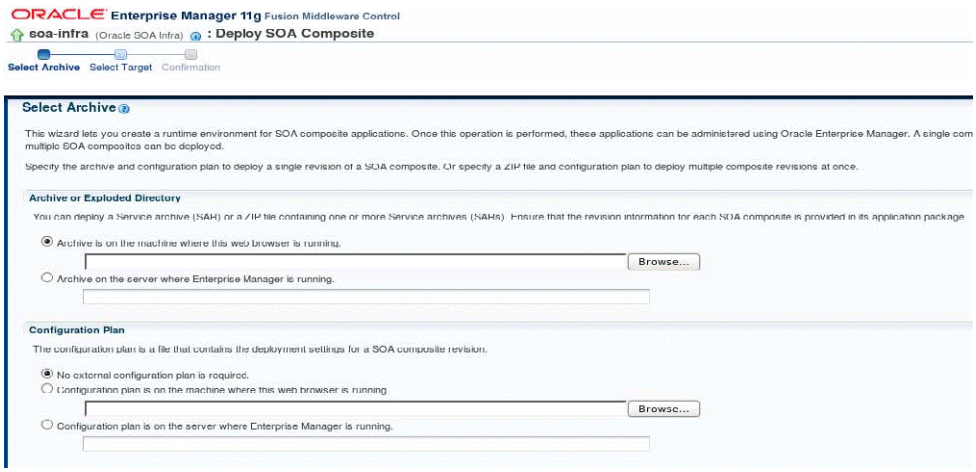
Note: Ensure that payload validation under soa-infra-->SOA Administration-->Common Properties is set to false before you proceed.

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.
8. Upload the composite delivered with the ExpressWorkflows patch:
9. Click Next.
10. Select the default SOA partition. Click Next.
11. Click Deploy.

9.2.2 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. For detailed information on options for other applications to access OER events, see [Chapter 9.4 of the Oracle Fusion Middleware Configuration Guide for Oracle Enterprise Repository].

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"

9.2.2.1 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 in the OER installation at <oer webapp name>\WEB-INF\classes\EndPointEventSubscription.xml

e.g. C:\ofm_wls1036\user_projects\applications\oer_domain\applications\oer_11.1.1.7.0\oer-app\WEB-INF\classes\ EndPointEventSubscription.xml

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:

Figure 9–1 Configuring OER Event Manager to deliver events to your SCA

```
<sub:EventSubscriptionData xmlns:sub="http://www.bea.com/infra/events/subscription"
xmlns:xsi='http://www.w3.org/2001/XMLSchema-instance'>

    <!-- Subscription for 11g Express Workflows -->
    <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>8002</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 ALEPM -->
            <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>weblogic</sub:username>
                    <sub:password>v2_1.JBOEAIV9jhhaGGJEn0vwPg==</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.DefaultHTTPEventNotifier</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 deliver
        <!--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 Example 1 above. The comments on the rest of the tags are self explanatory. The above example is available in the Express Workflows patch, located in the sample file <OERHome>\repository111\core\express-workflows-config\EndPointEventSubscription.xml

> **Note:** Passwords must be encrypted, using either of the two encryption tools provided with OER. See Section 5.2 Generating Encrypted Passwords in the Oracle Fusion Middleware Configuration Guide for Oracle Enterprise Repository.

Changes to this file will not take effect until the OER 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.

9.2.2.2 Enabling the Event Manager

With the Event Manager configuration done, now the Event Manager must be enabled 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.

9.2.3 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 it's XSD) is provided with a sample workflow preconfigured to run when the configuration steps are completed. In the OER installation, this sample file is at <OERHome>\repository111\core\express-workflows-config\expressWorkflow.xml

A more detailed explanation of the different elements of this XML is provided in the Modifying the Sample Workflow section further. For now, we will configure the file and enable the sample workflow.

Setting up the configuration file and enabling the sample workflow

1. Create a new OER user with the access to launch the Asset Editor, edit assets, and approve tabs. See Section 1.3 of the Oracle® Fusion Middleware Configuration Guide for Oracle Enterprise Repository for information on users, roles and Basic Access Settings.
2. Save the original sample ExpressWorkflow.xml with a new filename.
3. Edit ExpressWorkflow.xml: Enter the host: 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>
                <tabs>
                    <tab name="Overview"/>
<tab name="Technical"/>
<tab name="Documentation"/>
                </tabs>
            </tier>
            <tier name="Tier2">
                <approvers>
<oeruser>
                    <username>[Your New Username]</username>
</oeruser>
                </approvers>
                <tabs>
                    <tab name="Taxonomy"/>
                </tabs>
            </tier>
```

5. Encrypt the passwords in the file as per Section 5.2 Generating Encrypted Passwords in the Oracle® Fusion Middleware Configuration Guide for Oracle Enterprise Repository.
6. Save file.
7. Verify the OER server is running, then run the Express Workflows validation script (<OERHome>\repository111\core\express-workflows-config\validateWorkflow.sh) 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:

   <SOAHome>\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 the next section and run the sample.

9.3 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

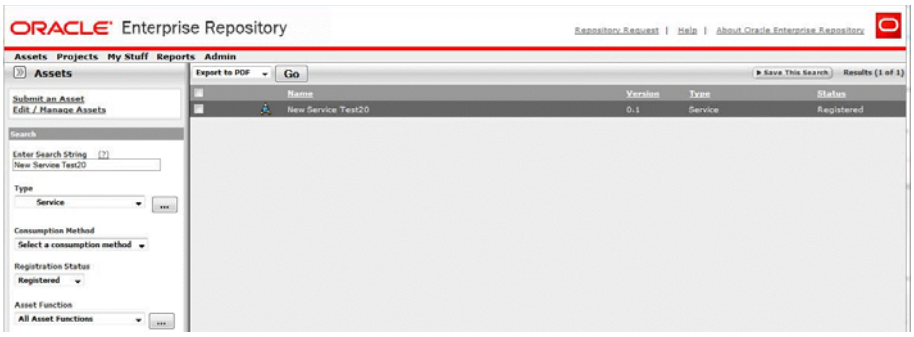
1. Login to OER as admin or any user with asset creation rights
2. From the Asset Editor, select File>New.
3. In the Create a New Asset, enter the asset name.
4. Select type, Service and initial state: Unsubmitted.
5. Click OK. The new asset tabs display.
6. On the Overview tab, select Common Project for Producing Project. Click Save.

Run the Sample

1. Login to OER as an admin user.
2. Go to "MyStuff", 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 as the new user you just created.
5. Go to MyStuff, 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.



9.4 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.

9.4.1 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

9.4.1.1 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 ExpressWorkflow only if they have one and only one producing project specified in the ExpressWorkflow.xml file.

9.4.1.2 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"/>
 <tab name="Technical"/>
 <tab name="Documentation"/>
                </tabs>
            </tier>
            <tier name="Tier2">
                <approvers>
 <oeruser>
                    <username>[Your New Username]</username>
 </oeruser>
                </approvers>
                <tabs>
                    <tab name="Taxonomy"/>
                </tabs>
            </tier>
```

- 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".
- Tier name identifies a combination of asset tabs.
- Tier approver indicates an existing OER user to whom the asset will be assigned, either upon asset acceptance for the first specified tier, of upon approval of all tabs of a previous tier.
- Tier tab name indicates specific asset tabs, which must exist for the specified asset types, which will be approved within a particular tier.

9.4.1.3 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 producing project:

```
<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" />
<tabs>
<tab name="Management Review" />
</tabs>
  </action>
</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 assetType 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.

10

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 10.1, "Getting Started With the Oracle Registry Repository Exchange Utility"
- Section 10.2, "Configuring the Oracle Registry Repository Exchange Utility"
- Section 10.3, "Using the Oracle Enterprise Repository Exchange Utility"

10.1 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 10.1.1, "What is the Oracle Registry Repository Exchange Utility?"
- Section 10.1.2, "Example Use Cases"

10.1.1 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.

10.1.1.1 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 10.3.2.1, "Synchronizing the Metadata Published from Oracle Enterprise Repository to Oracle Service Registry" and Section 10.3.2.2, "Synchronizing the Metadata from Oracle Service Registry to Oracle Enterprise Repository".
- 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 10–4.
- 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.

10.1.1.2 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.

10.1.2 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.
>
> For more information, see Section A.2.2, "Automated Workflows".

- Exchange Utility is capable of bringing in services from third-party UDDI registries such as SAP. For more information about integration with SAP, see the "Integration with SAP" chapter in *Oracle Fusion Middleware Integration Guide for Oracle Enterprise Repository*.
- Exchange Utility can also receive metrics from Oracle Service Registry, which are published to Oracle Service Registry from Oracle Business Transaction Management 10*g*. For more information, see the "Integration with Amberpoint" chapter in *Oracle Fusion Middleware Integration Guide for Oracle Enterprise Repository*.

10.2 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 10.2.1, "Installing and Configuring the Oracle Registry Repository Exchange Utility"
- Section 10.2.2, "Installing Oracle Service Registry"
- Section 10.2.3, "Configuring the Oracle Registry Repository Exchange Utility Configuration File"
- Section 10.2.4, "Configuring Oracle Enterprise Repository Categorizations in the UDDI Mappings File"
- Section 10.2.5, "Configuring the tModelKey UDDI Setting"

- Section 10.2.6, "Understanding the Oracle Registry Repository Exchange Utility's Property File"

10.2.1 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 10.2.1.1, "Install the Oracle Registry Repository Exchange Utility"
- Section 10.2.1.2, "Configure Using the Asset Type Name"

10.2.1.1 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>\repositoryXXX\core\tools\solutions`

The Oracle Registry Repository Exchange Utility files for Oracle Enterprise Repository 11*g* are as follows:

- **11.1.1.x.x-RR-ExchangeUtility.zip**: Contains the package for the Exchange Utility.

You can unzip the `11.1.1.x.x-RR-ExchangeUtility.zip` file into the directory on your file system where Oracle Enterprise Repository is installed, typically `Oracle_HOME\repositoryXXX`. 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.

10.2.1.2 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.

10.2.2 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*.

10.2.3 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 10.2.3.1, "Setting the Repository Connection Information"
- Section 10.2.3.2, "Setting the Registry Connection Information"
- Section 10.2.3.3, "Setting the Repository Query"
- Section 10.2.3.4, "Setting the Destination Registries"
- Section 10.2.3.5, "Setting the Registry Query"
- Section 10.2.3.6, "Setting the Source Registry"

10.2.3.1 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>/repositoryXXX/core/tools/solutions/11.1.1.x.x-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 5, "Diagnostics and Password Encryption".

10.2.3.2 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>/repositoryXXX/core/tools/solutions/11.1.1.x.x-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>/repositoryXXX/core/tools/solutions/11.1.1.x.x-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>
```

10.2.3.3 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 10.2.3.3.1, "Query by Name"
- Section 10.2.3.3.2, "Query by Registration Status"
- Section 10.2.3.3.3, "Query by Categorizations"
- Section 10.2.3.3.4, "Query by Filters"

10.2.3.3.1 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=""/>
```

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.

10.2.3.3.2 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

10.2.3.3.3 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="AssetFuntion" value="Governance"/>
```

10.2.3.3.4 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>
```

10.2.3.4 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.

10.2.3.5 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.

10.2.3.6 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>
```

10.2.4 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" />
```

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.

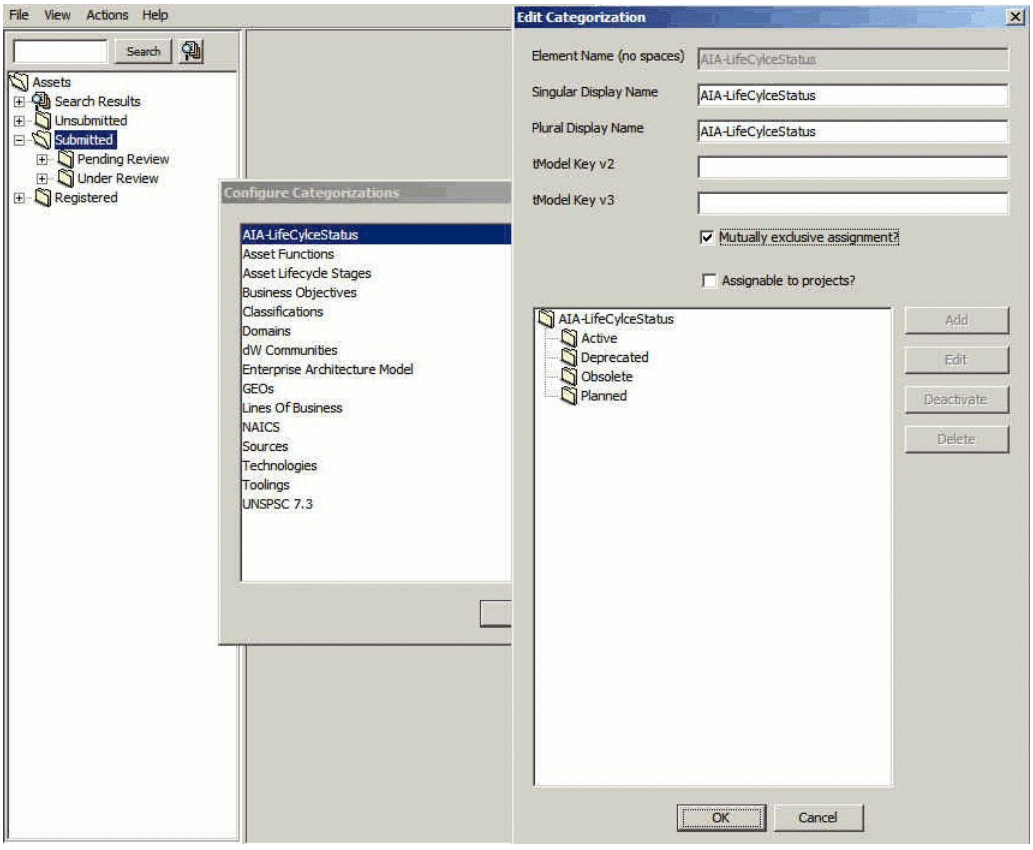
10.2.5 Configuring the tModelKey UDDI Setting

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 10–1. The key entered in this field must be the tModel Key of the associated Oracle Service Registry categorization.

Figure 10–1 The Edit Categorization Dialog



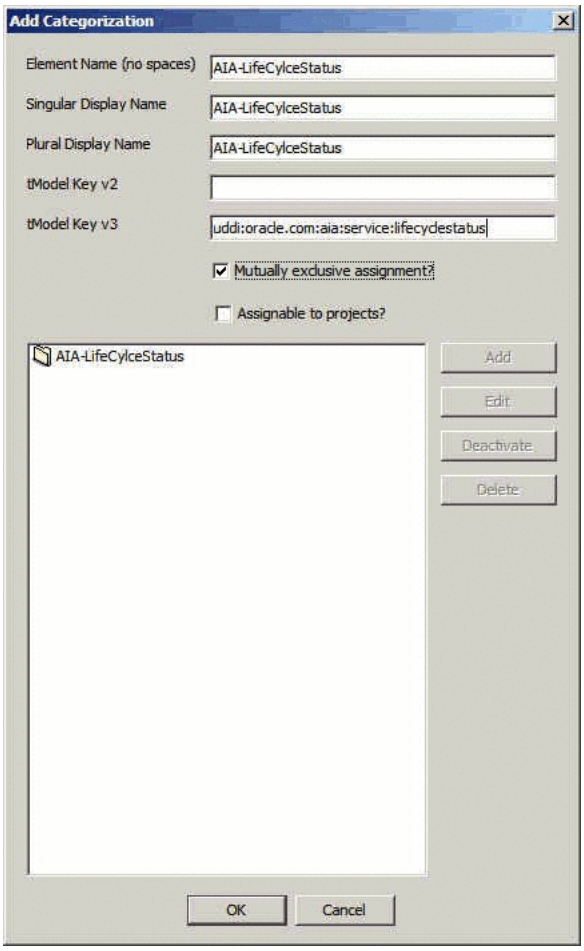
The steps to configure the tModelKey UDDI setting are as follows:

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-LifeCylceStatus` categorization.
3. Enter the values in the Add Categorization dialog, as shown in Figure 10–2.

Figure 10–2 The Add Categorization Dialog

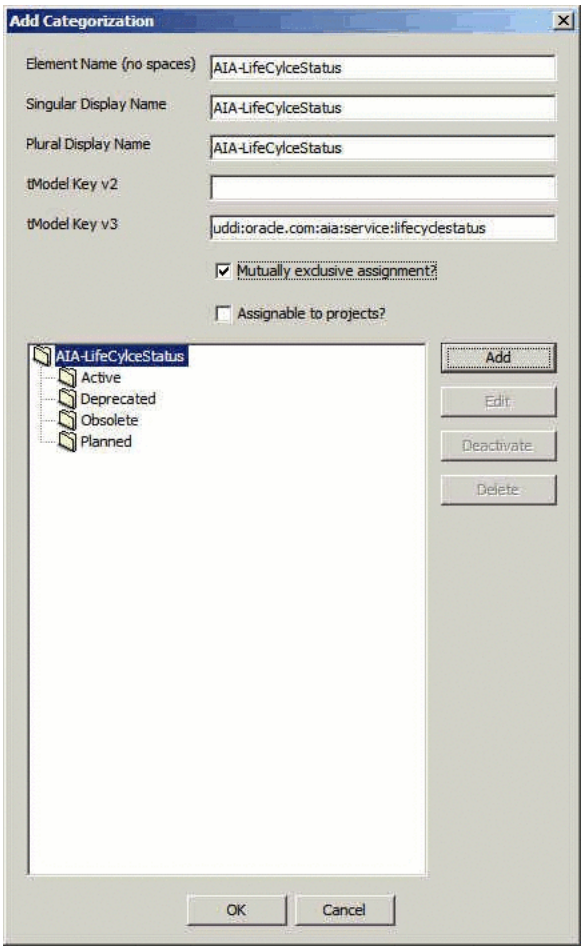


4. Select **AIA-LifeCycleStatus** in the pane below the tModel Key v3 field, and then click **Add**. The Add Categorization dialog is displayed.
5. Enter Active in the Name field, as shown in Figure 10–3.

Figure 10–3 The Add Categorization Dialog

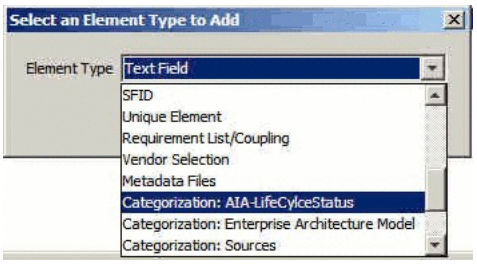
6. Click **OK**.
7. Similarly, add the following values for the categorization: Deprecated, Obsolete, and Planned, as shown in Figure 10–4.

Figure 10–4 The Add Categorization Dialog



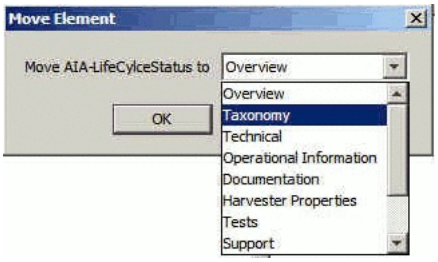
8. Click **OK**.
9. In the Asset Editor window, click **Actions**, **Manage Types**. The Type Manager window is displayed.
10. Select the **Service** asset type. The Service asset type details are displayed in the right pane.
11. In the Tabs pane, select **Taxonomy**. The corresponding elements for the 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 10–5.

Figure 10–5 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 10–6.

Figure 10–6 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 10–7 UDDIMappings.xml File

```
<categorizationType alerCategorizationTypeName="AIA-LifecycleStatus" alerCategorizationTypeId="50000"
                    uddiCategoryTModelKey="uddi:oracle.com:aia:service:LifecycleStatus">
    <categorization alerCategorization="Active" uddiKeyName="LifecycleStatus" uddiKeyValue="Active"/>
    <categorization alerCategorization="Deprecated" uddiKeyName="LifecycleStatus" uddiKeyValue="Deprecated"/>
    <categorization alerCategorization="Obsolete" uddiKeyName="LifecycleStatus" uddiKeyValue="Obsolete"/>
    <categorization alerCategorization="Planned" uddiKeyName="LifecycleStatus" uddiKeyValue="Planned"/>
</categorizationType>
```

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.

10.2.6 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 *Oracle Fusion Middleware User's Guide for Oracle Enterprise Repository*.

10.2.7 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.

> **Noteus:** All services are consumable by Oracle Service Bus in Oracle Service Registry unless otherwise noted.

Table 10–1 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.

10.2.7.1 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 10–2 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	▪ OSB will not consume the XSD ▪ Published as Messaging Service Type with Input XML/Output XML
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	▪ OSB will not consume the XSD ▪ Published as Messaging Service Type with Input XML/Output XML
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

10.2.7.2 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 10–3 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	▪ Not consumable by OSB in OSR. ▪ MFL artifacts do not have a remote URI in OER.
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.

10.2.7.3 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 10–1 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>
```

10.3 Using the Oracle Enterprise Repository Exchange Utility

This section describes how you can use the Oracle Registry Repository Exchange Utility to synchornize metadata and search for exchanged metadata in Oracle Enterprise Repository.

This section contains the following topics:

- Section 10.3.1, "Running the Oracle Registry Repository Exchange Utility"
- Section 10.3.2, "How the Exchanged Metadata Is Synchronized"
- Section 10.3.3, "Searching for Oracle Service Registry Exchanged Metadata in Oracle Enterprise Repository"
- Section 10.3.4, "Checking the Oracle Registry Repository Exchange Utility Log File"
- Section 10.3.5, "Known Issues"

10.3.1 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 10–4 defines the configuration options available when running the Oracle Registry Repository Exchange Utility.

Table 10–4 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.
		■ To publish Services to Oracle Service Registry from Oracle Enterprise Repository, use:
		`orrxu.bat -mode publish`
		■ To receive Services from Oracle Service Registry into Oracle Enterprise Repository, use:
		`orrxu.bat -mode receive`

`-config <File Name>`	Optional	Passes the `orrxu.xml` file as a parameter. Example usage: `orrxu.bat -mode publish -config C:\rm\uddi\orrxu.xml` If this parameter is omitted, the configuration XML file is loaded from the current directory where the orrxu.bat file is located using the system's Classpath.
`-map <Dir Name>`	Optional	Generates a `UDDIMappings.xml` 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 -config 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: `orrxu.bat -map c:/rm/uddi` For more information about the `UDDIMappings.xml` file, see Section 10.2.4, "Configuring Oracle Enterprise Repository Categorizations in the UDDI Mappings File".
`-publish_ tmodel <File Name>`	Optional	Publishes all the t-models configured in the `UDDIMappings.xml` 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 -config parameter from the default location. Example usage: `orrxu.bat - publish_tmodel c:/rm/UDDIMappings.xml` For more information about the `UDDIMappings.xml` file, see Section 10.2.4, "Configuring Oracle Enterprise Repository Categorizations in the UDDI Mappings File".

10.3.1.1 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 10.3.1.1.1, "Prerequisities"
- Section 10.3.1.1.2, "Synchronizing Using Timer Based Workflows"
- Section 10.3.1.1.3, "Synchronizing When Events are Triggered"
- Section 10.3.1.1.4, "Example Use Cases"

For more information about the `workflow.xml` file and configuring workflows, see Chapter A, "Configuring Oracle Enterprise Repository Workflow".

10.3.1.1.1 Prerequisities

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, `11.1.1.x.x-RR-ExchangeUtility.zip`, from *<Oracle_HOME>\repository111\core\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.

10.3.1.1.2 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 10–5 shows the names and descriptions of the timer based workflows:

Table 10–5 Timer Based Workflows

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

Table 10–5 (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>
```

10.3.1.1.3 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 differerent 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.

10.3.1.1.4 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.

10.3.2 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 10.3.2.1, "Synchronizing the Metadata Published from Oracle Enterprise Repository to Oracle Service Registry"
- Section 10.3.2.2, "Synchronizing the Metadata from Oracle Service Registry to Oracle Enterprise Repository"

10.3.2.1 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 10.3.2.1.1, "Business Entities"
- Section 10.3.2.1.2, "Service Keys"
- Section 10.3.2.1.3, "Endpoint"
- Section 10.3.2.1.4, "Categorizations"
- Section 10.3.2.1.5, "Registration and Active Status"
- Section 10.3.2.1.6, "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 10.3.5, "Known Issues".

10.3.2.1.1 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.

10.3.2.1.2 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.

For more information about modifications to asset types, see *Oracle Fusion Middleware User's Guide for Oracle Enterprise Repository*.

10.3.2.1.3 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.

10.3.2.1.4 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.

10.3.2.1.5 Registration and Active Status The Registration and Active Status are added to Category Bags. Figure 10–8 illustrates how references appear in Oracle Service Registry.

Figure 10–8 Oracle Service Registry t-model Categories for a WSDL Service

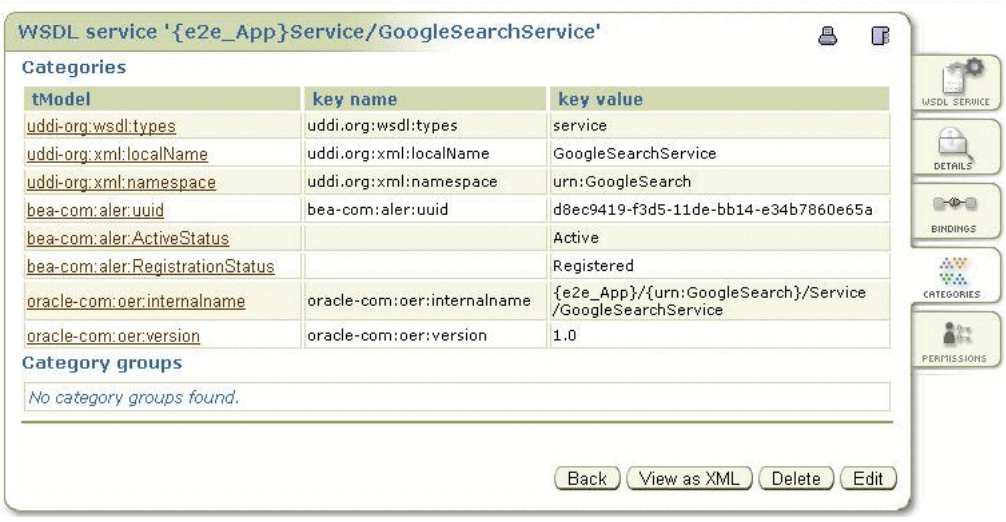


Figure 10–9 illustrates how two endpoints that are linked to a service in Oracle Enterprise Repository appear in Oracle Service Registry.

Figure 10–9 WSDL Bindings in Oracle Service Registry

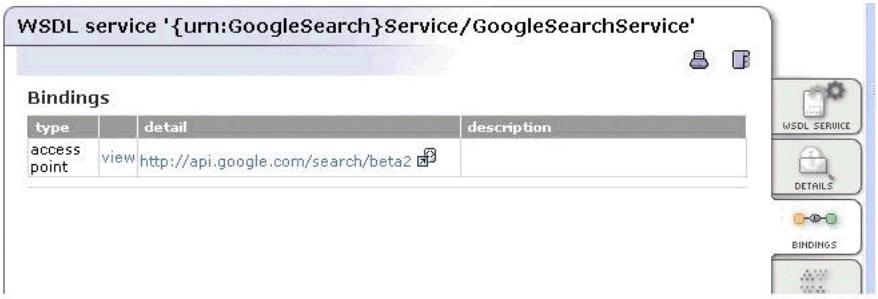


Figure 10–10 and Figure 10–11 illustrate how the different entities and their relationships appear in the Oracle Enterprise Repository Navigator.

Figure 10–10 Entity Relationship in Oracle Enterprise Repository Navigator

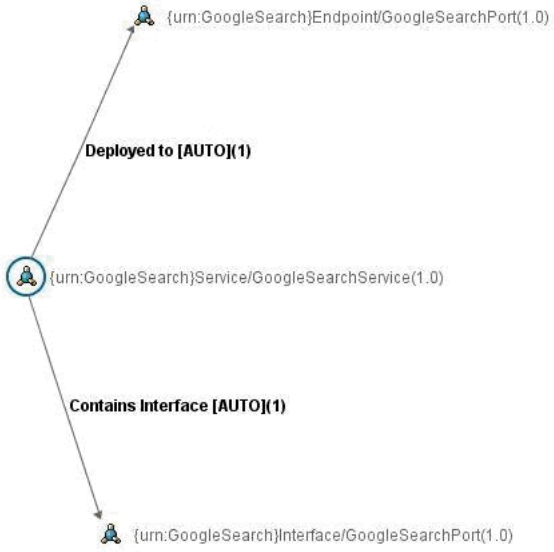
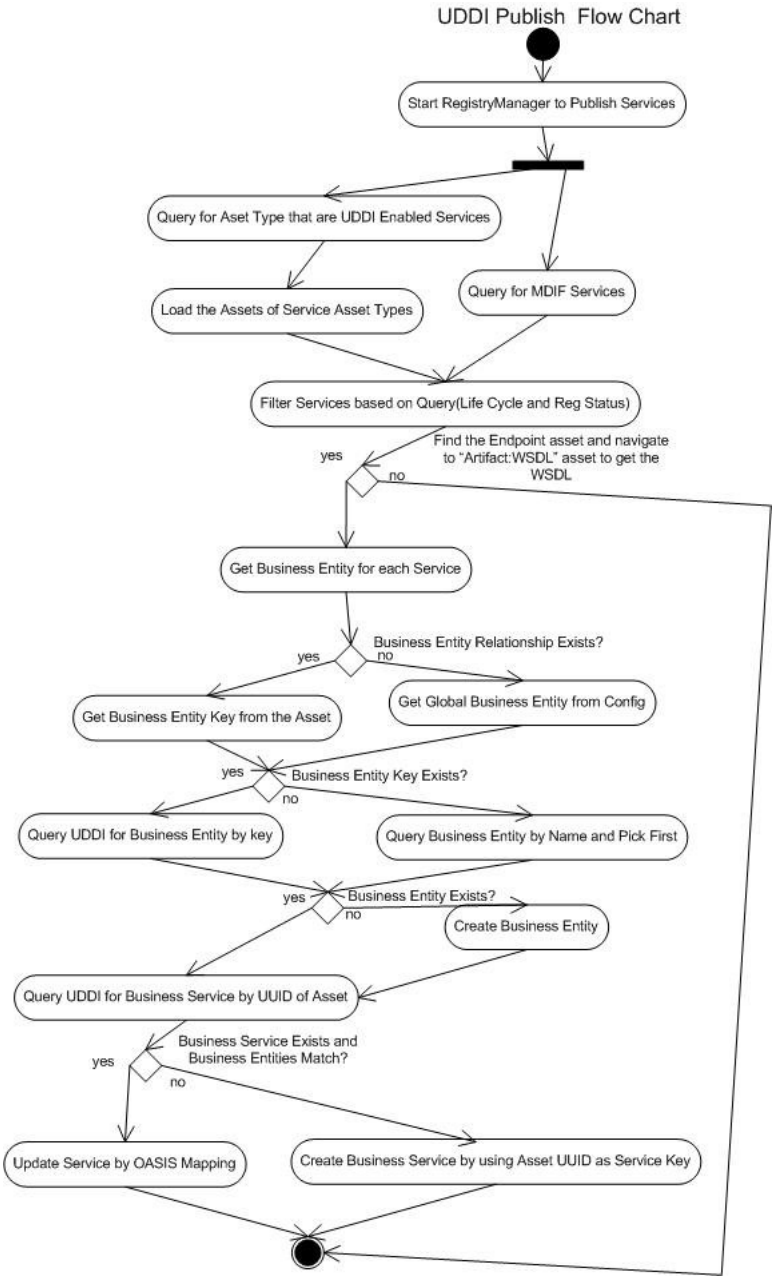


Figure 10–11 Another Entity Relationship in Oracle Enterprise Repository Navigator

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

Figure 10–12 illustrates the Oracle Enterprise Repository > Oracle Service Registry metadata synchronization described in this section.

Figure 10–12 Flow of Metadata Published from Oracle Enterprise Repository to Oracle Service Registry



10.3.2.2 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 10.3.2.2.1, "Business Entities"
- Section 10.3.2.2.2, "Endpoint"

- Section 10.3.2.2.3, "Categorizations"
- Section 10.3.2.2.4, "Sample Flow of Metadata from Oracle Service Registry to Oracle Enterprise Repository"

10.3.2.2.1 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.

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

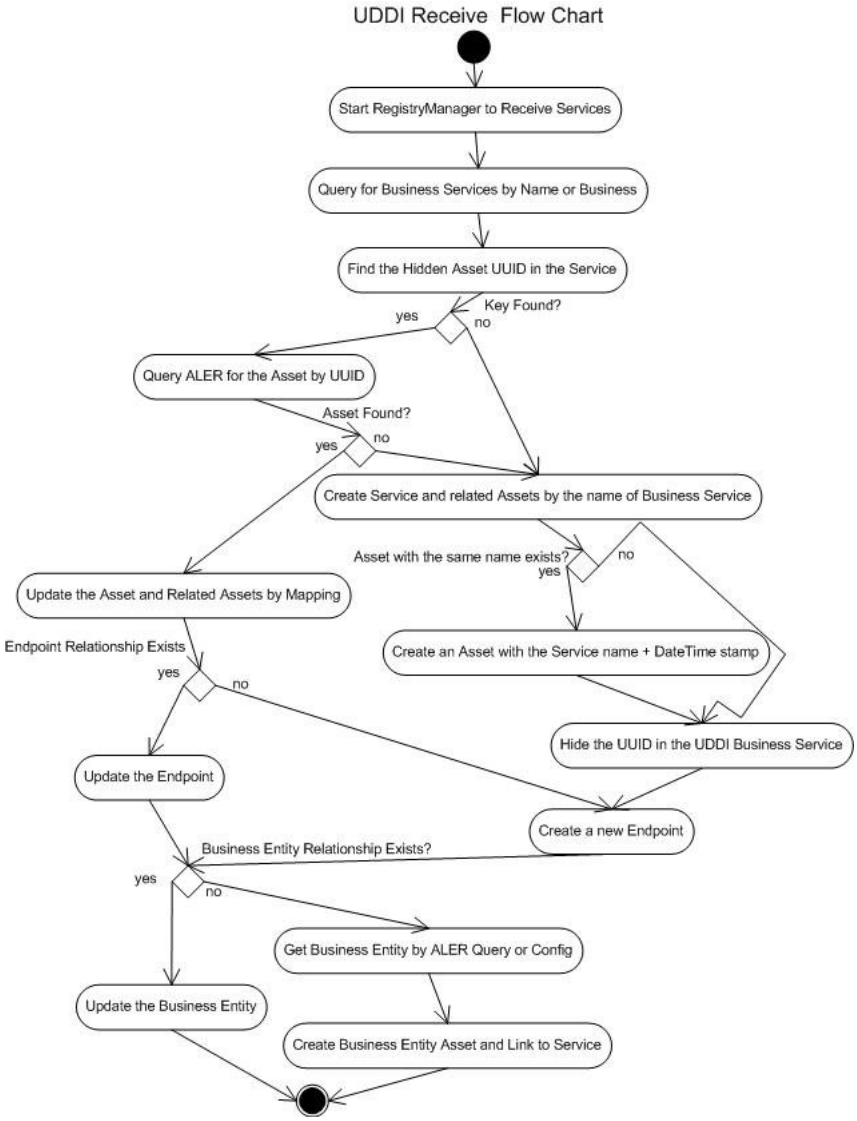
- If yes, create check if the Endpoint assets are the same as the existing Binding Templates using the UUID. If they are same, update the Endpoints.
- If no, create the Endpoints for each Binding Template and relate them to the Service asset.

10.3.2.2.3 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.

10.3.2.2.4 Sample Flow of Metadata from Oracle Service Registry to Oracle Enterprise Repository Figure 10–13 illustrates the Oracle Service Registry > Oracle Enterprise Repository metadata synchronization described in this section.

Figure 10–13 Flow of Metadata Received from Oracle Service Registry



10.3.3 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:

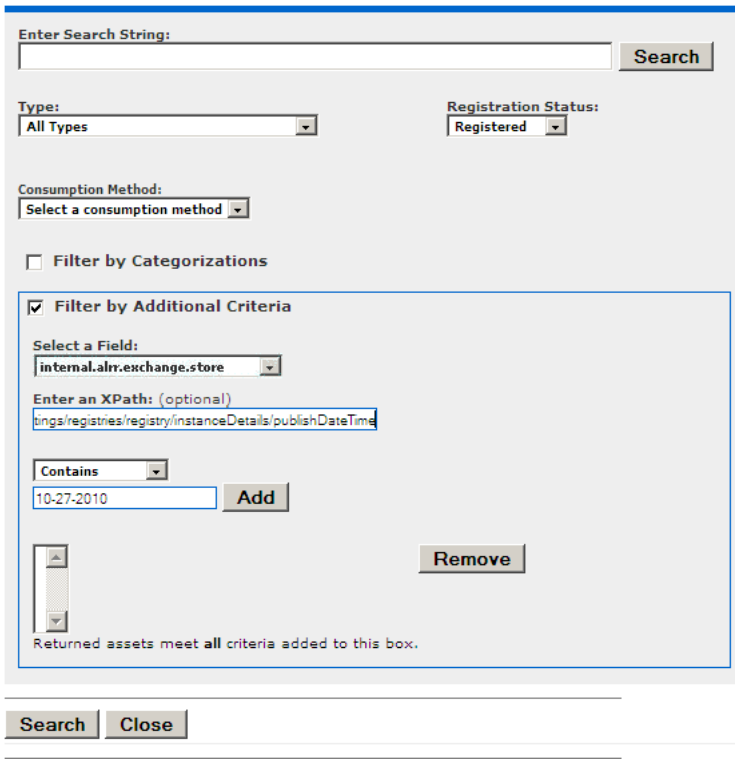
- Publish or Receive Date and Time
- Source and Destination Registries.

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 10–14.

Figure 10–14 More Search Options Dialog



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

   For more information about using the Oracle Enterprise Repository search options, see *Oracle Fusion Middleware User's Guide for Oracle Enterprise Repository*.

10.3.4 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 `<ExhangeUtility Tool Home>` directory. The logging options can be changed by updating the `log4fl.properties` file located in the `<ExhangeUtility Tool Home>` directory.

10.3.5 Known Issues

This section describes the known issues when using the Oracle Registry Repository Exchange Utility. It contains the following topics:

- Section 10.3.5.1, "Resynchronizing Oracle Service Registry Services"
- Section 10.3.5.2, "Publishing Services to Oracle Service Registry"
- Section 10.3.5.3, "Running an Incorrect Version of Java"

10.3.5.1 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.

10.3.5.2 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.

10.3.5.3 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)
```

11

Configuring Prescriptive Reuse

This chapter provides an overview of compliance templates, and their role in prescriptive reuse.

This chapter contains the following sections:

- Section 11.1, "Overview"
- Section 11.2, "Creating a Compliance Template Type"
- Section 11.3, "Creating a Compliance Template"
- Section 11.4, "Apply a Compliance Template to a Project"
- Section 11.5, "Using Assets Associated with Compliance Templates (Prescribed Assets)"
- Section 11.6, "Running a Project Compliance Report"

11.1 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 "Import Export Tool" in *Oracle Fusion Middleware User's Guide for Oracle Enterprise Repository*.

11.2 Creating a Compliance Template Type

This procedure is performed in the Type Manager. This example illustrates the creation of a Project Profile compliance template type.

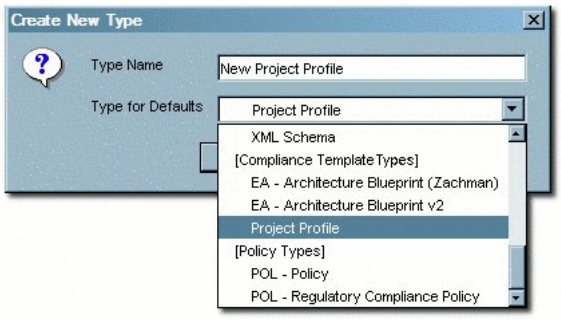
Caution: Adding a compliance template constitutes a permanent change to the Oracle Enterprise Repository system.

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 11–1.

Figure 11–1 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 11–2.

Figure 11–2 Type Manager

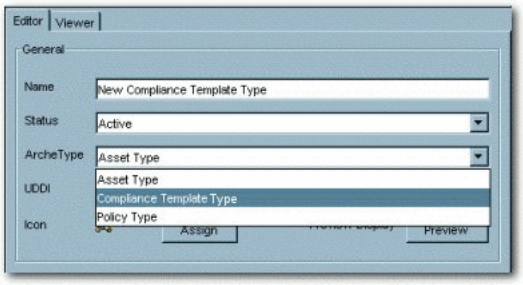
If no Compliance Template Type is available in the Type Manager:

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:

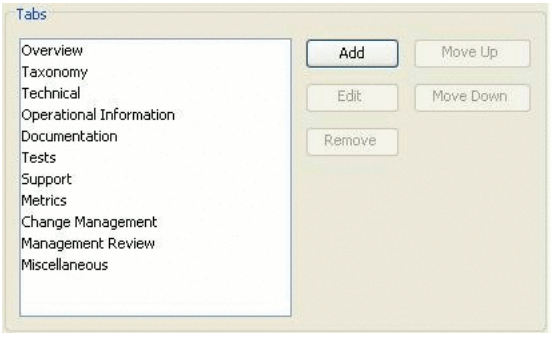
1. Select **New** in the File menu. The Create New Type dialog is displayed.
2. Enter the appropriate information in the Type Name text box.
3. In the Type for Defaults list, select any Type from the list.
4. Click **OK**. The new Type is displayed in the tree in the Type Manager sidebar.
5. Open the 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 11–3.

Figure 11–3 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 11–4.

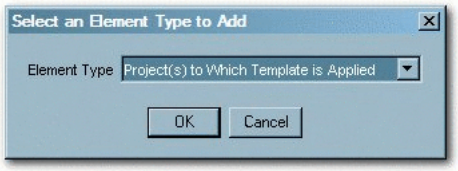
Figure 11–4 Tabs Section



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 11–5.

Figure 11–5 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 11–6.

Figure 11–6 Edit Project(s) to Which Template is Applied Dialog

14. Click **OK**. The new elements 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 11–7.

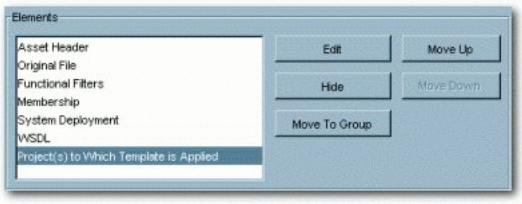
Figure 11–7 Hidden Elements Section

18. Click the **Display in Group** button. The Move Element dialog is displayed, as shown in Figure 11–8.

Figure 11–8 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 11–9 Elements Section



21. Click **Save** in the File menu.

11.3 Creating a Compliance Template

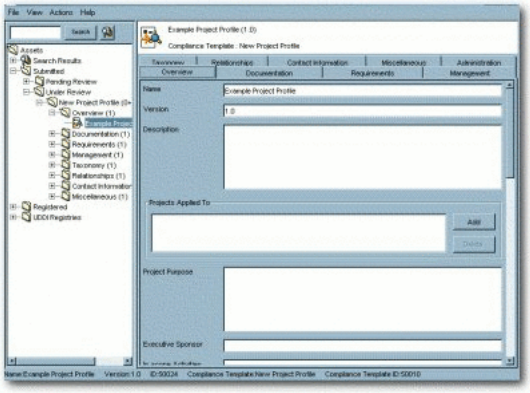
This procedure is performed in the Asset Editor, and is very similar to Advanced Asset Submission, as described in the "Advanced Asset Submission" section in *Oracle Fusion Middleware User's Guide for Oracle Enterprise Repository*.

1. Select **New** in the File menu. The Create a New Asset dialog is displayed, as shown in Figure 11–10.

Figure 11–10 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 11–11.

Figure 11–11 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, Prescibed 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.

For more information, see the Registration Process section in *Oracle Fusion Middleware User's Guide for Oracle Enterprise Repository*.

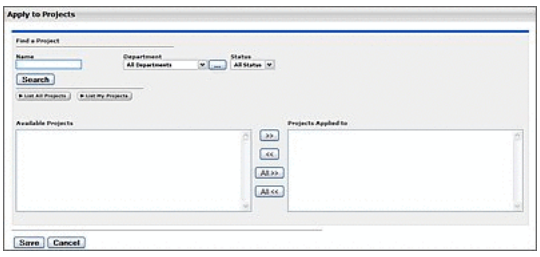
Follow this procedure to create Project Profiles, Architecture Blueprints, or other variations of the Compliance Template.

11.4 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 11–12.

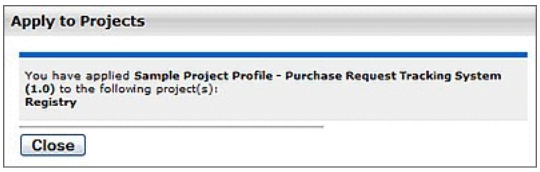
Figure 11–12 Apply to Projects Dialog



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 11–13.

Figure 11–13 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.

> **Note:** If the enterprise repository plugin for Eclipse or .Net is installed, the assets identified in the compliance templates will also appear in the project's development environment.

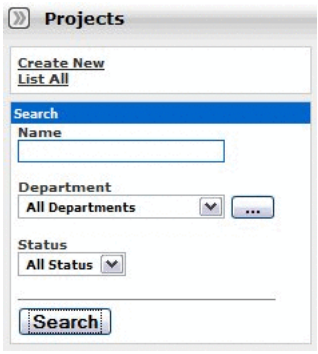
11.5 Using Assets Associated with Compliance Templates (Prescribed Assets)

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.

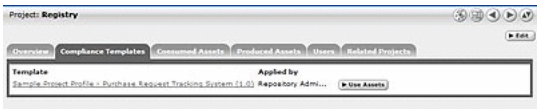
1. Use Search or other means, as shown in Figure 11–14, to locate the project.

Figure 11–14 Projects Screen



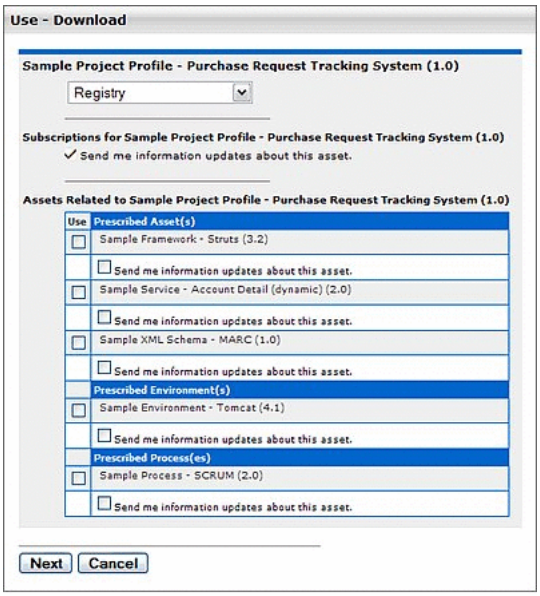
2. Any compliance templates applied to the project is listed in the Compliance Templates section of the project detail, as shown in Figure 11–15.

Figure 11–15 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 11–16.

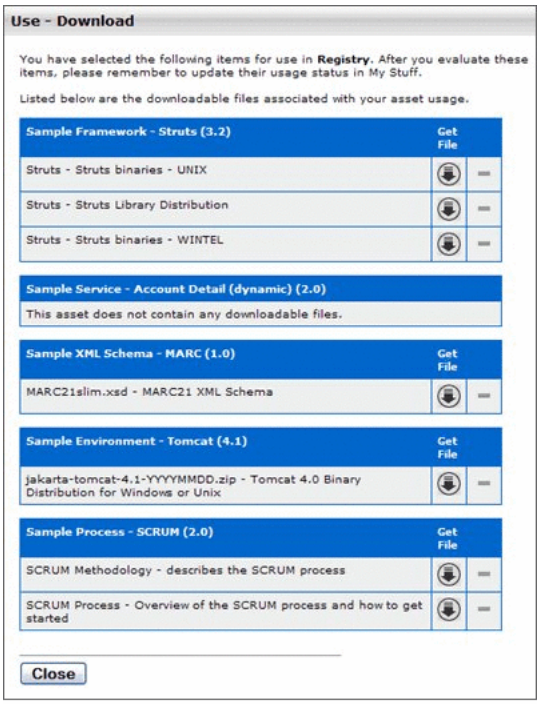
Figure 11–16 Use - Download Dialog



> **Note:** The Use Assets button will appear only to users who are assigned to the project.

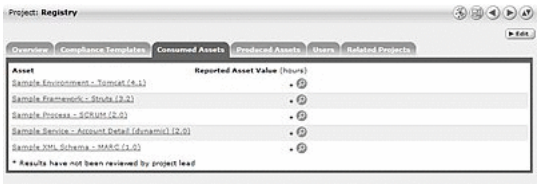
4. Use the options in the Use column to select the assets to be downloaded.
5. Click the **Next** button. The Use / Download dialog confirms the asset selections, as shown in Figure 11–17.

Figure 11–17 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 11–18.

Figure 11–18 Consumed Assets Tab



Prescribed assets can also be consumed from VS .Net and Eclipse environments. For more information about the process, see the VS .Net and Eclipse sections in *Oracle Fusion Middleware Integration Guide for Oracle Enterprise Repository*.

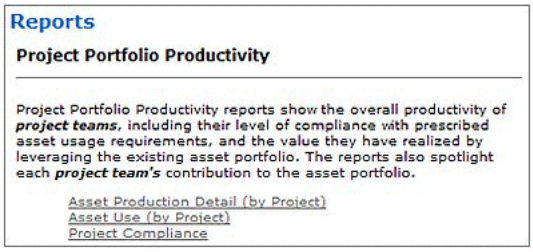
11.6 Running a Project Compliance Report

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.

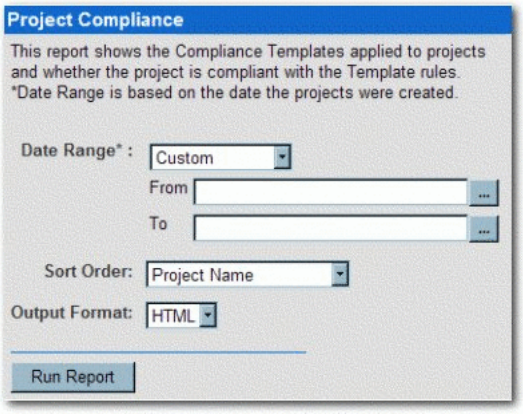
1. Click the **Project Compliance** link in the Reports sidebar, as shown in Figure 11–19.

Figure 11–19 Reports Section



The Project Compliance report form opens in the main pane, as shown in Figure 11–20.

Figure 11–20 Project Compliance Report Form

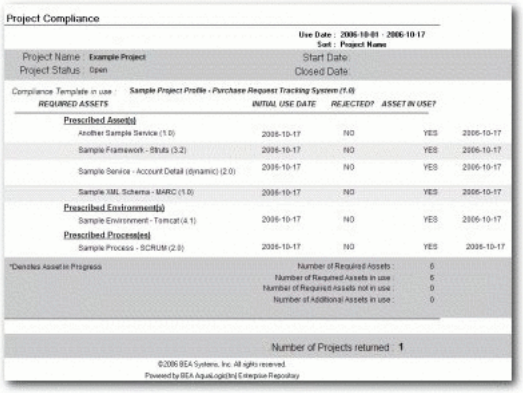


2. Use the Date Range, Sort Order, and Output Format lists to set the report parameters.

> **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 11–21.

Figure 11–21 Project Compliance Report



Project Compliance

Use Date : 2006-10-01 - 2006-10-17
Sort : Project Name

Project Name : Example Project
Project Status : Open

Start Date:
Closed Date:

Compliance Template in use : *Sample Project Profile - Purchase Request Tracking System (1.0)*

REQUIRED ASSETS	INITIAL USE DATE	REJECTED?	ASSET IN USE?	
Prescribed Asset(s)				
Another Sample Service (1.0)	2006-10-17	NO	YES	2006-10-17
Sample Framework - Struts (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 Environment(s)				
Sample Environment - Tomcat (4.1)	2006-10-17	NO	YES	2006-10-17
Prescribed Process(es)				
Sample Process - SCRUM (2.0)	2006-10-17	NO	YES	2006-10-17

*Denotes 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 AquaLogic(tm) Enterprise Repository

12

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 12.1, "Overview"
- Section 12.2, "Viewing Assets in Progress"
- Section 12.3, "Enabling Assets in Progress"
- Section 12.4, "Disabling Assets in Progress"
- Section 12.5, "Enabling Viewing of Assets in Progress Through an IDE"

12.1 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.

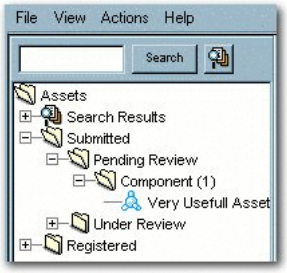
Assets in progress appear in Oracle Enterprise Repository as Unregistered assets. The actual Registration Status of any asset in progress (as indicated in the Status column in search results and in the asset detail) is determined during asset submission.

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 12–1 Oracle Enterprise Repository Assets Screen

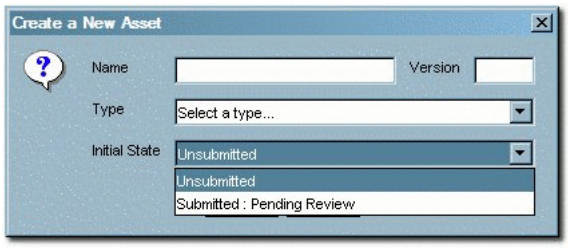
Figure 12–1 shows the Oracle Enterprise Repository Assets screen. Assets submitted via Submit an Asset enter the Submitted - Pending Review folder in the file tree in the Asset Editor, as shown in Figure 12–2.

Figure 12–2 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 12–3.

Figure 12–3 Create a New Asset



Assets in either state enter Oracle Enterprise Repository as Unregistered assets, and will appear in searches filtered for Unregistered assets.

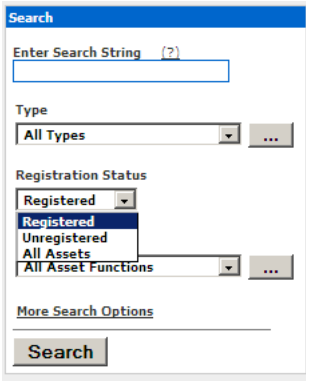
12.2 Viewing Assets in Progress

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 12–4.

Figure 12–4 Search Dialog



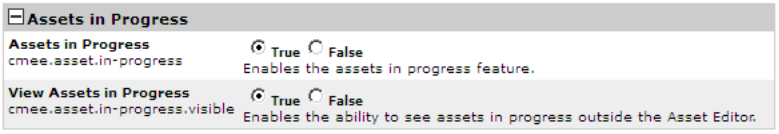
5. Click **Search**. A list of all Assets in Progress is displayed in the upper frame of the main pane.

12.3 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.

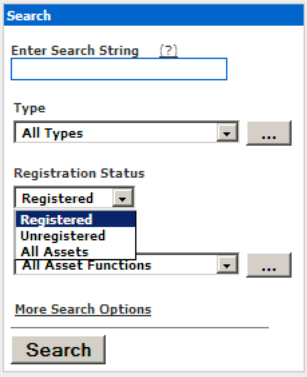
1. Click **System Settings** in the left pane.
2. Enter `Assets in Progress` in the Search field. The Assets in Progress setting appears in the Functional Settings pane, as shown in Figure 12–5.
3. Ensure the property is set to `True`.
4. Click **Save**.
5. Enter the `cmee.asset.in-progress.visible` 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 12–5.

Figure 12–5 Assets in Progress Properties



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 12–6.

Figure 12–6 Search Dialog



12.3.1 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**.

12.4 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. Locate **Assets in Progress** in the list of settings in the main pane.
3. Set the **Assets in Progress** property to `False`.
4. Locate **View Assets in Progress** in the list of settings in the main pane.
5. Set the **View Assets in Progress** property to `False`.
6. Click **Save**.

12.5 Enabling Viewing of Assets in Progress Through an IDE

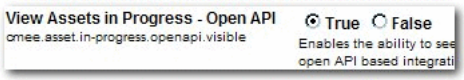
The Oracle Enterprise Repository Open API (part of the Oracle Enterprise Repository Extensibility Framework) makes it possible to enable viewing of both registered and unregistered assets in search results through an IDE.

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 12–7.

Figure 12–7 View Assets in Progress



4. Ensure the property is set to `True`.
5. Click **Save**.

13

Configuring Automated Usage Detection

This chapter describes Software File Identification(SFID) and how it can be used to automatically increment asset usage in Oracle Enterprise Repository.

This chapter contains the following sections:

- Section 13.1, "Overview"
- Section 13.2, "Enabling SFID"
- Section 13.3, "Adding the SFID Element to a Type"
- Section 13.4, "Adding an SFID to an Asset"
- Section 13.5, "SFID Tool Kit"
- Section 13.6, "SFID Manager"
- Section 13.7, "Calculating the SFID Programmatically"
- Section 13.8, "SFID Limitations"

13.1 Overview

There are several ways in which Oracle Enterprise Repository can track asset usage. Oracle Enterprise Repository strives to make Governance as seamless as possible for the developer, therefore automatic usage detection, or SFID is available from Eclipse and .Net environments. In these environments, developers receive a copy of a wsdl, schema, or other artifact. If an organization wants to automatically track the usage of these artifacts, they tag them with an SFID in Oracle Enterprise Repository. When the artifacts appear in the developer's environment, Oracle Enterprise Repository recognizes the artifact and increments usage. This means that a developer does not have to get the artifact from the repository. They can pull it from their desktop, or from a version control system. Regardless of the source, Oracle Enterprise Repository will recognize a match and increment usage.

Usage tracking is the primary method that Oracle Enterprise Repository uses to generate Reuse ROI.

Software File Identification (SFID) provides the ability to determine asset usage independent of the manual asset Use - Download process within Oracle Enterprise Repository. The SFID process tags selected files within an asset with a unique SFID fingerprint. This tag is then used to detect when and where an asset is used, even if the asset was acquired through means other than the Use - Download button in Oracle Enterprise Repository. An instance of usage is recorded by Oracle Enterprise Repository when tagged files within the asset are opened in a developer's IDE.

SFID requires the installation of the Oracle Enterprise Repository Plug-in Eclipse or for Visual Studio .NET. For more information about repository access capabilities from a supported IDE, see the following documentation:

- Repository Access in Eclipse Using WorkSpace Studio at *Oracle Fusion Middleware Integration Guide for Oracle Enterprise Repository*
- Oracle Enterprise Repository Integration with Visual Studio .NET at *Oracle Fusion Middleware Integration Guide for Oracle Enterprise Repository*

13.2 Enabling SFID

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

1. Click **System Settings** in the left pane.
2. Enter **cmee.usagedetection.enable** in the Enable New System Settings text box.
3. Click **Enable**. The Enable Automated Usage Detection option is displayed in the list of properties in the main pane, as shown in Figure 13–1.

Figure 13–1 Functional Settings

4. Ensure the property is set to `True`.
5. Click **Save**. SFID is now enabled.
   - The SFID element can now be added to types in the Type Manager.
   - When the SFID element is added to an asset type, an SFID section will appear on assets of that type when viewed in the Asset Editor. It is within that section that files can be selected for SFID fingerprinting.

Figure 13–2 Automated Usage Detection

   - SFID functions are now available in the Oracle Enterprise Repository REX.

> **Note:** For more information about REX, see "Repository Extensibility Framework" in Oracle Fusion Middleware Integration Guide for Oracle Enterprise Repository.

13.3 Adding the SFID Element to a Type

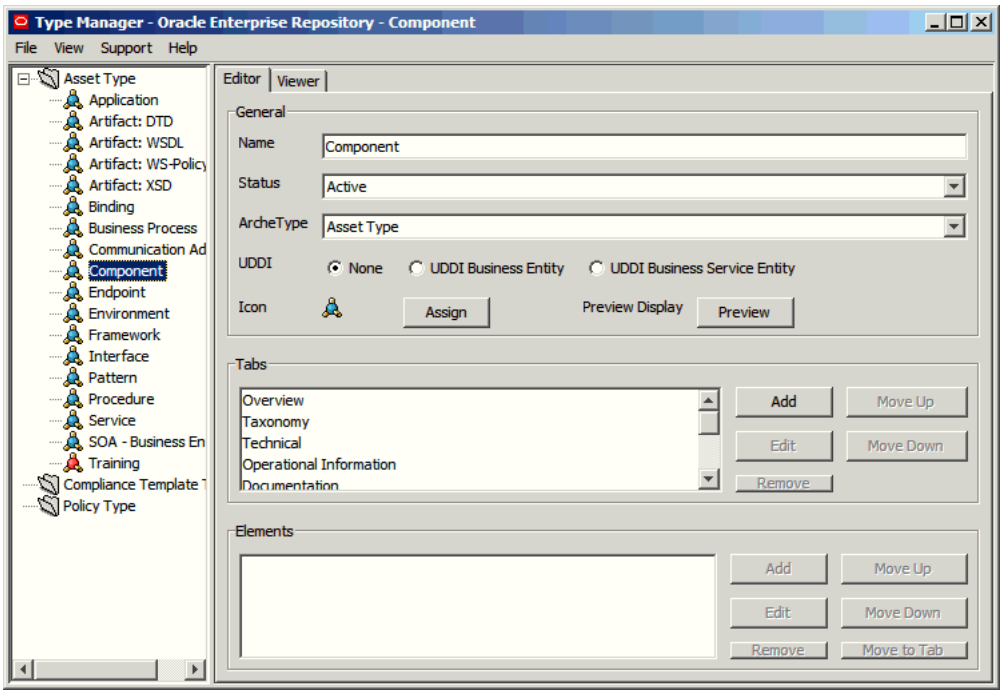
Adding the SFID element to a type enables assets of that type to be automatically detected.

Note: You will also need to add an SFID to the individual assets to be tracked.

This procedure is performed in the Type Manager.

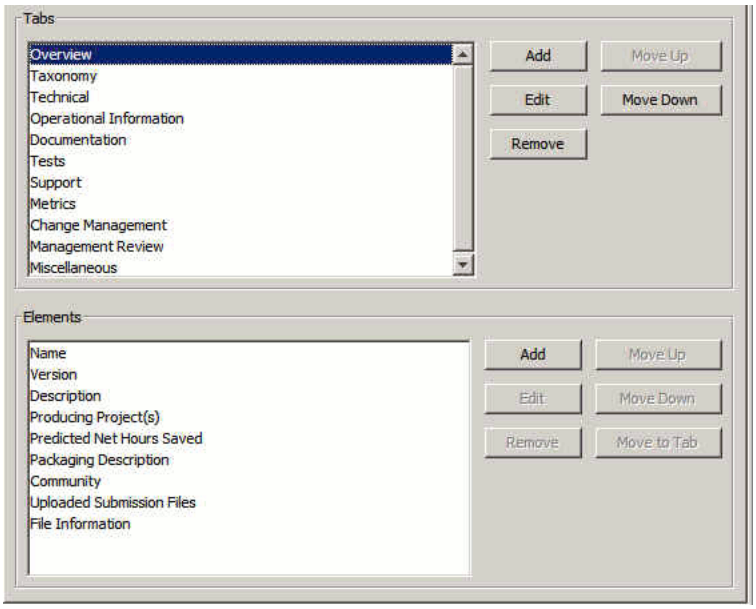
1. Select the type to which the SFID element is to be added, as shown in Figure 13–3.

Figure 13–3 Type Manager Window



2. By default, the Editor tab is displayed. If not, click the **Editor** tab.
3. From the list in the Tabs section, select the tab to which to add the SFID element, as shown in Figure 13–4.

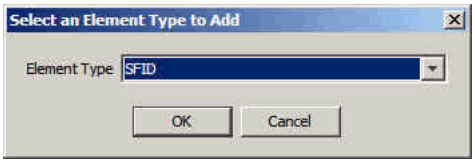
Figure 13–4 Tabs List



A list of the elements on that tab appears in the Elements section.

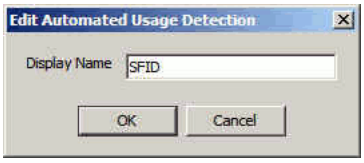
4. Click the **Add** button in the Elements section. The Select an Element Type to Add dialog is displayed.
5. Click **SFID** from the Element Type list, as shown in Figure 13–5.

Figure 13–5 Select an Element Type to Add Dialog



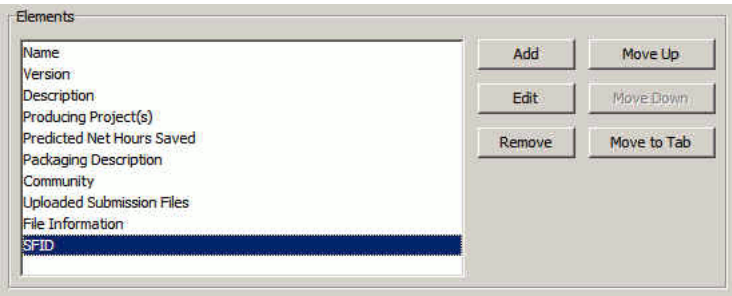
6. Click **OK**. The Select an Element Type to Add dialog closes, and the Edit Automated Usage Detection dialog is displayed, as shown in Figure 13–6.
7. Enter a name for the SFID element.

Figure 13–6 Edit Automated Usage Detection Dialog



8. Click **OK**. The new element is listed in the Elements section, as shown in Figure 13–7.

Figure 13–7 Elements Section



9. Click **File**, **Save** to save the modified asset type.
10. Exit the Type Manager.
11. In the Asset Editor, open an asset of the type to which the SFID element was added.

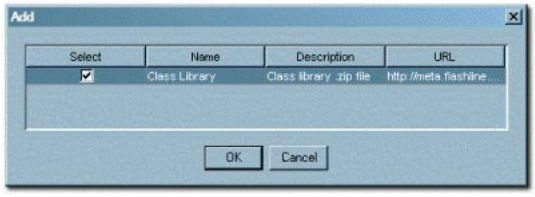
    The SFID section now appears on the tab to which it was assigned.

13.4 Adding an SFID to an Asset

To track assets through an IDE, select the unique artifact file representing the asset. The file is assigned an SFID, and usage is automatically incremented every time the file is found in an IDE project. This procedure is performed in the Asset Editor.

1. Select the asset to which an SFID is to be added.
2. Click the **Add** button in the Automated Usage Detection section.
3. Select the files to be fingerprinted from the list in the Add dialog, as shown in Figure 13–8.

Figure 13–8 Add Dialog



4. Click **OK**.

   The system downloads the selected files, calculates the SFID, and then deletes the local (downloaded) copies of the files. A warning message is displayed if fingerprinting for any of the selected files is unsuccessful.

13.4.1 Update an SFID File

This procedure is performed in the Asset Editor.

1. Select the SFID file to be updated from the list in the Automated Usage Detection section.
2. Click the **Update** button. The Perform Operation window opens, indicating the progress of the update.

13.4.2 Remove an SFID File

This procedure is performed in the Asset Editor.

1. Select the SFID file to be removed from the list in the Automated Usage Detection section.
2. Click the **Remove** button. The file is removed from the list.

13.5 SFID Tool Kit

The SFID Toolkit provides a framework that allows third party developers to develop applications that integrate with Oracle Enterprise Repository's REX subsystem for the purpose of submitting SFID Candidates. The SFID toolkit provides an application that will create an SFID Candidate for each file in a specified directory and then submit those SFID Candidates to the Oracle Enterprise Repository system for storage and analysis.

> **Note:** For more information about REX subsystems, see "Subsystems Overview" in *Oracle Fusion Middleware Integration Guide for Oracle Enterprise Repository*.

Notes on the Calculation of SFIDs

- A file is considered a text file if it contains no characters above ASCII code 128.
- The content of a text file is normalized before the SFID hash is computed.
  - All CR, LF, and Tab chars are stripped.
  - All text is converted to lower case.
- XML Files are canonicalized as per the W3C canonicalization standard (*http://www.w3.org/TR/xml-c14n*) before the SFID hash is computed.

13.5.1 Installation and Configuration

This section describes the steps to install and configure the SFID toolkit. This section contains the following topics:

- Section 13.5.1.1, "Enabling Harvesting of Unrecognized SFIDs"
- Section 13.5.1.2, "Object Model"
- Section 13.5.1.3, "Description of Sample Directory Crawler"

13.5.1.1 Enabling Harvesting of Unrecognized SFIDs

Some organizations mandate that every SOA asset deployed should be in the Oracle Enterprise Repository. Companies can monitor compliance to this mandate by harvesting unrecognized SFIDs. For example, companies may want to monitor the build scripts for unrecognized SFIDs. Additionally, Service monitoring tools may discover services that are not in Oracle Enterprise Repository. These "rogue services" can be assigned an SFID and harvested into Oracle Enterprise Repository as asset candidates. Once in Oracle Enterprise Repository, they can be further evaluated and processed.

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

1. Click **System Settings** in the left pane. The System Settings page is displayed in the main pane.

2. Enter `cmee.sfid.harvesting.enable` in the Enable New System Setting text box.
3. Click the **Enable** button. The Enable Harvesting of Unrecognized SFIDs property appears in the list of system settings.

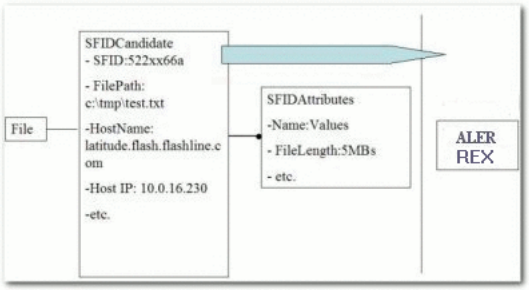
Figure 13–9 Enable Harvesting of Unrecognized SFIDs

4. Ensure property is set to `True`.
5. Click **Save**. The property is now enabled.

13.5.1.2 Object Model

Figure 13–10 describes the object model for the SFID.

Figure 13–10 Object Model



13.5.1.3 Description of Sample Directory Crawler

- Iterate through all the files and folders of a specified folder.
  - Instantiate an SFIDCandidate for each candidate
  - Populate general metadata such as path and sfid.

    For each SFIDCandiate, create SFIDAttributes that describe arbitrary data to be captured, such as Eclipse Workspace.
- Invoke the Oracle Enterprise Repository REX method sfidCandidateSubmit, passing an array of SFIDCandidate objects.
- AlerUtil Object
  - Instantiate a AlerUtil object to connect to Oracle Enterprise Repository.

```
String lUrl = "http://localhost:9080/oer/services/FlashlineRegistry";
String lUserName = "username";
String lPassword = "password";
lAlerUtil = new AlerUtil(lUrl, lUserName, lPassword);
```

  - Use the AlerUtil object to invoke a Oracle Enterprise Repository REX method.

```
lAlerUtil.getFlashlineRegistry().sfidCandidateSubmit(lAlerUtil.getAuthToken
(), lSfidCandidates);
```

  - The AlerUtil object provides a series of convenience methods.

* getAuthToken

  Obtain a valid authToken for the for current username and password. If the authToken is expired, obtain a new authToken. Otherwise, a UserValidationException is thrown.

* calculateSfidValue

  Calculate the SFID for a file or path and return the hash as a string.

* sfidCandiateListToArray

  Convert a list of SFIDCandidates to an array of SFIDCandidates.

* sfidAttributeListToArray

  Convert a list of SFIDAttributes to an array of SFIDAttributes.

* sfidAttributeValueListToArray

  Convert a list of SFIDAttributeValues to an array of SFIDAttributeValues.

* createCandidate(File)

  Create a standard SFIDCandidate for a file. SFIDCandidate fields filepath, hostip, hostname, and sfid is pre-populated. An SFIDAttribute named Filelength is attached to the SFIDCandidate with the value of length of the file.

* createCandidate(File, Map)

  Create a standard SFIDCandidate for a file as specified by the method createCandidate(File). Each (String)key/(String)value in the map will attach an additional SFIDAttribute to the SFIDCandidate whose name is key and whose value is value.

* createCandidate(File, List)

  Create a standard SFIDCandidate for a file as specified by the method createCandidate(File). Attach each SFIDAttribute passed in the list to the SFIDCandidate.

* sendCandidate

  Send a single SFIDCandidate for harvesting.

* sendCandidates

  Send a list of SFIDCandidates for harvesting.

13.6 SFID Manager

The SFID Manager provides an easy way to view SFID candidates harvested by Oracle Enterprise Repository.

Use of the SFID Manager requires System Administrator privileges.

This section contains the following topics:

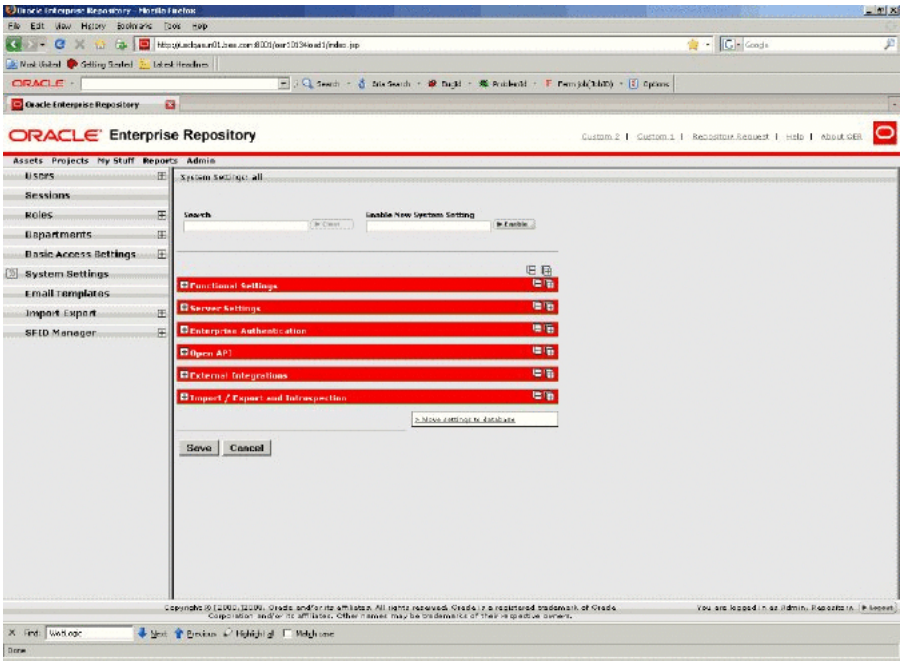
- Section 13.6.1, "Enabling the SFID Manager"
- Section 13.6.2, "Using the SFID Manager"

13.6.1 Enabling the SFID Manager

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

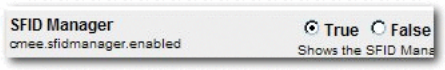
1. Click **System Settings** in the left pane. The System Settings page is displayed in the main pane, as shown in Figure 13–11.

Figure 13–11 Oracle Enterprise Repository System Settings Page



2. Enter `cmee.sfid.harvesting.enable` in the Enable New System Setting text box.
3. Click the **Enable** button. SFID Manager is displayed in the list of system settings, as shown in Figure 13–12.

Figure 13–12 SFID Manager



4. Ensure the True option is selected.
5. Click **Save**.
6. Click **Admin** in the Oracle Enterprise Repository menu bar to refresh the Admin screen.

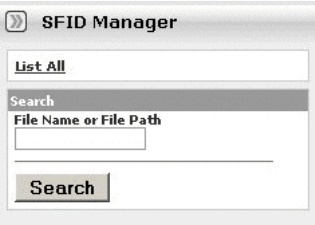
   The SFID Manager section now appears in the left pane.

13.6.2 Using the SFID Manager

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

1. Click **SFID Manager** in the left pane.
2. Click **List All** in the SFID Manager section, as shown in Figure 13–13, to display a list of available SFID candidates.

Figure 13–13 SFID Manager



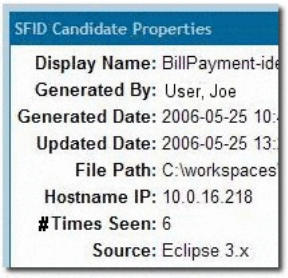
The number of available candidates affects the time it takes for this operation to complete. SFID candidates can also be located using the SFID Manager search function. Identified candidates are listed in the upper section of the main pane, as shown in Figure 13–14.

Figure 13–14 Name - SFID List

The number of times the listed SFID has been detected (Times Seen)is indicated in the # column.

3. Click any listed candidate to display extended information in the bottom section of the main pane, as shown in Figure 13–15.

Figure 13–15 SFID Candidate Properties



The displayed information consists of a core set of attributes common to all candidates:

- Display Name

  The name of the candidate. Generally, the simple file name.

- Generated By

  The username of the user that caused the candidate to be generated.

- Generated Date

  The date on which the candidate was created.

- Updated Date

  The last time that the candidate record was modified in the database.

- File Path

The absolute path to the file on the machine from which the candidate was harvested.

- Hostname/IP

  The hostname or IP address of the machine from which the candidate was harvested.

- # Times Seen

  The number of times this candidate has been viewed in this context. This metric gives an indication of the frequency of file use, which provides some indication of the value of the file in this context.

- Source

  A description of the context, for example, Eclipse, Ant.

The display also includes an extended set of attributes that will vary based on the context from which the candidate was harvested.

13.7 Calculating the SFID Programmatically

The `ArtifactFingerprinter` java library provides functionality for calculating SFID fingerprints programmatically in the client-side code.

Installation

Ensure that you have the `components-flashtrax.jar` file and it is in your java classpath for installation.

Usage

The com.oracle.artifact.ArtifactAlgorithm class contains several static instances, each of which encapsulates the fingerprinting algorithm for a particular type of file.

The supported algorithms are

- **DEFAULT**: fingerprints based on the file contents, without any transformations
- **XML**: normalizes the XML contents to remove whitespace, and normalize namespace abbreviations
- **XSD**: same as XML, plus inlines XSD imports and includes
- **WSDL**: same as XSD, plus inlines WSDL imports and includes, removes documentation, and orders WSDL elements by name
- **WSDL_ABSTRACTING**: same as WSDL, plus removes endpoint and binding information

Example

An example that describes calculating the SFID programmatically is as follows:

ArtifactFingerprinter fp = ArtifactAlgorithm.WSDL.newFingerprinter();

String sfid = fp.getFingerprint(new File("mytest.wsdl"));

13.8 SFID Limitations

SFID System Limitations:

- The file must be accessible to generate the SFID.

The client machine/user must be provided with a path (URI) to the file. The URI must be accessible from the client system, without the need for user intervention, such as requests for authentication information).

- SFIDs are associated with files (1:1) which are associated with assets (n:n)
- SFIDs can be added to files through the Asset Editor or through REX.
- SFID is not supported on Oracle JDeveloper. It is supported only for VS .Net and Eclipse.

Limitations of the Asset Editor (SFID generation)

- An SFID can be associated only with extractable files.
- An SFID can be generated only for files that are accessible to the client.

Limitations of the ANT Integration (SFID comparison)

- SFIDs can be generated for ALL accessible files (for comparison).

14

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 14.1, "Overview"
- Section 14.2, "Activating the Policy Management Feature"
- Section 14.3, "Creating a Policy Type"
- Section 14.4, "Adding the Policy Assertion Element to the Policy Type"
- Section 14.5, "Creating a Policy"
- Section 14.6, "Adding the Applied Asset Policies Element to a Type"
- Section 14.7, "Applying a Policy to an Asset"
- Section 14.8, "Viewing Policy Status"
- Section 14.9, "Validating Policy Assertions"
- Section 14.10, "Running the Oracle Enterprise Repository Policy Reports"

14.1 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 "Import Export Tool" in *Oracle Fusion Middleware User's Guide for Oracle Enterprise Repository*.

14.2 Activating the Policy Management Feature

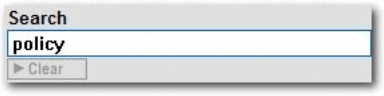
To activate the policy management feature, perform the following steps:

Step 1: Enable the Policy Management System Settings

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

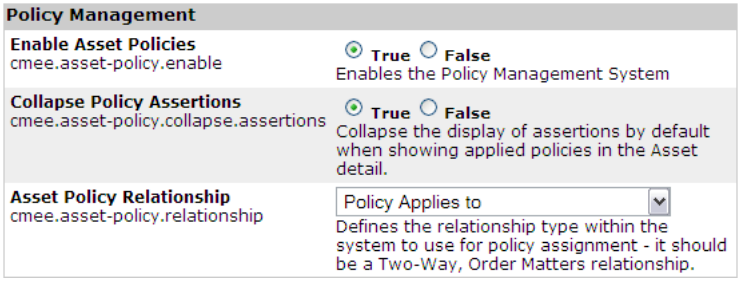
1. Click **System Settings** in the left pane.
2. Enter policy in the System Settings Search text box, as shown in Figure 14–1.

Figure 14–1 System Setting Search



The Policy Management group is displayed, as shown in Figure 14–2.

Figure 14–2 Policy Management



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>/repositoryXXX/core/tools/solutions, and then using the Import/Export Tool to import them into your environment.

14.3 Creating a Policy Type

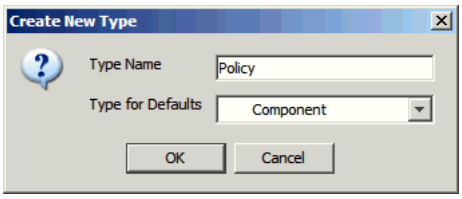
If you prefer to create your own policy instead of using the policy template in the Base Data, you can do so, which is explained in this section. A policy type template can be created using any existing asset type or compliance template type as the basis. A Policy Type is included in the Policy Management Solution Pack.

1. Click **Edit/Manage Assets** in the Oracle Enterprise Repository Assets screen to launch the Asset Editor.

Figure 14–3 Assets Section

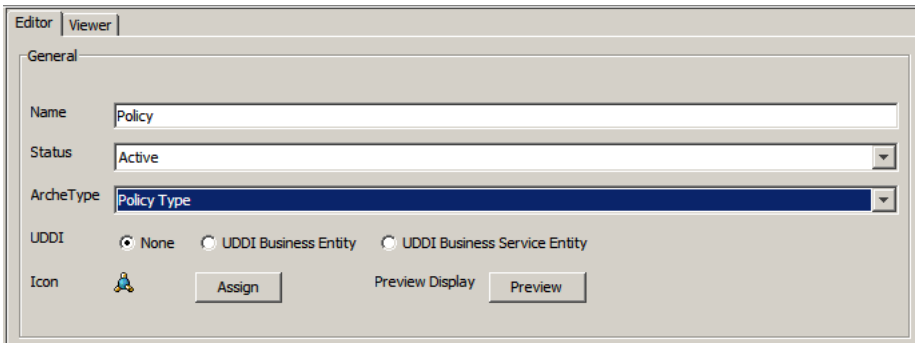
2. Open the Actions menu in the Asset Editor.
3. Click **Manage Types** to launch the Type Manager.
4. Open the File menu and click **New**. The Create New Type dialog is displayed.
5. Enter `Policy` in the Type Name field and select a template from the Type for Defaults list, as shown in Figure 14–4.

Figure 14–4 Create New Type Dialog



6. Click **OK**.
7. Select **Policy Type** from the Archetype list in the Editor tab in the Type Manager.

Figure 14–5 Type Manager



You have successfully created your own policy type.

14.4 Adding the Policy Assertion Element to the Policy Type

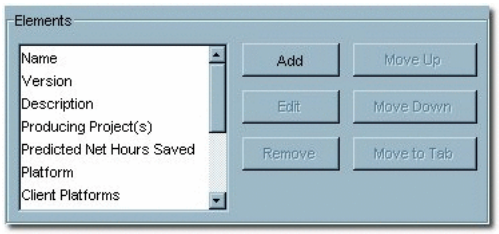
A policy assertion states that conditions must be true for an asset. For example, a policy assertion might state that the asset should have a "defect density rate of less that 1%". Each assertion is validated individually. If the asset satisfies all assertions (meaning that all assertions have a "pass" 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 14–6.

Figure 14–6 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 14–7.

Figure 14–7 Select an Element Type to Add

4. Click **OK**. The Edit Policy Assertions dialog is displayed, as shown in Figure 14–8.
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 14–8 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.

14.5 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.

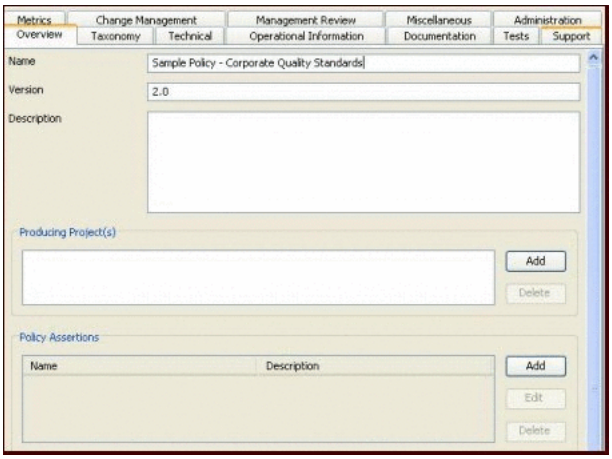
This procedure is performed in the Asset Editor.

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 14–9.

Figure 14–9 Create a New Asset Dialog

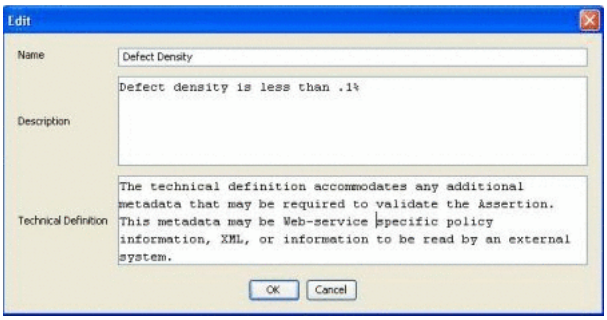
3. Enter a version number (optional).
4. Select the appropriate policy type from the Type pull-down.
5. Select the appropriate Initial State from that pull-down.
6. Click **OK**. The policy opens in the Asset Editor.

Figure 14–10 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.

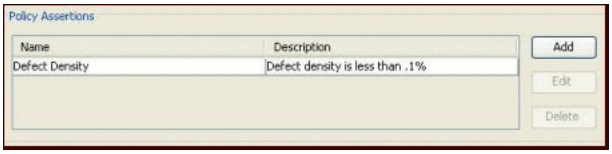
Figure 14–11 Edit Dialog



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 14–12 Policy Assertions



10. Repeat the procedure to add additional assertions to the policy.

14.6 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 perfomed 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.

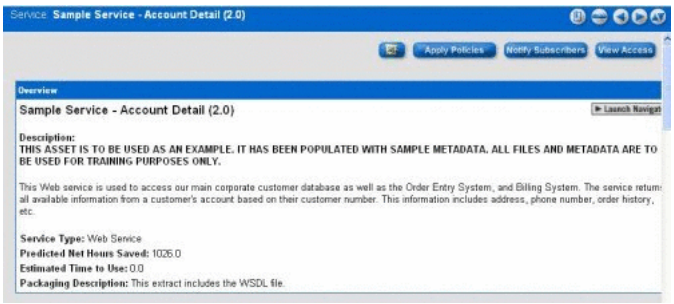
14.7 Applying a Policy to an Asset

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.

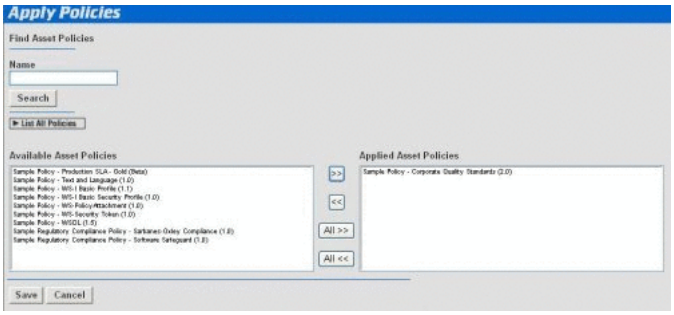
1. Using Oracle Enterprise Repository's Search or other means, locate the asset to which the policy is to be added.
2. Click the **Apply Policies** button in the asset detail display, as shown in Figure 14–13.

Figure 14–13 Overview



3. Click **List All Policies** to display a list in the Available Asset Policies box, as shown in Figure 14–14.

Figure 14–14 Apply Policies



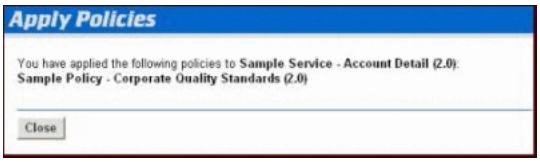
4. Use the >> or All >> buttons to move selected policies to the Applied Asset Policies box.

   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 14–15.

Figure 14–15 Apply Policies

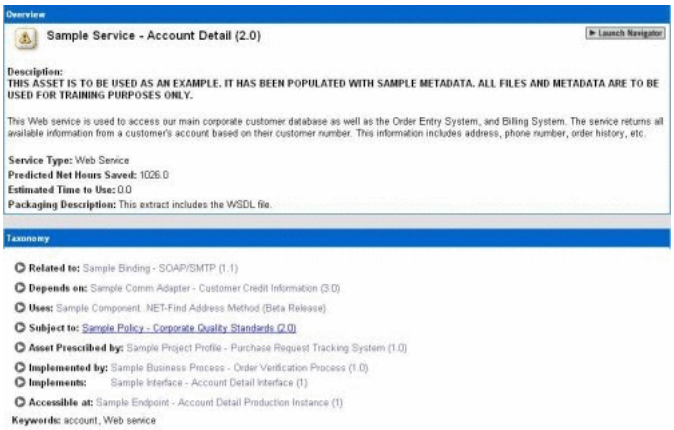


7. Click **Close**.

8. Refresh the asset detail frame.

9. The presence of the icon in the Overview section of the asset detail, as shown in Figure 14–16, indicates that a policy has been applied. (A smaller version of the same icon indicates policy assignment for assets listed in search results.)

Figure 14–16 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 14–17. Click the text link for any listed policy to display its details.

Figure 14–17 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.

14.8 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 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
	Status: Applied Indicates that one or more policies have been applied to the asset, and that pass/fail status is indeterminate. Any combination of passed and indeterminate assertion statements results in indeterminate asset status.
	Status: Passed Indicates that the asset is in compliance with every assertion statement within every applied policy.
	Status: Failed Indicates that the asset is not in compliance with all assertion statements within all applied policies. Failure to comply with any assertion statement within any applied policy results in Failed status.

14.9 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.

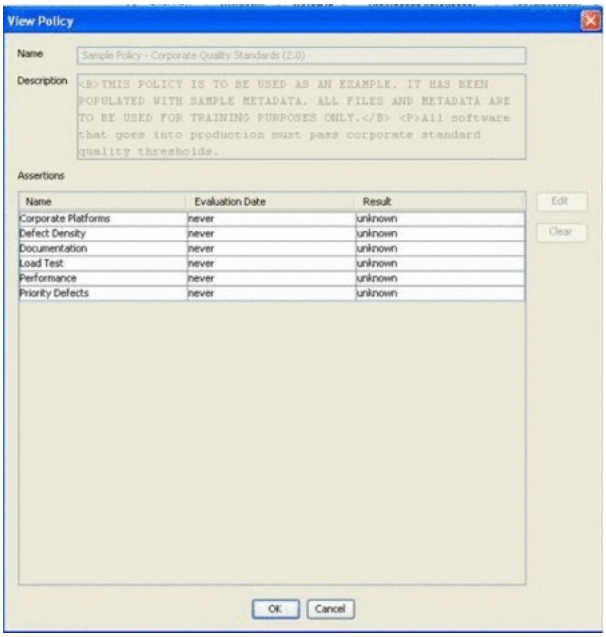
1. Locate the asset to be updated.
2. Locate the Applied Asset Policies section within the asset.

   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 14–18.

Figure 14–18 Applied Asset Policies

4. Click **Details**. The View Policy dialog is displayed, as shown in Figure 14–19.

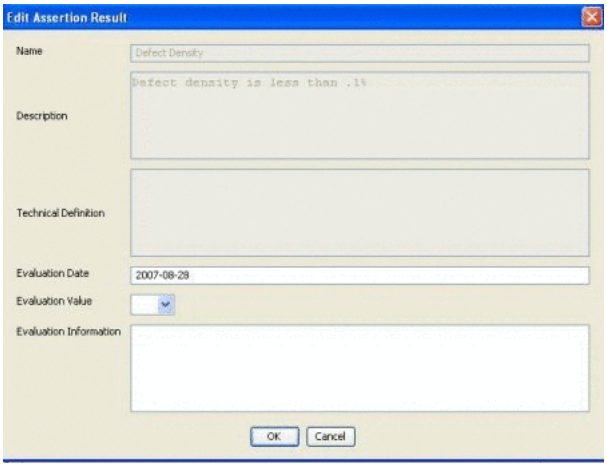
Figure 14–19 View Policy Dialog



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.

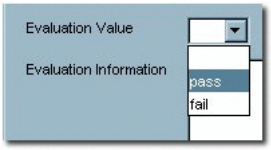
5. Select the assertion statement to be updated.
6. Click **Edit**. The Edit Assertion Result dialog is displayed, as shown in Figure 14–20.

Figure 14–20 Edit Assertion Result Dialog



7. Click the Evaluation Value list and select the appropriate status, as shown in Figure 14–21.

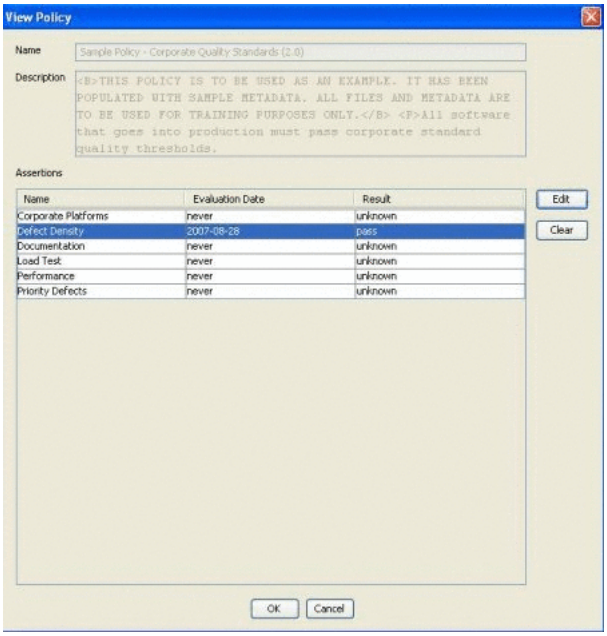
Figure 14–21 Evaluation Value List



In this example the pass status is selected. Note that additional details can be provided in Evaluation Information.

8. Click **OK**. The updated status is displayed in the Result column in the View Policy dialog, as shown in Figure 14–22.

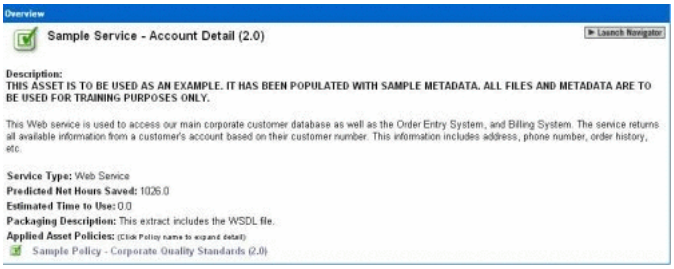
Figure 14–22 View Policy Dialog



Note: To change an assertion value back to unknown, select the assertion and click **Clear**.

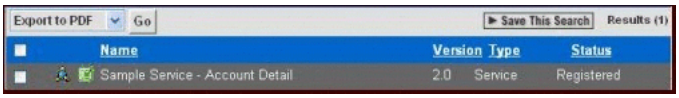
9. Continue to validate remaining assertions in the same manner.
10. When finished, click **OK** to close the View Policy dialog.
11. Save the changes to the asset.
12. An icon with a green tick mark appears in the asset detail, as shown in Figure 14–23.

Figure 14–23 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 14–24.

Figure 14–24 Search Results



14.10 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.

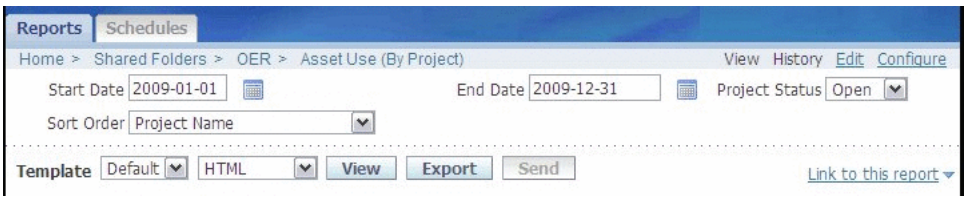
Policy Status Report

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:

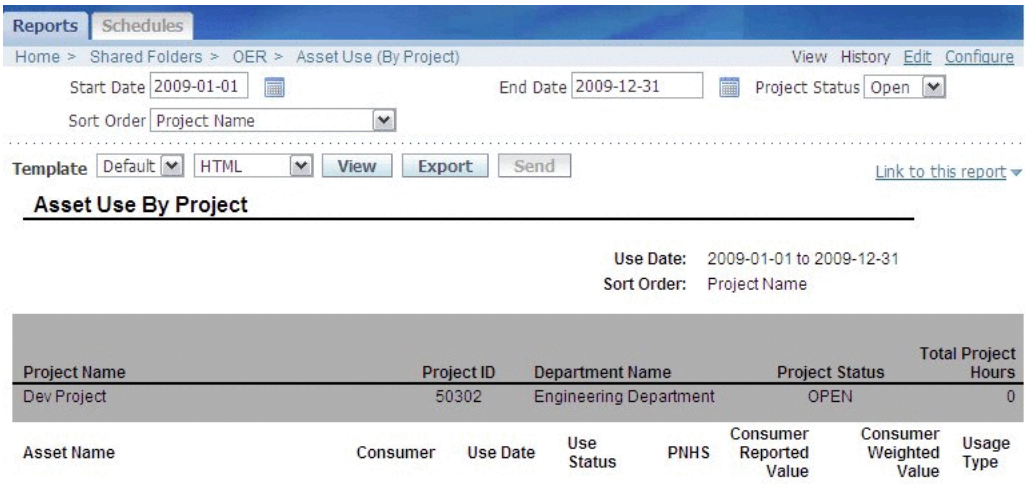
1. In Oracle Enterprise Repository Reports page, click a report name. The BI Publisher opens the Reports tab in a new browser window, as shown in Figure 14–25.

Figure 14–25 Reports Tab



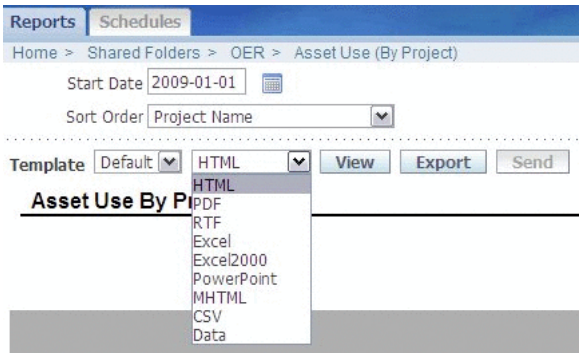
2. Click the **View** button. The details of the selected report are displayed, as shown in Figure 14–26.

Figure 14–26 Reports Tab - Detailed View



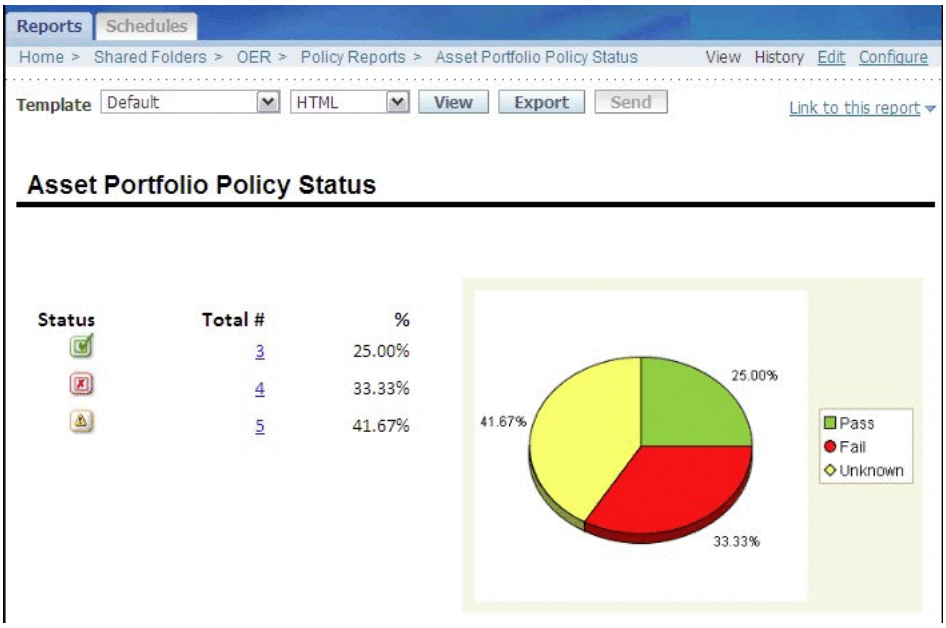
3. You can select different formats, as shown in Figure 14–27, from the second list that appears next to **Template**.

Figure 14–27 List of Formats



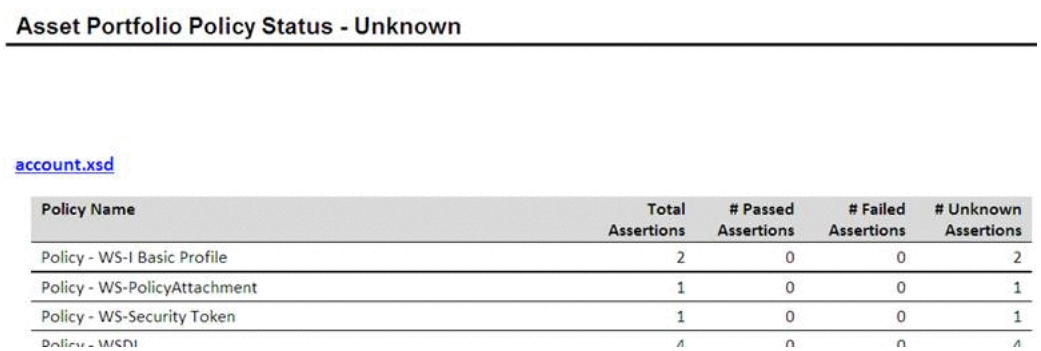
4. In the Asset Portfolio Management section, click the Asset Portfolio Policy Status option. The Asset Portfolio Policy Status page is displayed, as shown in Figure 14–28.

Figure 14–28 Asset Portfolio Policy Status Page



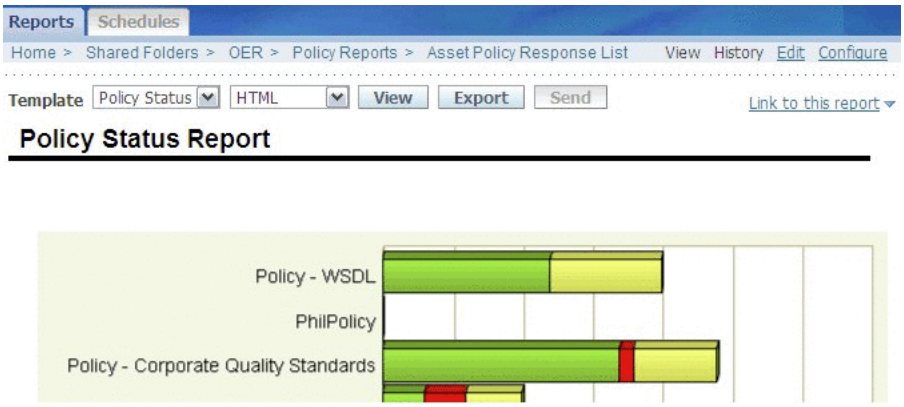
5. To view the detail for a set of assets, click the number next to the icon for the appropriate Policy Status. The Asset Portfolio Policy Status - Status detail report is displayed, as shown in Figure 14–29.

Figure 14–29 Asset Portfolio Policy Status Report Page



6. In the Asset Portfolio Management section, click the Policy Status option. The Policy Status Report page is displayed, as shown in Figure 14–30.

Figure 14–30 Policy Status Report Page



15

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 15.1, "Overview"
- Section 15.2, "Import the Contract Management Solution Pack"
- Section 15.3, "Instructions for Service Providers"
- Section 15.4, "Instructions for Service Consumers"

15.1 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.

15.1.1 Prerequisites

- Harvester Solution Pack (11.1.1.x.x-OER-Harvester-Solution-Pack.zip)
- Policy Solution Pack (Policy-Management-Solution-Pack.zip)

15.2 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 "Import Export Tool" in *Oracle Fusion Middleware User's Guide for Oracle Enterprise Repository*.

15.3 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.

15.3.1 Create the Contract

Use the Copy/Migrate option from the Oracle Enterprise Repository Asset Editor to create a new contract based on this sample.

1. Select the contract and click Edit. The Asset Editor displays.
2. Select File>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.

15.3.2 Edit the Contract

Enter provider information on the Overview tab, as well as additonal 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 so on.
2. Change the placeholder text indicated by brackets [] in the Description field.
3. Save your changes.

15.3.3 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, 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.

15.3.3.1 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).
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.

15.4 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.

This procedure is performed in the Asset Editor.

1. Select the contract and click Edit. The Asset Editor displays.
2. Select File>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.

Submit your contract: The contract is submitted to the provider for approval and registration.

- From the Administration tab, click Submit.

You will receive an email notification when the provider grants access to the service.

16

Configuring Reporting with BI Publisher

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

This chapter contains the following sections:

- Section 16.1, "Deploy Oracle Enterprise Repository Pre-configured BI Publisher Content"
- Section 16.2, "Security Model"
- Section 16.3, "Configuring the Oracle Enterprise Repository JDBC Data Source"
- Section 16.4, "Configuring the BI Publisher Server - Cache Expiration"
- Section 16.5, "Configuring the Oracle Enterprise Repository System Settings to Enable Reporting"

16.1 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.

16.1.1 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>/repositoryXXX/core/tools/solutions/reports* directory.

   In this directory you will find **11.1.1.7.0-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.

16.2 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* .

16.2.1 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.

16.2.2 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.

16.2.3 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.

16.3 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. Use the same ConnectString as what appears in the Oracle Enterprise Repository **database.properties** file. Use the same database authentication as Oracle Enterprise Repository.

   In a WebLogic domain, the **database.properties** file is found in <domain>/applications/oer_11.1.1.x.0/oer-app/WEB-INF/classes.

   These sections describe the database configuration for the following databases:

   - "Oracle"
   - "SQL Server"
   - "UDB"

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

Oracle

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 11*g*
- Oracle 10*g*
- Oracle 9*i*

Example for Oracle JDBC Connection:

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

SQL Server

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

- Microsoft SQL Server 2005
- Microsoft SQL Server 2000

Example for SQL Server JDBC Connection:

Database Driver Class	hyperion.jdbc.sqlserver.SQLServerDriver
Connect String	jdbc:hyperion:sqlserver://[HOST]:[PORT];DatabaseName=[DB_NAME]

Example	jdbc:hyperion:sqlserver://machine.example.com;databaseName=oer

UDB

You can select either of the following Driver types for IBM DB2 Universal Database depending on your DB2 database version. The JDBC library file (JAR) is installed as part of the BI Publisher Server installation therefore no extra Jar file is required.

- IBM DB2 Universal Database v9
- IBM DB2 Universal Database v8

Example for UDB JDBC Connection:

Database Driver Class	hyperion.jdbc.db2.DB2Driver
Connect String	jdbc:hyperion:db2://[hostHOST]:[portPORT];DatabaseName=[dbnameDB_NAME]
Example	jdbc:hyperion:db2://machine.example.com;DatabaseName=oer

16.4 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.

16.5 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 1.6, "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 16.2.2, "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.
11. 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.

17

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:

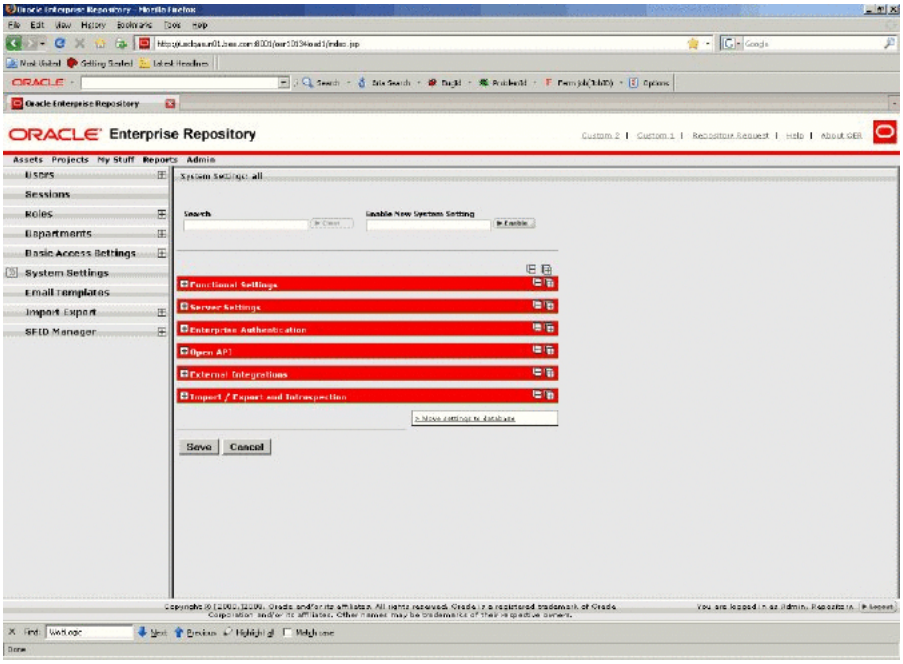
- Section 17.1, "Access System Settings"
- Section 17.2, "Search for System Settings"
- Section 17.3, "Enable System Settings Properties"
- Section 17.4, "Move Settings to Database"
- Section 17.5, "System Settings in Oracle Enterprise Repository"
- Section 17.6, "Known Issues"

17.1 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.

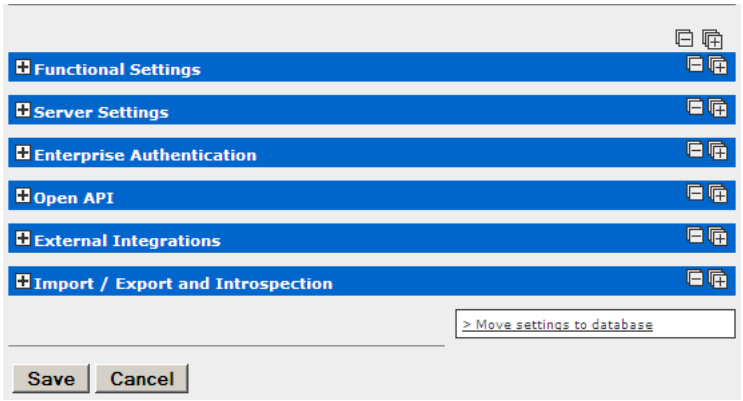
Figure 17–1 Oracle Enterprise Repository System Settings Page



The System Settings section opens in the main pane.

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

Figure 17–2 Functional Settings



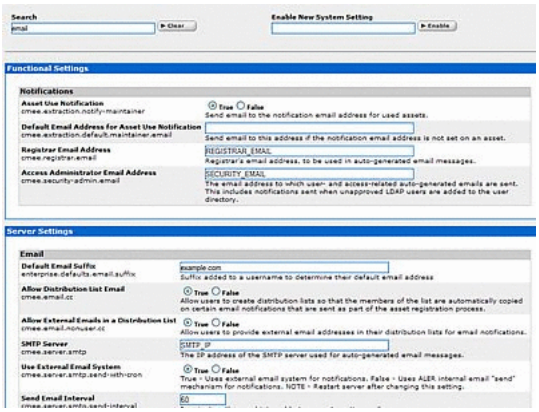
2. Click the plus icon to expand the display of any category or feature group.
3. Click the minus icon to collapse the display.

17.2 Search for System Settings

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

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

Figure 17–3 Functional Settings



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

17.3 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 text box.
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.

17.4 Move Settings to Database

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

1. 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.

For more information, see the Move Application Properties to the Database section in the *Oracle Fusion Middleware Installation Guide for Oracle Enterprise Repository*.

17.5 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 17.5.1, "Functional Settings"
- Section 17.5.2, "Server Settings"
- Section 17.5.3, "Enterprise Authentication Settings"
- Section 17.5.4, "Open API Settings"

- Section 17.5.5, "External Integration Settings"
- Section 17.5.6, "Import / Export and Introspection Settings"

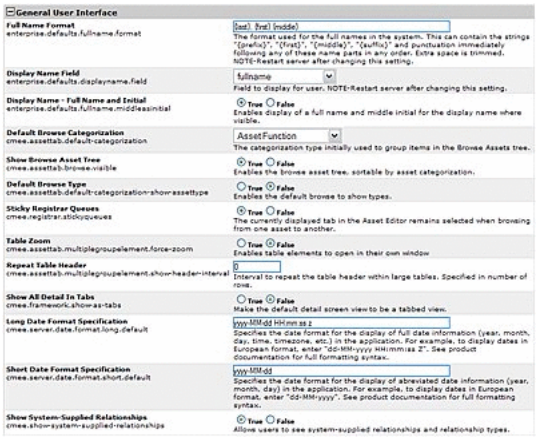
17.5.1 Functional Settings

This section contains the following topics:

- Section 17.5.1.1, "General User Interface"
- Section 17.5.1.2, "Search"
- Section 17.5.1.3, "Print Asset Detail via PDF"
- Section 17.5.1.4, "Projects and Navigator"
- Section 17.5.1.5, "Assets in Progress"
- Section 17.5.1.6, "Policy Management"
- Section 17.5.1.7, "Notifications"
- Section 17.5.1.8, "Access Control"
- Section 17.5.1.9, "Software File Identification (SFID)"
- Section 17.5.1.10, "Use - Download"
- Section 17.5.1.11, "Custom Links"
- Section 17.5.1.12, "Reports"
- Section 17.5.1.13, "Miscellaneous"

17.5.1.1 General User Interface

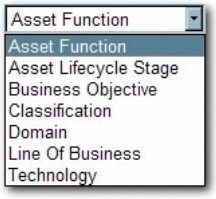
Figure 17–4 General User Interface



Properties

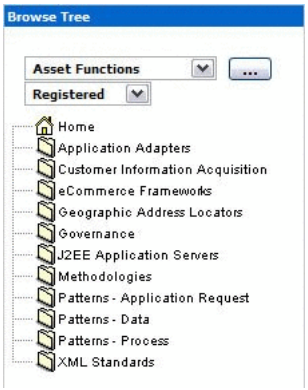
- Default Browse Categorization
  - cmee.assettab.default-categorization
  - The list selection determines the initial asset categorization in the Browse Tree on the Assets screen.

Figure 17–5 Asset Function List



Results in..

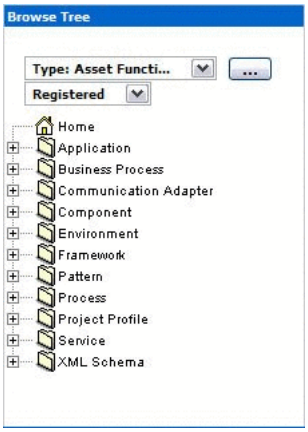
Figure 17–6 Browse Tree



- 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).

    True Results in..

Figure 17–7 Browse Tree

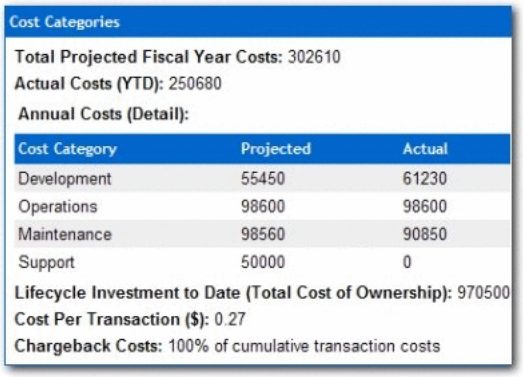


- Sticky Registrar Queues
  - cmee.registrar.stickyqueues

  - 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
  - cmee.assettab.multiplegroupelement.force-zoom
  - The True and False options toggle the ability to open table elements in their own window.
- Repeat Table Header
  - cmee.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.
    * Value of 0 results in...

Figure 17–8 Cost Categories



Cost Categories

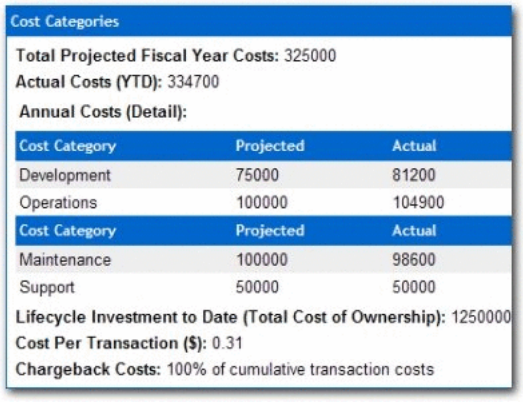
Total Projected Fiscal Year Costs: 302610
Actual Costs (YTD): 250680
Annual Costs (Detail):

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

    * Value of 2 results in...

Figure 17–9 Cost Categories



Cost Categories

Total Projected Fiscal Year Costs: 325000
Actual Costs (YTD): 334700
Annual Costs (Detail):

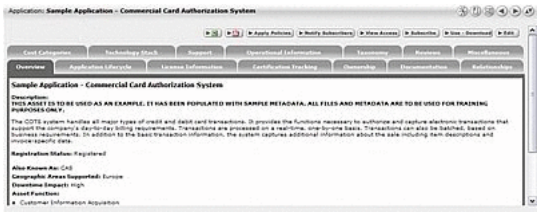
Cost Category	Projected	Actual
Development	75000	81200
Operations	100000	104900
Cost Category	Projected	Actual
Maintenance	100000	98600
Support	50000	50000

Lifecycle Investment to Date (Total Cost of Ownership): 1250000
Cost Per Transaction ($): 0.31
Chargeback Costs: 100% of cumulative transaction costs

- Show All Detail In Tabs

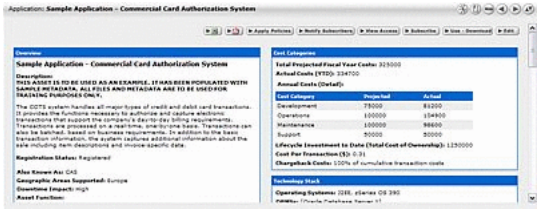
  - cmee.framework.show-as-tabs
  - The True and False options toggle the default display of the asset detail.
    * True results in the tabbed view...

Figure 17–10 Overview Tab



    * False results in the standard view...

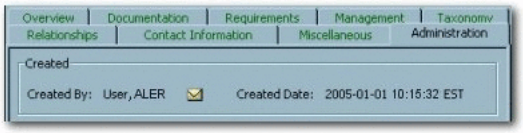
Figure 17–11 Sample Application



- 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).

    Results in..

Figure 17–12 Administration Tab



- 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.
- 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.

Results in..

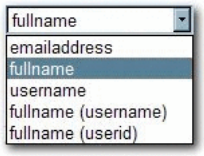
Figure 17–13 Full Name Format

- 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.
- Display Name Field
  - enterprise.defaults.displayname.field
  - The list selection determines the information displayed for the user.

    Note: Changes to this setting require server restart.

Figure 17–14 Display Name Field



Results in..

Figure 17–15 Full Name Format

- 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...

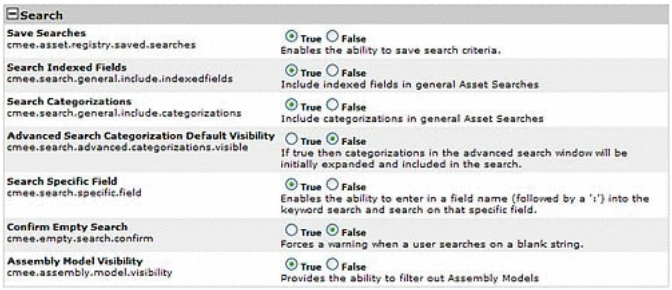
Figure 17–16 Full Name and Initial

False results in..

Figure 17–17 Full Name and Initial

17.5.1.2 Search

Figure 17–18 Search

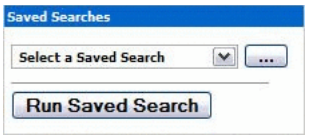


Properties

- 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 17–19 Saved Searches



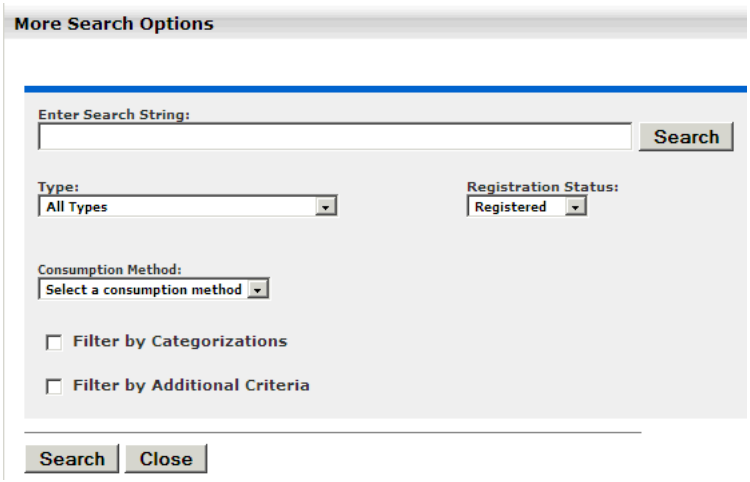
- Search Indexed Fields
  - cmee.search.general.include.indexedfields
  - The True and False options toggle the inclusion of indexed fields in general asset searches.
- Search Categorizations
  - cmee.search.general.include.categorizations

  - 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.

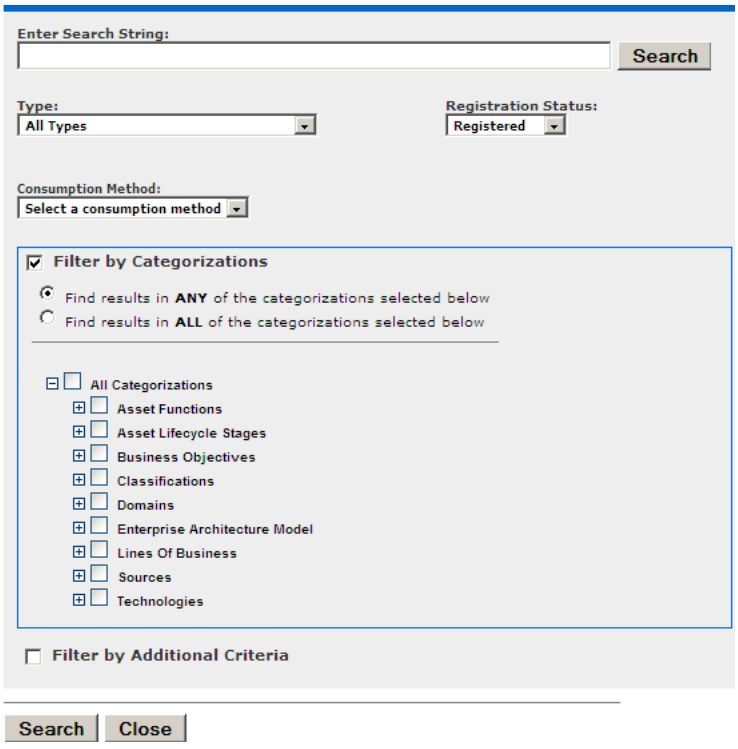
  False results in..

Figure 17–20 More Search Options - Categorizations Tree Not Expanded



True results in..

Figure 17–21 More Search Options - Categorizations Tree Expanded



- 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 17–22 Search

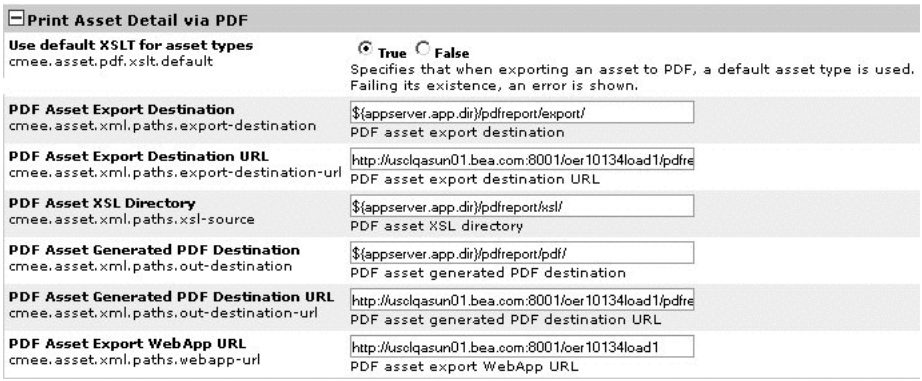
- Confirm Empty Search
  - cmee.empty.search.confirm
  - The True and False options toggle the display of a warning message when a user performs a search on a blank string.

    True displays a warning message.
- 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.
- Search Logging
  - cmee.log.searches will log search criteria for keyword searches only.
  - cmee.log.complete.searches will log search criteria for any criteria entered in a search.

17.5.1.3 Print Asset Detail via PDF

Figure 17–23 Print Asset Detail via PDF

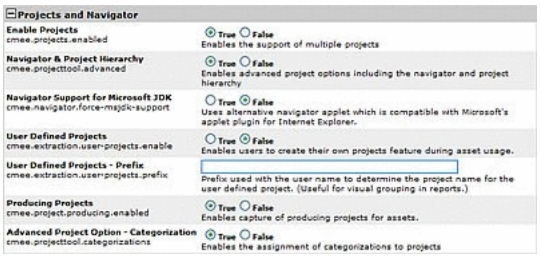


Properties

- 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.

17.5.1.4 Projects and Navigator

Figure 17–24 Projects and Navigator

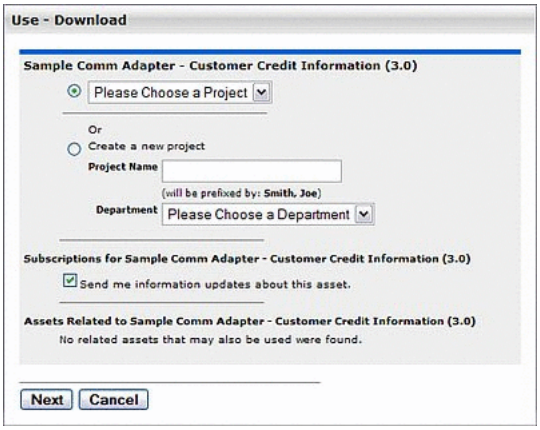


Properties

- 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.

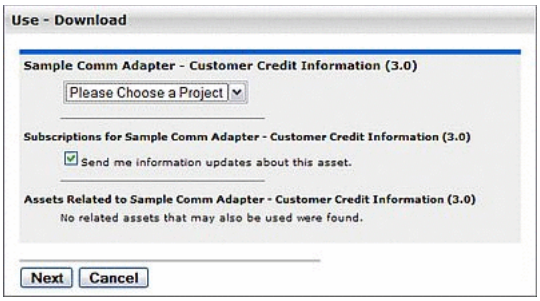
  True results in...

Figure 17–25 Use - Download



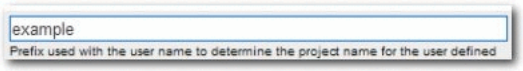
False results in...

Figure 17–26 Use - Download



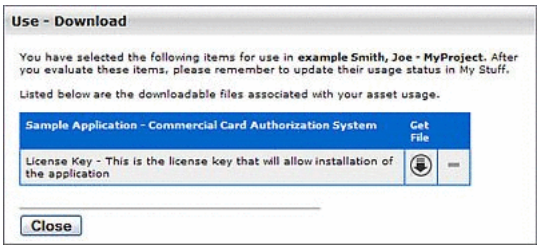
- User Defined Projects - Prefix
  - cmee.extraction.user-projects.prefix
  - As illustrated below, text entered in the property text box becomes the prefix for any user-defined projects. This is useful for visual grouping in reports. (User-defined Projects must be enabled.)

Figure 17–27 Example



Results in..

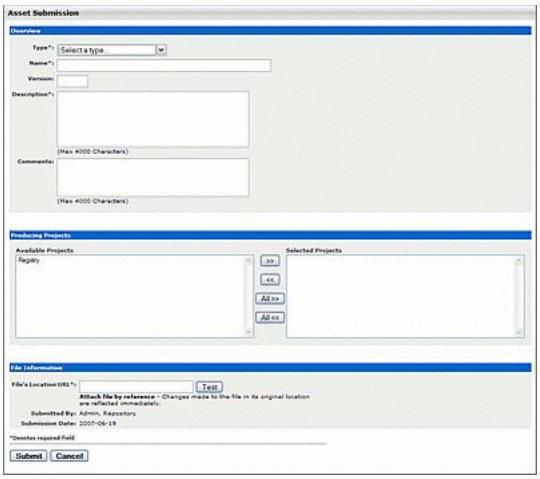
Figure 17–28 Use - Download



- 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.)

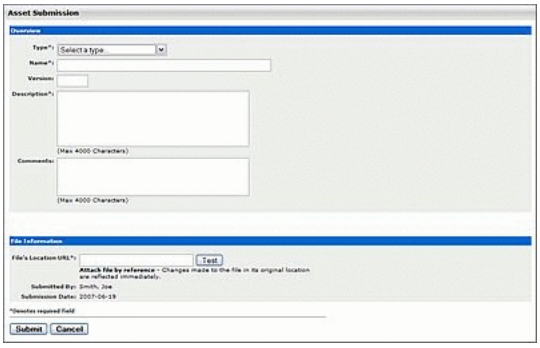
True results in...

Figure 17–29 Asset Submission



False results in...

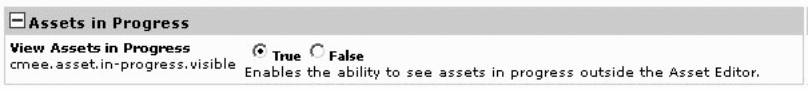
Figure 17–30 Asset Submission



- 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.

17.5.1.5 Assets in Progress

Figure 17–31 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.

17.5.1.6 Policy Management

Figure 17–32 Policy Management

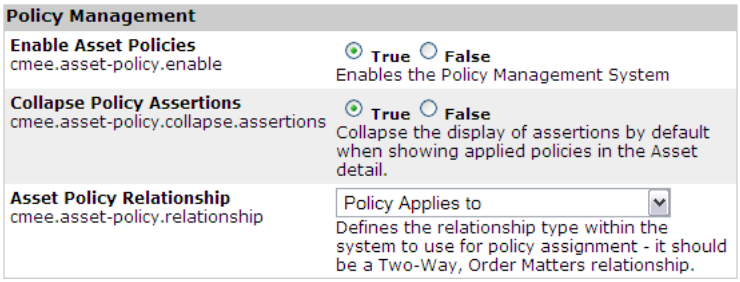
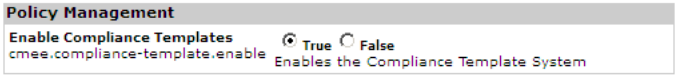


Figure 17–33 Policy Management

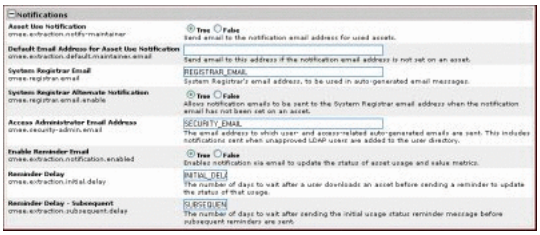


Properties

- Enable Compliance Templates (See Figure 17–33)
  - 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.

17.5.1.7 Notifications

Figure 17–34 Notifications



Properties

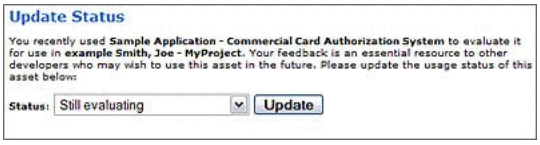
- Asset Use Notification
  - cmee.extraction.notify-maintainer
  - The True and False options toggle the feature that automatically sends an email to the notification address listed for an asset when that asset is used.

Figure 17–35 Miscellaneous

  - 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 below).

Figure 17–36 Update Status



- 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.

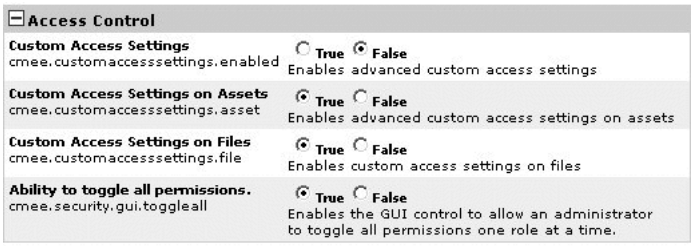
17.5.1.8 Access Control

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.

Figure 17–37 Access Control



17.5.1.9 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.

Figure 17–38 Software File Identification (SFID)

17.5.1.10 Use - Download

Properties

- Asset Usage Custom Field List
  - cmee.extraction.exported-custom-field.
  - This unique setting allows the creation of Custom Asset Usage Fields.

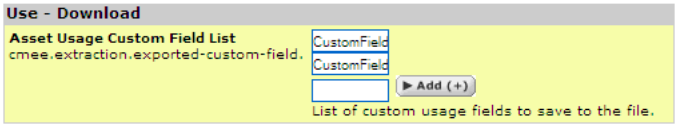
> **Note:** This property appears with a period at the end. When you enter this property in the Search field, enter the property along with the period.

Create Custom Usage Fields

1. Enter the property name in the Enable New System Setting box and press Enter.

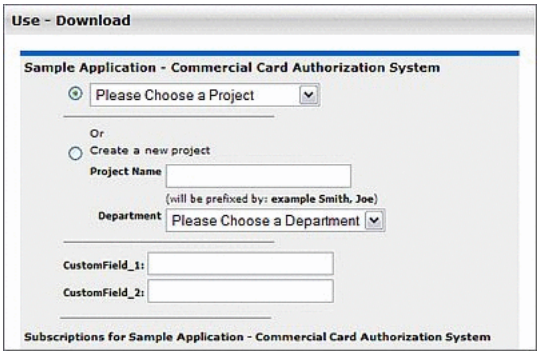
2. As necessary, click **Add** to open additional text boxes.

Figure 17–39 Use - Download



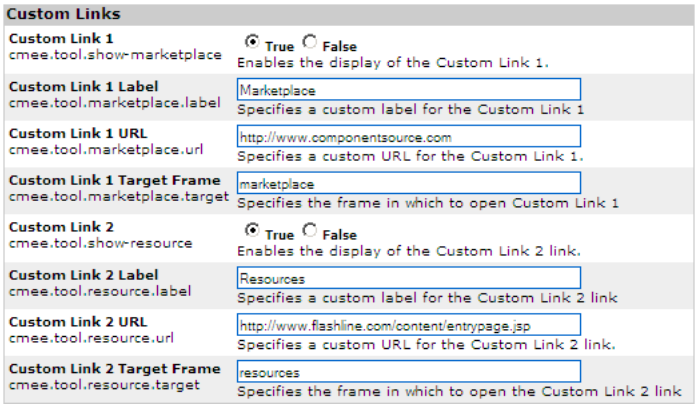
3. Enter a name for each additional Custom Usage Field in the corresponding text box.
4. Click Save when finished. The new Custom Usage Fields appear in the Use - Download dialog.

Figure 17–40 Use - Download



17.5.1.11 Custom Links

Figure 17–41 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 17–42 Custom Link

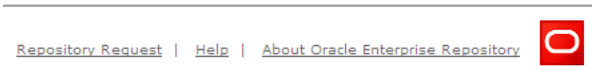


Figure 17–43 Custom Link 1 and Custom Link 2

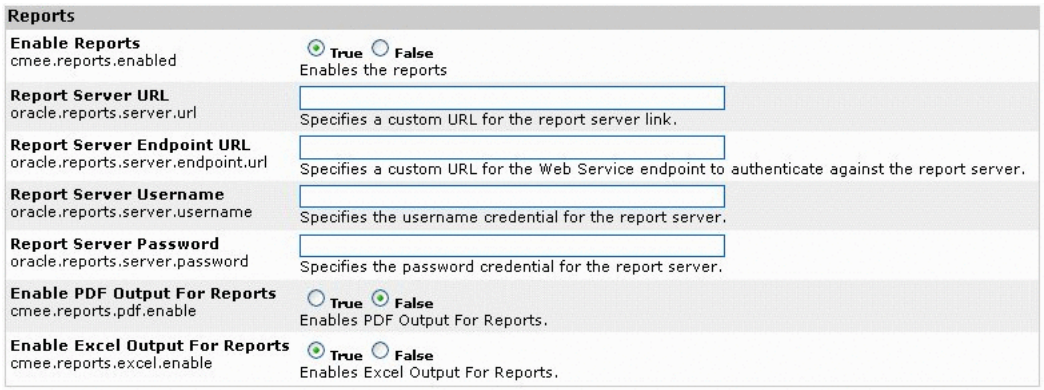
(Illustrates both Custom 1 Link and Custom 2 Link set to True.)

- 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.

17.5.1.12 Reports

Figure 17–44 Reports

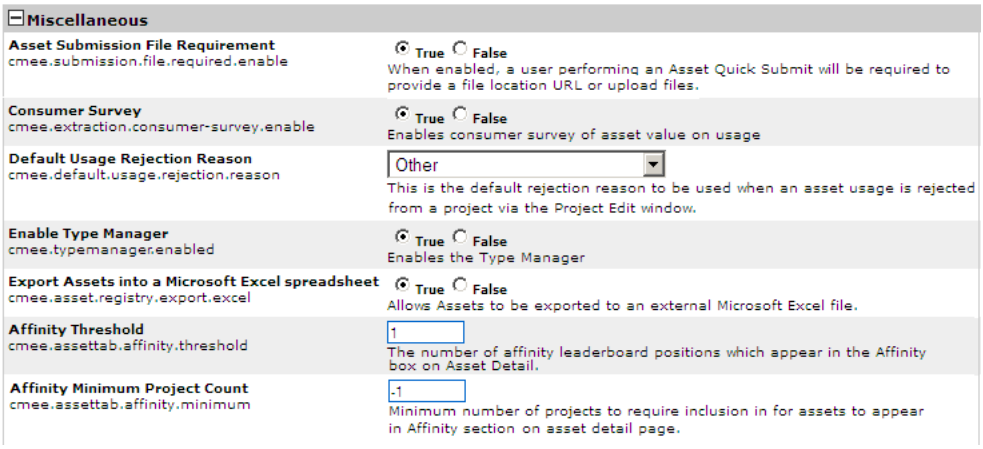


Properties

- 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 PDF Output for Reports
  - oracle.reports.pdf.enable
  - Enables PDF Output for Reports.
- Enable Excel Output for Reports
  - oracle.reports.excel.enable
  - Enables Excel Output for Reports.

17.5.1.13 Miscellaneous

Figure 17–45 Miscellaneous

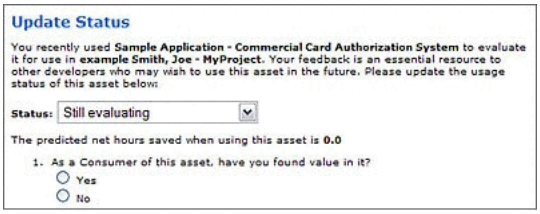


Properties

- 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.

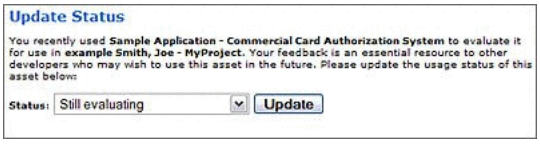
    True displays the full survey...

Figure 17–46 Update Status



False displays only the Status list.

Figure 17–47 Update Status



- 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.

Figure 17–48 Default Usage Rejection Reason

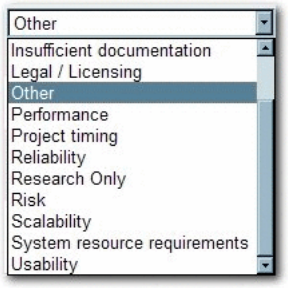


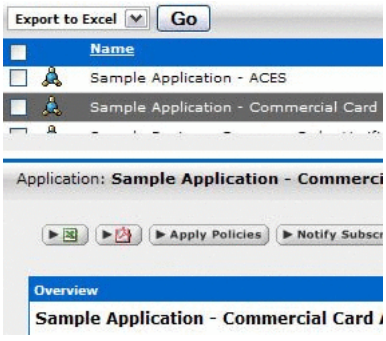
Figure 17–49 Consumed Assets

- 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 and the Excel button in the asset detail.

Figure 17–50 Export Assets



False removes the menu selection and button.

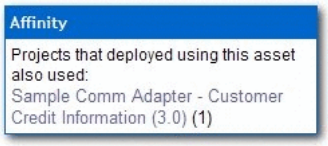
Affinity Settings

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.

Figure 17–51 Affinity



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.

17.5.2 Server Settings

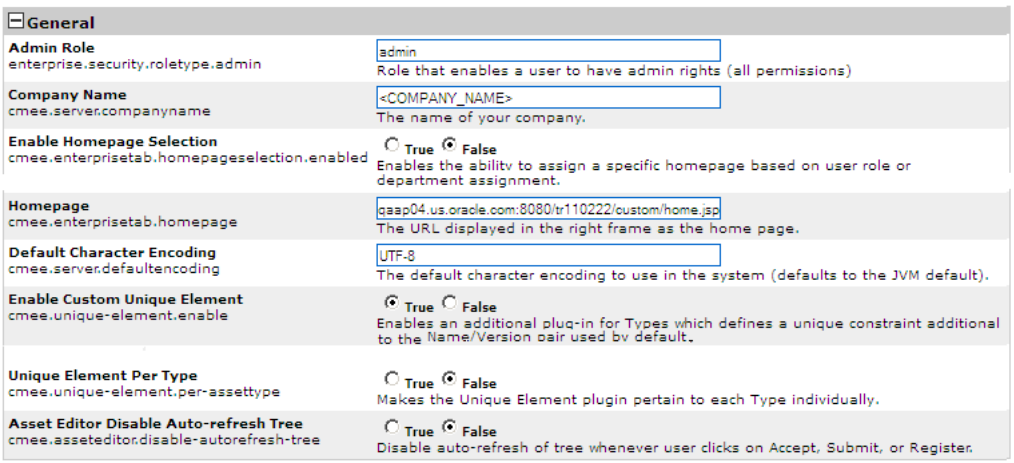
This section contains the following topics:

- Section 17.5.2.1, "General"
- Section 17.5.2.2, "Server Resources"
- Section 17.5.2.3, "Email"
- Section 17.5.2.4, "Upload Area"
- Section 17.5.2.5, "Audit"
- Section 17.5.2.6, "Caching"

Note:

17.5.2.1 General

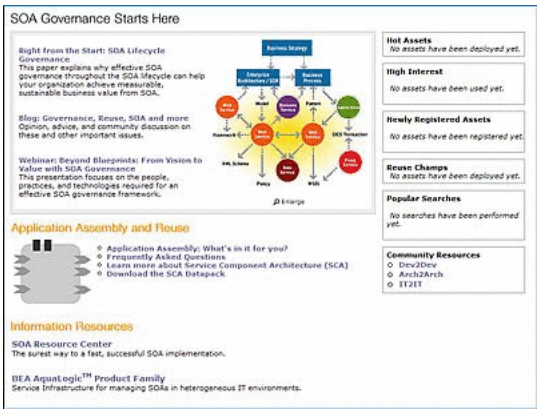
Figure 17–52 General



- 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.
- 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 17–53 SOA Governance

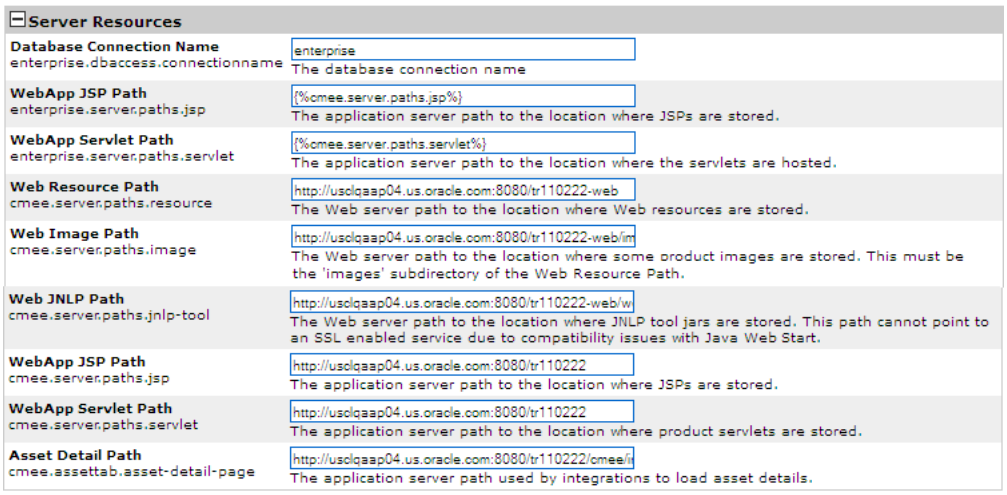


Change this URL as necessary to load a different page as the Oracle Enterprise Repository homepage.

- 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.

17.5.2.2 Server Resources

Figure 17–54 Server Resources



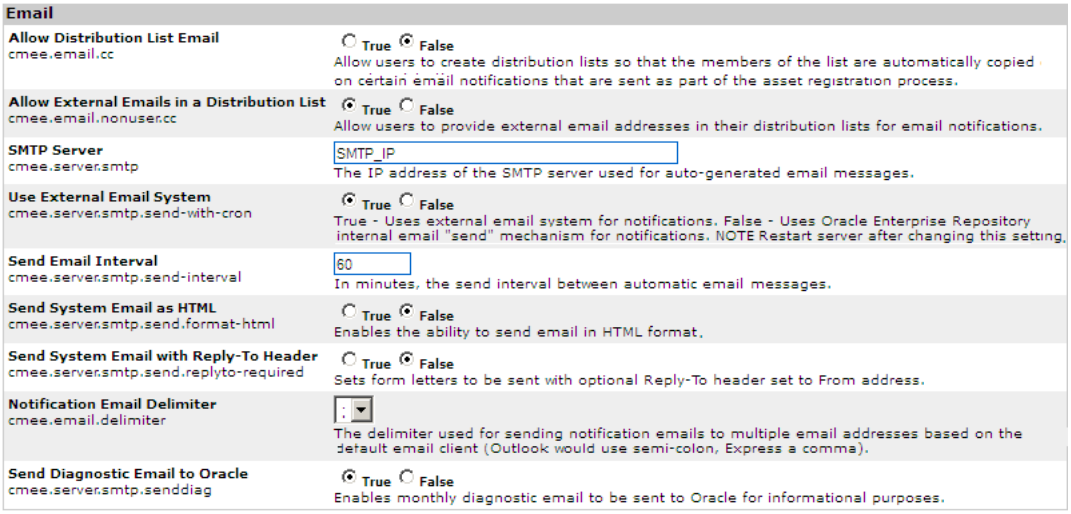
Properties

- 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

  - cmee.server.paths.jsp
  - The filepath to the location on the application server where JSPs are stored.
- WebApp Servlet Path
  - cmee.server.paths.servlet
  - The filepath to the location on the application server where product servlets are stored.
- Asset Detail Path
  - cmee.assettab.asset-detail-page
  - The application server path used by integrations to load asset details.

17.5.2.3 Email

Figure 17–55 Email



Properties

- Allow Distribution List Email
  - cmee.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
  - cmee.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
  - cmee.server.smtp

  - The hostname or IP address of the SMTP server used for auto-generated email messages.

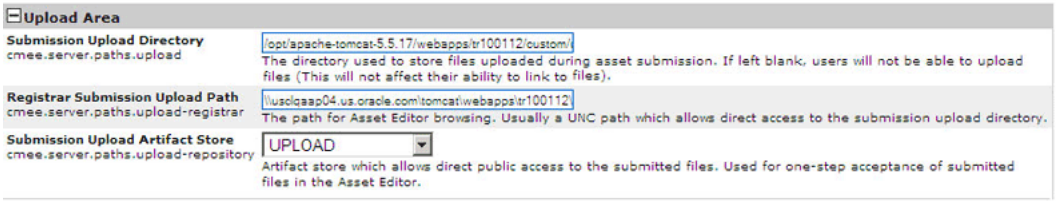
- Use External Email System
  - cmee.server.smtp.send-with-cron
  - True and False options determine the selection of the email mechanisms for automated Oracle Enterprise Repository messages. True enables the external mechanism. False enables Oracle Enterprise Repository's internal email send mechanism.

    Note: Changes to this setting require server restart.
- Send Email Interval
  - cmee.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.

17.5.2.4 Upload Area

Figure 17–56 Upload Area

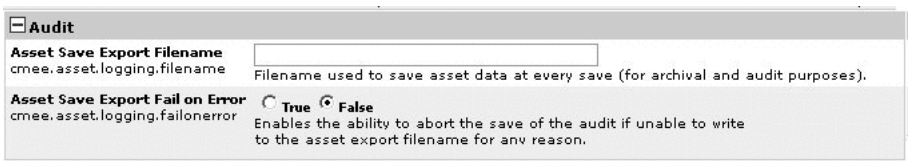


Properties

- 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/Sharity.
    * 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.

17.5.2.5 Audit

Figure 17–57 Audit



Properties

- Asset Save Export Filename
  - cmee.asset.logging.filenamep
  - 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.

17.5.2.6 Caching

Figure 17–58 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.

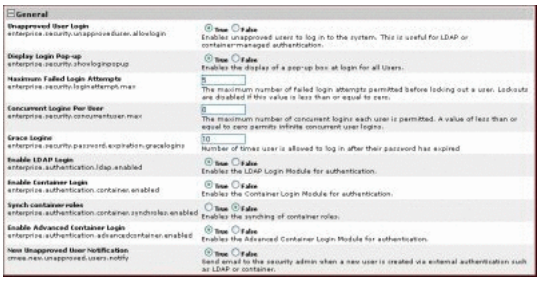
17.5.3 Enterprise Authentication Settings

This section contains the following topics:

- Section 17.5.3.1, "General"
- Section 17.5.3.2, "Authentication Token"
- Section 17.5.3.3, "Password Settings"
- Section 17.5.3.4, "Plugin Login Settings"
- Section 17.5.3.5, "LDAP and Active Directory Settings"
- Section 17.5.3.6, "Single Sign-On (SSO) Settings"
- Section 17.5.3.7, "Container Login Settings"
- Section 17.5.3.8, "Cookie Login Settings"

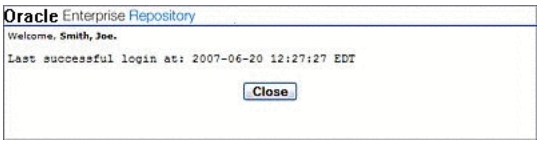
17.5.3.1 General

Figure 17–59 General



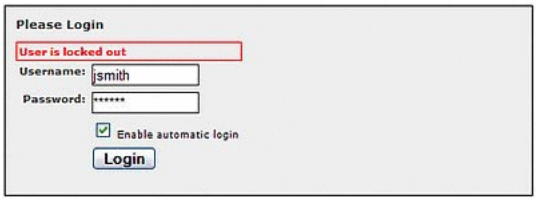
- Default Email Suffix
  - enterprise.defaults.email.suffix
  - The information supplied in this text box is automatically suffixed to usernames to form the default user email address.
- Unapproved User Login
  - enterprise.security.unapproveduser.allowlogin
  - The True and False options toggle the ability for unapproved users to log in to the system. This is useful when using LDAP or container-managed authentication.
- Display Login dialog
  - 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.

Figure 17–60 Oracle Enterprise Repository Login Page



- 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 17–61 Oracle Enterprise Repository Login Page



- 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.
- Enable Advanced Container Login
  - enterprise.authentication.advancedcontainer.enabled
  - Select the True option to enable the Advanced Container Login Module for authentication.
- Synchronize Container Roles
  - enterprise.authentication.container.synchroles.enabled
  - Enables the synchronization of existing Java EE container roles.
- 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.

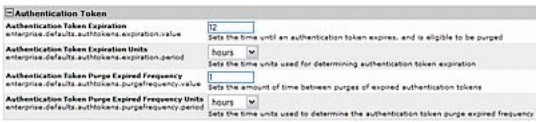
LDAP Settings and its description are as follows:

LDAP Settings	Description
ldap.ssl.enabled	Enables an SSL Connection for LDAP; defaults to false
ldap.host	Host name for the LDAP server; required
ldap.port	LDAP server port number; defaults to 389; optional
ldap.mask	The LDAP mask, either the authentication bind or search criteria; required
ldap.allow-user-creation	Determines if unapproved user accounts can be created; defaults to true; optional

LDAP Settings	Description
ldap.version	The LDAP version used, which can be either 2 or 3; defaults to 2; optional
ldap.bindDN	The distinguished name of the administrator account; optional
ldap.bindPassword	The password for the administrator account; optional
ldap.baseDN	Start location for a search, works with ldap.mask and ldap.scope to define the search; optional
ldap.scope	Determines the scope of the search; optional
ldap.uniqueIDAttrib	Attribute Name that Identifies a Found Entry
ldap.emailAttrib	Found Entry Email Attribute Name
ldap.givennameAttrib	Found Entry First Name Attribute Name
ldap.surnameAttrib	Found Entry Last Name Attribute Name
ldap.telephoneAttrib	Found Entry Telephone Number Attribute Name
ldap.middlenameAttrib	Found Entry Middle Name Attribute Name
ldap.enable-sync-depts	If true, get user departments from LDAP upon login
ldap.enable-synch-roles	If true, get user roles from LDAP upon login
ldap.deptAttrib	Name of attribute of the departments of the entry found
ldap.rbac.roleAttrib	Name of attribute of the roles of the entry found
ldap.redirectDnAttrib	Attribute which identifies a second level lookup to retrieve user info; must be a dn(distinguished name); optional
ldap.retrieve-data-as-admin	Retrieve data using the admin account
ldap.assign-default-roles	If true, users shall be assigned default roles
ldap.follow-referrals	If true, follow referral
ldap.auto-create-missing-roles	If true, auto create missing roles
ldap.auto-create-missing-depts	If true, auto create missing departments

17.5.3.2 Authentication Token

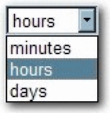
Figure 17–62 Authentication Token



- 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, as shown in Figure 17–63, to select the units of time to be used in conjunction with the Authentication Token Expiration setting.

Figure 17–63 Authentication Token Expiration Units List



- 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.

17.5.3.3 Password Settings

Figure 17–64 Password Settings

- 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:

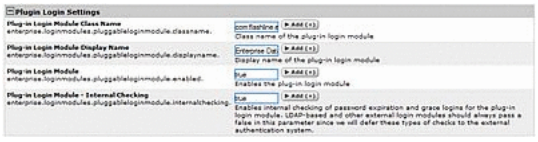
    ` ~ ! @ # $ % ^ & * ( ) - _ = + [{] } \ \ | ; : ' \ " , < . > / ?

    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.

17.5.3.4 Plugin Login Settings

Figure 17–65 Plugin Login Settings



- 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.

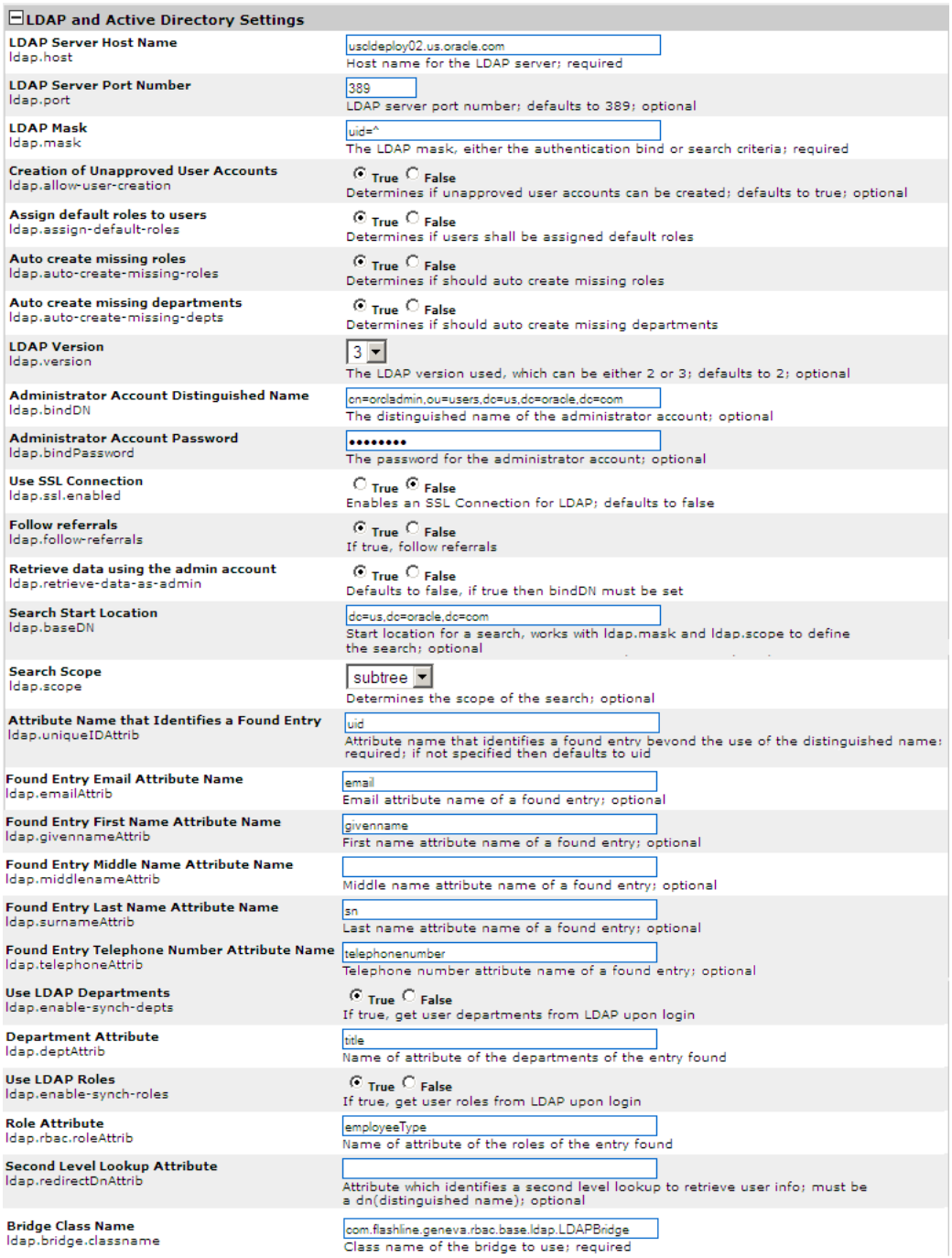
    Click **+Add** as necessary to add and enable/disable additional plug-in modules.
- Plug-in Login Module - Internal Checking
  - enterprise.loginmodules.pluggableloginmodule.internalchecking
  - Enter `true` in the text box to enable internal checking of password expiration and grace logins for the configured 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.

17.5.3.5 LDAP and Active Directory Settings

Figure 17–66 LDAP and Active Directory Settings



- 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

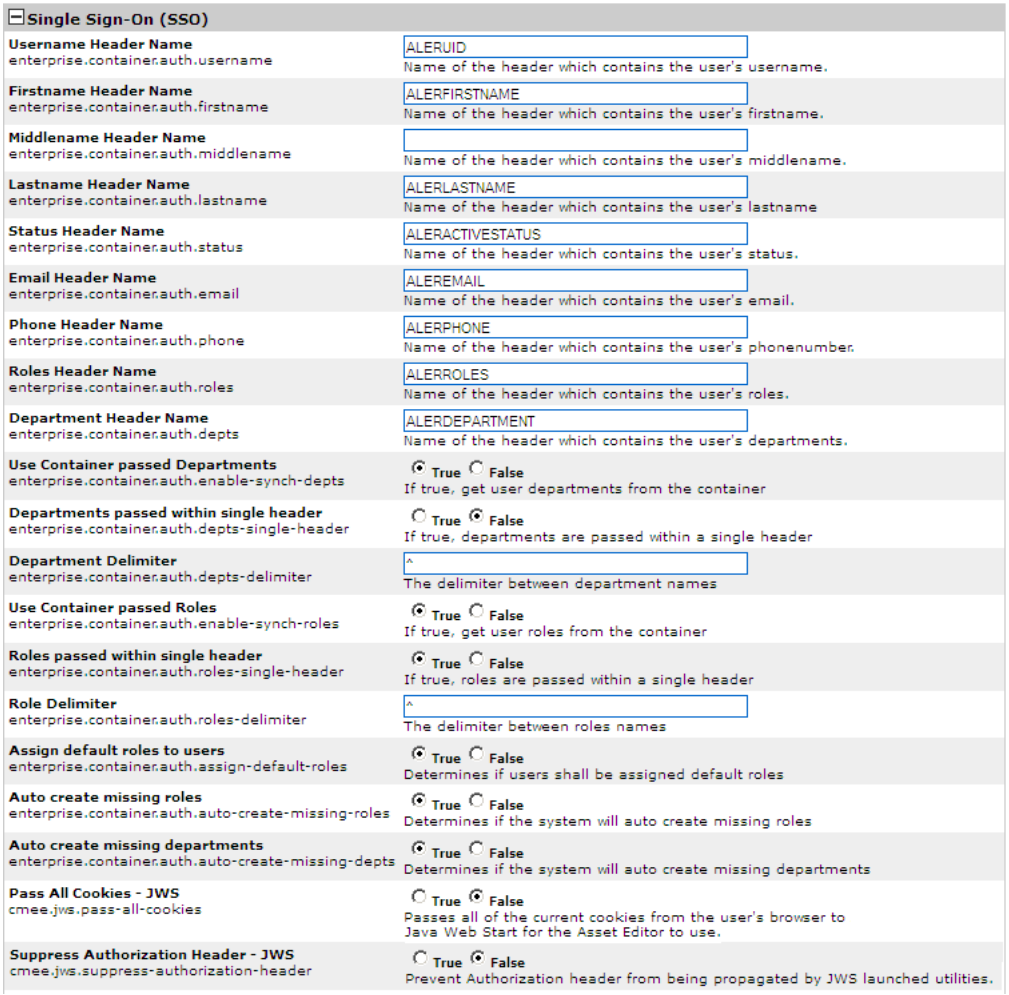
  - Enter the LDAP server port number. The default is 389. This is an optional field.
- LDAP Mask
  - ldap.mask
  - Enter the LDAP mask, either the authentication bind or search criteria. This is a mandatory field.
- Creation of Unapproved User Accounts
  - ldap.allow-user-creation
  - Select True to determine 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.

17.5.3.6 Single Sign-On (SSO) Settings

Figure 17–67 Single Sign-On (SSO) Settings



- 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 phonenumber.
- 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.

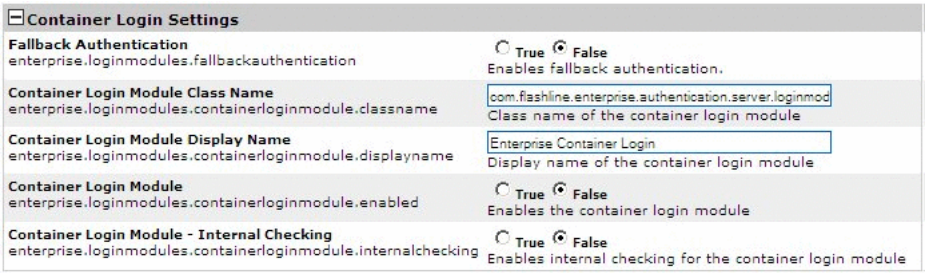
SSO Settings and its description are as follows:

SSO Settings	Description
enterprise.container.auth.username	Name of the header which contains the user's username
enterprise.container.auth.firstname	Name of the header which contains the user's firstname
enterprise.container.auth.middlename	Name of the header which contains the user's middlename
enterprise.container.auth.lastname	Name of the header which contains the user's lastname

SSO Settings	Description
enterprise.container.auth.status	Name of the header which contains the user's status. Values may be 0=ACTIVE, 10=UNAPPROVED, 20=LOCKED_OUT, 30=INACTIVE
enterprise.container.auth.email	Name of the header which contains the user's email
enterprise.container.auth.phone	Name of the header which contains the user's phonenumber
enterprise.container.auth.roles	Name of the header which contains the user's roles
enterprise.container.auth.depts	Name of the header which contains the user's departments
enterprise.container.auth.enable-synch-depts	If true, get user departments from the container
enterprise.container.auth.enable-synch-roles	If true, get user roles from the container
enterprise.container.auth.assign-default-roles	If true, users shall be assigned default roles
enterprise.container.auth.auto-create-missing-roles	If true, the system will auto create missing roles
enterprise.container.auth.auto-create-missing-depts	If true, the system will auto create missing departments
enterprise.container.auth.roles-single-header	If true, roles are passed within a single header
enterprise.container.auth.roles-delimiter	The delimiter between roles names
enterprise.container.auth.depts-single-header	If true, departments are passed within a single header
enterprise.container.auth.depts-delimiter	The delimiter between department names

17.5.3.7 Container Login Settings

Figure 17–68 Container Login Settings

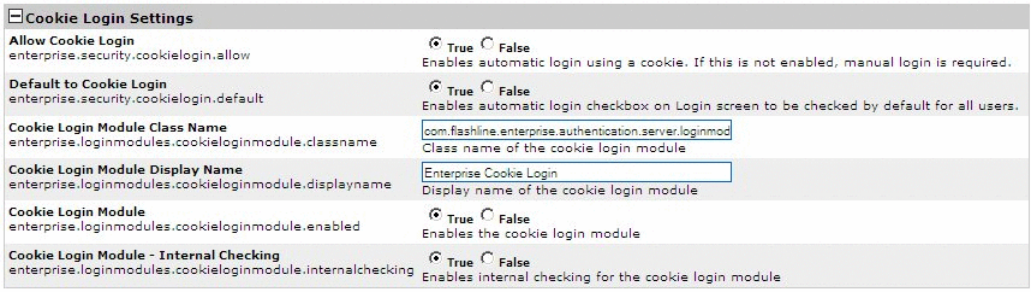


- Fallback Authentication
  - enterprise.loginmodules.fallbackauthentication
  - Enables fallback authentication.
- Container Login Module Class Name
  - enterprise.loginmodules.containerloginmodule.classname

  - 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.

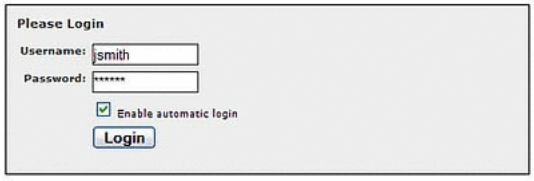
17.5.3.8 Cookie Login Settings

Figure 17–69 Cookie Login Settings



- Allow Cookie Login
  - enterprise.security.cookielogin.allow
  - Select the True option to enable automatic login using a cookie. If disabled, manual login is required.
- Default to Cookie Login
  - enterprise.security.cookielogin.default
  - Select the True option to set Enable automatic login as the default in the Oracle Enterprise Repository login dialog.

Figure 17–70 Oracle Enterprise Repository Login Page

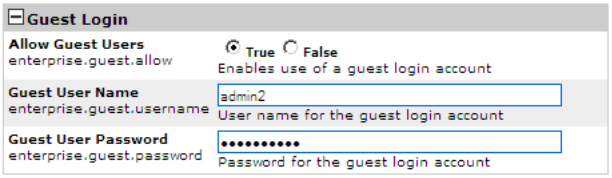


- Cookie Login Module Class Name

  - enterprise.loginmodules.cookieloginmodule.classname
  - Text entered in the text box identifies the class name of the cookie login module.
- Cookie Login Module Display Name
  - enterprise.loginmodules.cookieloginmodule.displayname
  - Text entered in the text box appears as the display name of the cookie login module.
- Cookie Login Module
  - enterprise.loginmodules.cookieloginmodule.enabled
  - Select the True option to enable the cookie login module.
- Cookie Login Module - Internal Checking
  - enterprise.loginmodules.cookieloginmodule.internalchecking
  - Select the True option to enable internal checking for the cookie login module.

17.5.3.9 Guest Login Settings

Figure 17–71 Guest Login Settings



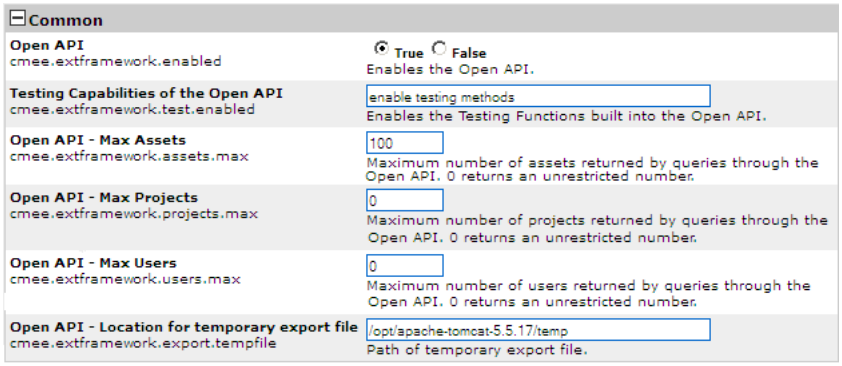
- Allow Guest Users
  - enterprise.guest.allow
  - Select the True option to enable use of a guest login account.
- Guest User Name
  - enterprise.guest.username
  - The value entered in the text field represents the user name for the guest login account.
- Guest User Password
  - enterprise.guest.password
  - The value entered in the text field represents the password for the guest login account.

17.5.4 Open API Settings

This section contains the following open API settings:

17.5.4.1 Open API

Figure 17–72 Open API



Properties

- Open API
  - cmee.extframework.enabled
  - Enables the Open API.
- Testing Capabilities of the Open API
  - cmee.extframework.test.enabled
  - Enables the testing Functions built into 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.

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

17.5.5 External Integration Settings

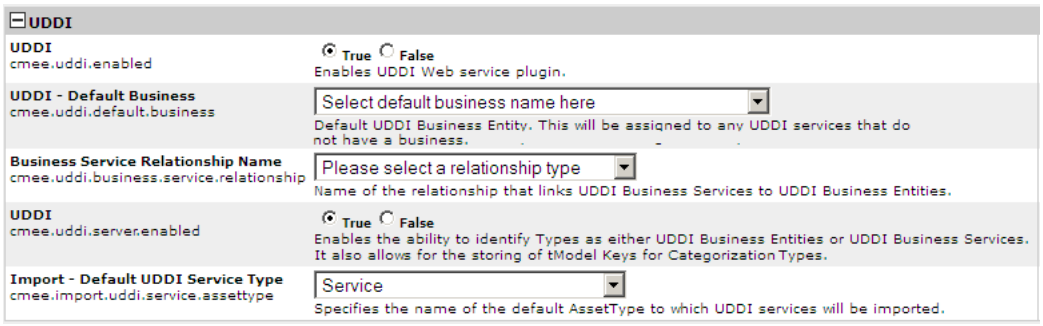
This section contains the following topics:

- Section 17.5.5.1, "UDDI"

- Section 17.5.5.2, "File Stores"
- Section 17.5.5.3, "Rational ClearCase"
- Section 17.5.5.4, "Eventing"

17.5.5.1 UDDI

Figure 17–73 UDDI



- 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.

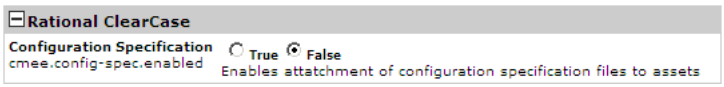
17.5.5.2 File Stores

Figure 17–74 File Stores

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

17.5.5.3 Rational ClearCase

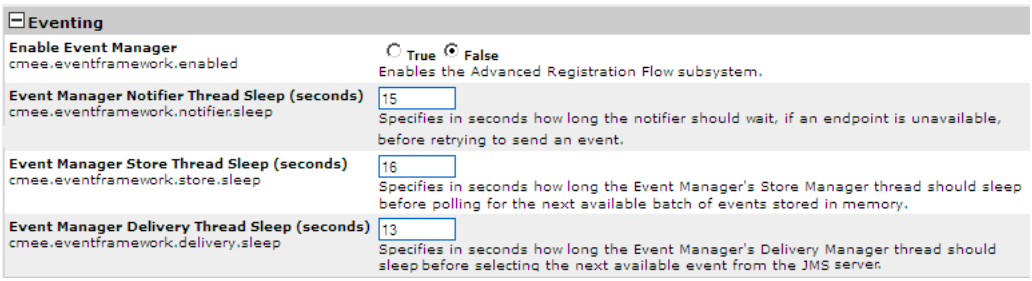
Figure 17–75 Rational ClearCase



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

17.5.5.4 Eventing

Figure 17–76 Eventing



- 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.

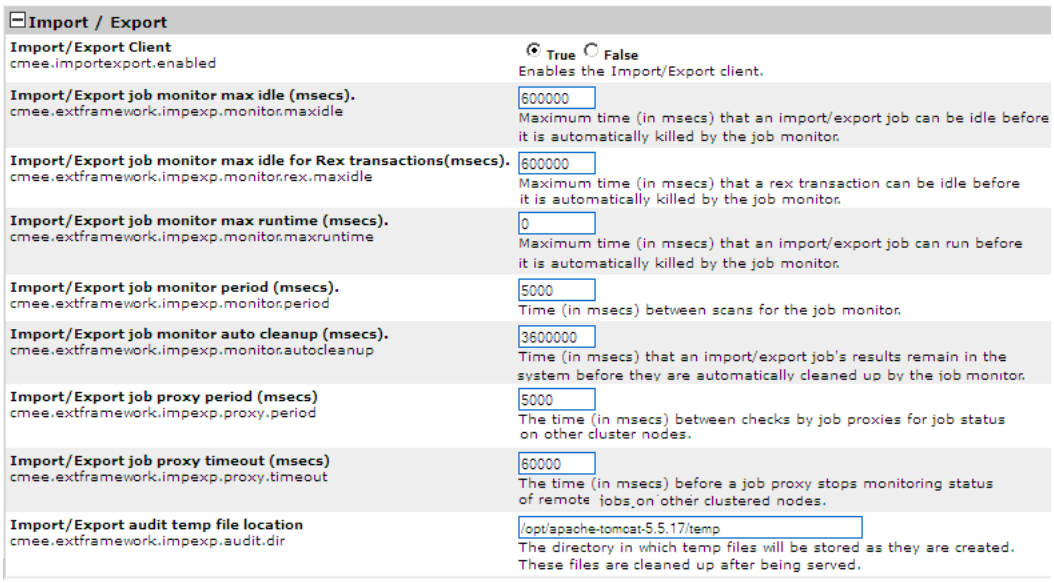
17.5.6 Import / Export and Introspection Settings

This section contains the following topics:

- Section 17.5.6.1, "Import / Export"
- Section 17.5.6.2, "WSDL and XSD"

17.5.6.1 Import / Export

Figure 17–77 Import / Export

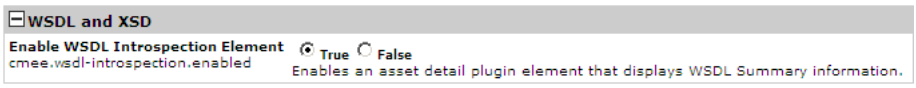


- Import/Export Client
  - cmee.importexport.enabled
  - Select the True option to enable the Import / Export client.
- Import/Export job monitor max idle (msecs)
  - cmee.extframework.impexp.monitor.maxidle
  - The value entered in the text field represents the maximum time (in msecs) that an import/export 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 msecs) between scans for the job monitor.
- Import/Export job monitor auto cleanup (msecs)
  - cmee.extframework.impexp.monitor.autocleanup
  - The value entered in the text field represents the time (in msecs) 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 (msecs)
  - cmee.extframework.impexp.proxy.period
  - The value entered in the text field represents the time (in msecs) between checks by job proxies for job status on other cluster nodes.
- Import/Export job proxy timeout (msecs)
  - cmee.extframework.impexp.proxy.timeout
  - The value entered in the text field represents the time (in msecs) 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.

17.5.6.2 WSDL and XSD

Figure 17–78 WSDL and XSD



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

17.6 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.

A

Configuring Oracle Enterprise Repository Workflow

This chapter describes how to set up and configure Oracle Enterprise Repository Workflow and how to use the registration process to manage assets in Oracle Enterprise Repository.

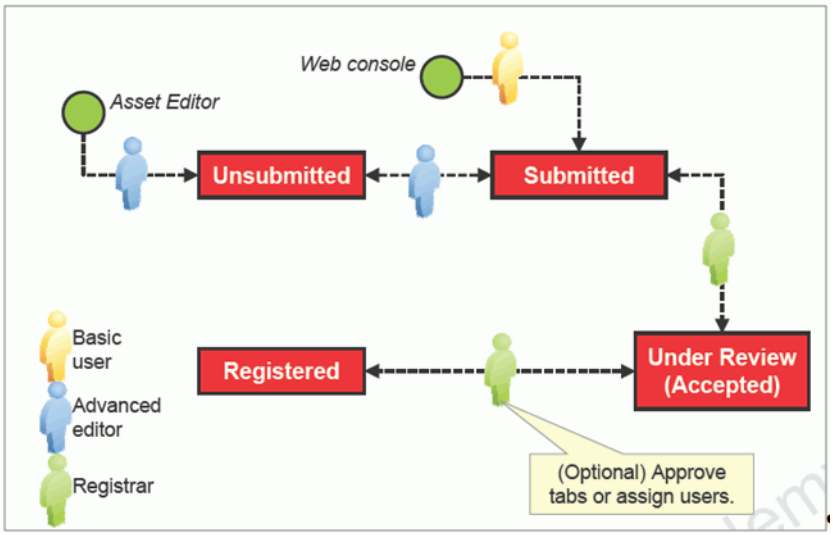
This chapter contains the following sections:

- Section A.1, "Introduction to Oracle Enterprise Repository Workflow"
- Section A.2, "Introduction to Oracle Enterprise Repository Automated Workflows"
- Section A.3, "Installing Oracle Enterprise Repository Workflow into Oracle BPM 10.3"
- Section A.4, "Configuring the Oracle Enterprise Repository Event Manager"
- Section A.5, "Administering Oracle Business Process Management Processes"
- Section A.6, "Configuring Automated Workflows"
- Section A.7, "Configuring JMS Servers for Oracle Enterprise Repository"
- Section A.8, "Monitoring and Managing Events"
- Section A.9, "Extending the Event Manager for Web Service Endpoints"
- Section A.10, "Working with OER Workflow"
- Section A.11, "Getting Started with Automated Workflows"
- Section A.12, ""Community Flow" Use Case"

A.1 Introduction to Oracle Enterprise Repository Workflow

Assets determined to be appropriate for reuse are submitted to the repository. Submitted assets are reviewed by the registrar, who determines which assets will proceed through the registration process.

Figure A–1 Oracle Enterprise Repository Workflow



As shown in Figure A–1 assets accepted by the registrar enter the registration work queues. The submitter can track the asset's progress toward registration using the My Submissions folder on the My Stuff page. Submitters are notified about rejected assets and the reason for rejection (for instance, a duplicate asset).

Oracle Business Process Management (Oracle BPM) and OER

Processes drive the business. If you execute a sales order, escalate a hiring decision, decide which records to keep, or determine how to price a product, you are acting within an implicit or explicit business process. Oracle Business Process Management is a complete set of tools for creating, executing, and optimizing business processes. Oracle BPM leverages your existing IT investments and is specially tuned for line-of-business owners.

Oracle Enterprise Repository bundles pre-built BPM asset registration flows that automate the registration and governance processes. Pre-built BPM asset registration flows can be configured to use the asset types, Lifecycle stages, categorizations, users and other OER constructs specific to your installation. The registration flows can be chained together to support a wide variety of use cases out of the box. If the pre-built flows are not sufficient for your purposes, you can use the predefined Oracle BPM endpoint or create your own Web Service endpoints to subscribe to Oracle Enterprise Repository events. You can use the Oracle BPM Designer to create new Governance workflows. There are also event monitoring and logging tools for troubleshooting and tuning purposes.

For more information on developing workflows: in Oracle Business Process Manager, see *Oracle Fusion Middleware Modeling and Implementation Guide for Oracle Business Process Management*.

A.2 Introduction to Oracle Enterprise Repository Automated Workflows

Oracle Enterprise Repository provides a set of predefined flows designed to automate common Oracle Enterprise Repository tasks, such as asset submission, acceptance, registration, synchronization of Oracle Enterprise Repository and Oracle Service Registry, and other governance processes.

This section includes the following topics:

- Section A.2.1, "Software Components"
- Section A.2.2, "Automated Workflows"
- Section A.2.3, "Event Management Tools"

A.2.1 Software Components

Automated Workflows includes the following software components:

Oracle Enterprise Repository Event Manager

The Event Manager emits asset registration events in the form of Web Service messages. These events trigger pre-built flows that automate Oracle Enterprise Repository asset submission, acceptance, registration, and other governance processes.

For more information, see Section A.4, "Configuring the Oracle Enterprise Repository Event Manager".

Subscription Manager

The Subscription Manager is XML-based configuration file that is responsible for managing the event subscriptions by the Web Service endpoints (either the predefined ALPBM endpoint or user-defined endpoints) where matched events are delivered. The Event Manager uses the **EndPointEventSubscription.xml** file to load information about the endpoints where events need to be delivered.

For more information, see Section A.4, "Configuring the Oracle Enterprise Repository Event Manager".

JMS Server

The Event Manager uses an embedded version of Apache ActiveMQ JMS Server that is enabled by default. The embedded JMS server is configured to run out-of-the-box without any additional configuration. However, you can also configure the Event Manager to use an external JMS server, such as Weblogic Server JMS or IBM MQSeries.

For more information, see Section A.7, "Configuring JMS Servers for Oracle Enterprise Repository".

Event Monitor

A tool to monitor the events that are generated by the Event Manager. The Event Monitor peeks into the event traffic and prints information, such as the event body and event properties.

For more information, see Section A.8, "Monitoring and Managing Events".

Oracle Business Process Management

Oracle Business Process Management is included with your purchase of Oracle Enterprise Repository. You may use Oracle Business Process Management to run the pre-built registration flows, modify existing workflows that are supplied with Oracle Enterprise Repository, and implement new Repository-centric workflows.

A.2.2 Automated Workflows

The automated workflows can be run out-of-the box or can be tailored to suit your environment. For more information, see Section A.6, "Configuring Automated Workflows".

The following automated workflows are delivered:

- **Community Assignment Flow**: Provides a way to automate the asset acceptance, assignment, and registration process by allowing the configuration of automated assignment rules and also provides the notion of federated registrars among different authorities. For more information, see Section A.6.4.1, "Configuring Community Flows".
- **Automated Acceptance and Automated Registration Flow**: In addition to using the Community Flows to automatically accept and register the assets, a number of user roles can be used to accept and register assets. For more information, see Section A.6.4.2, "Configuring Automated Acceptance and Automated Registration Flows".
- **Multi-tier Approval Flow**: Structures the tab approval process in multiple steps called tiers. Asset approval tabs can be grouped in tiers, and the Multi-tier Approval flow tracks each tier to verify whether all the tabs are approved by the designated approvers. As soon as the last tab in a tier is approved, the flow starts the next tier by assigning the asset to the next level of designated approvers. For more information, see Section A.6.5, "Multi-tier Automatic Assignment Flows".
- **Metadata Change Flow**: Exposes a flexible framework where state changes or metadata changes can be wired to actions. The Metadata Change flows come with the a set of pre-bundled actions. New actions can be developed in the form of Oracle Enterprise Repository flows and can be plugged in. For more information, see Section A.6.6, "Metadata Change Flows".
- **Time-based Escalation Flow**: Tracks assets in various states and notifies all interested parties. There are four different kinds of time-based escalation flows and each one can be configured individually. For more information, see Section A.6.7, "Time-based Escalation Flows".
- **Validation Expiration Flow**: Tracks expired assets prior to the specified expiration date, as well as at the day of expiration, and sends warning notifications to all interested parties. For more information, see Section A.6.8, "Validation Expiration Flows".
- **AutoSyncAlerToUddi Flow**: Moves a service from Oracle Enterprise Repository to Oracle Service Registry when the Asset Lifecycle Stage is changed from Stage 4 - Build to Stage 5 - Production. For more information, see Section 10.3.1.1, "Invoking the Oracle Registry Repository Exchange Utility Using Workflows".

A.2.3 Event Management Tools

There are event monitoring and logging tools for troubleshooting and tuning purposes.

A.2.3.1 Web-based Process Administrator

The Oracle Business Process Management Process Execution Administrator actively manages the orchestration of asset registration events in the form of Web Service messages. For more information, see Section A.5, "Administering Oracle Business Process Management Processes".

A.2.3.2 Log Viewer

The Oracle Business Process Management Log Viewer enables you to read information logged by the Process Execution Engine. A set of log files is created for each project you define. The Studio Log Viewer reads the files and displays them to help you monitor and trace Engine execution. For more information, see Section A.5.5, "Using the Oracle Business Process Management Log Viewer".

A.2.3.3 Email Notification Templates

The Automated Registration Flows automatically send E-mail notifications under many circumstances. Administrators can customize the E-mail subject, body, etc., the same way as other E-mail templates. See Section A.6.9, "Customizing Email Notification Templates".

A.2.3.4 Workflow Configuration Tools

There are workflow configuration tools for generating new configuration file, refreshing existing files, and encrypting passwords. For more information, see Section A.8, "Monitoring and Managing Events".

A.2.3.5 Generating a New Config File

Oracle Enterprise Repository administrators may need to configure and customize flows because there are new asset types, projects, categorizations, etc. The Generate Config XML tool connects to Oracle Enterprise Repository and creates a new file that can be customized.

A.2.3.6 Refreshing an Existing Config File

The Refresh Config XML tool lets you to refresh a Config XML file without restarting the Event Manager.

A.2.3.7 Encrypting Config File Passwords

The security Encrypt Password tool lets you to encrypt the passwords for security reasons.

A.3 Installing Oracle Enterprise Repository Workflow into Oracle BPM 10.3

The Oracle Enterprise Repository bundles pre-built BPM asset registration flows that automate the registration and governance processes defined in Section A.10, "Working with OER Workflow". For ease of use, you can use the predefined Oracle BPM endpoint or create your own Web Service endpoints to subscribe to Oracle Enterprise Repository events. You can use the Oracle BPM Designer to create new Governance workflows. There are also event monitoring and logging tools for troubleshooting and tuning purposes.

The Oracle BPM Workflow archive is bundled with OER and is found at <OER_HOME>/core/tools/solutions. The archive includes workflow.xml along with OS and Ant scripts (build.xml) primarily for initializing the OER workflow engine in Oracle BPM. The Ant script in particular depends on another file, build.properties. This file identifies the location of your Oracle BPM installation, server, and database.

Additional workflow-related utilities can be found at <OER_HOME>/core/workflow-tools. These tools are generally intended to help with workflow maintenance, such as clearing the OER event queue or refreshing OBPM with the latest workflow configuration.

This section describes how to install the Oracle Enterprise Repository Workflow into Oracle BPM 10.3.2. It contains the following topics:

- Section A.3.1, "Requirements"
- Section A.3.2, "Step 1: Install Oracle BPM 10.3.2"
- Section A.3.3, "Step 2: Unzip the Oracle Enterprise Repository Workflow Installer"

- Section A.3.4, "Step 3: Configure the build.properties File"
- Section A.3.5, "Step 4: Configure the setenv File"
- Section A.3.6, "Step 5: Generate (or Regenerate) the workflow.xml File"
- Section A.3.7, "Step 6: Edit the workflow.xml File"
- Section A.3.8, "Step 7: Encrypt the workflow.xml File"
- Section A.3.9, "Step 8: Copy the JDBC jar(s)"
- Section A.3.10, "Step 9: Verify Permissions"
- Section A.3.11, "Step 10: Run the Setup Script"
- Section A.3.12, "Step 11: Verify the Setup Script"

A.3.1 Requirements

Before installing the Oracle Enterprise Repository Workflow consider these requirements:

- Apache ANT version 1.6.5 or later
- Oracle BPM 10.3.2 Enterprise Install
- Oracle Enterprise Repository 11*g*
- Oracle Enterprise Repository URL and port number
- DBA User account able to create users/tables/indexes within the selected database server
- Appropriate JDBC drivers for your selected database server
- Identify the DB permissions required to do the install
- JDK version 1.6 or later (JDK version 1.5 for installation in non-GUI mode)
- Download and apply the latest hotfix build to the Oracle BPM installation

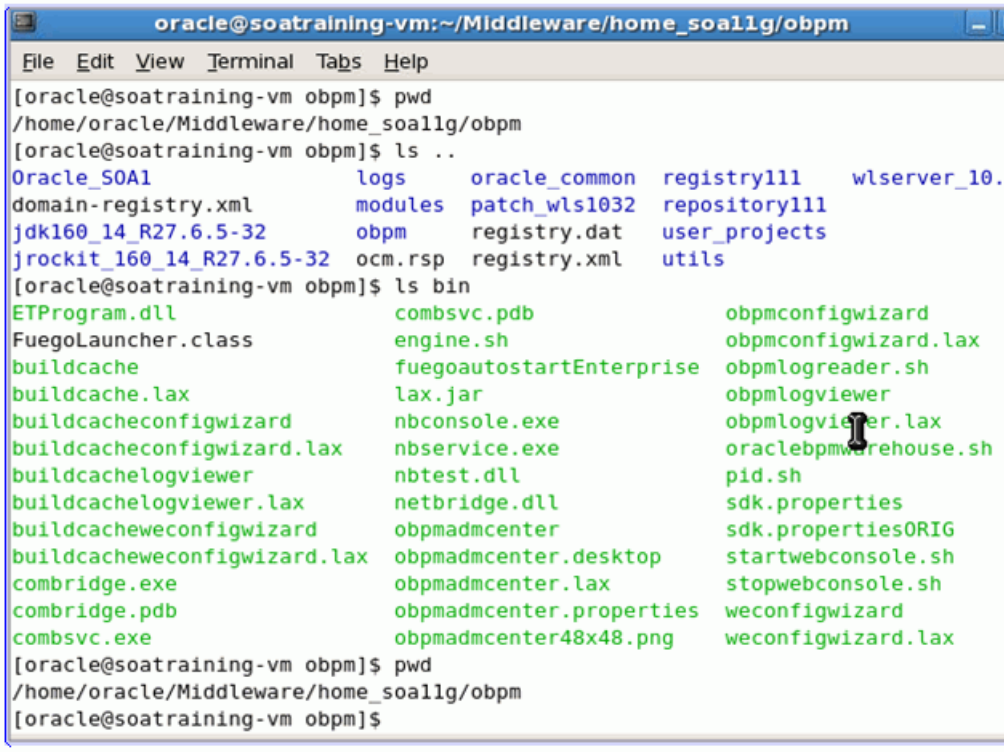
A.3.2 Step 1: Install Oracle BPM 10.3.2

To install Oracle BPM 10.3.2:

1. Download Oracle Business Process Management (Enterprise) 10gR3 (10.3.2) from http://www.oracle.com/technetwork/middleware/bpm/downloads/index-100737.html.
2. Set JAVA_HOME to point to Ps4JDK.
3. Run the BPM Install wizard and accept the defaults. Enter /Oracle/Middleware/obpm/enterprise as your installation directory.
4. Once you finish the Install Wizard, select Exit Installer and close down the wizard.

   The /Oracle/Middleware/obpm/enterprise/bin directory holds all the scripts to launch the BPM engines as shown in Figure A–2.

Figure A–2 Example of the /Oracle/Middleware/obpm/enterprise/bin/directory



A.3.3 Step 2: Unzip the Oracle Enterprise Repository Workflow Installer

To unzip the Oracle Enterprise Repository Workflow installer:

1. In the Oracle Enterprise Repository installation location, find the Oracle Enterprise Repository Workflow installer in the following directory: *ORACLE_HOME/repositoryXXX/core/tools/solutions/11.1.1.x.x-OBPM-Workflow-Management-Scripts.zip*.
2. Extract the zip file into the `/Oracle/Middleware/obpm` directory on the server where Oracle BPM is installed.

   Two directories are created from this zip file: **OBPM_SetupScripts** and **workflow**. It also creates a build.txt file.

A.3.4 Step 3: Configure the build.properties File

Enter the correct values for the `ORACLE_HOME\obpm\OBPM_SetupScripts\build.properties` file.

Table A–1 describes the properties listed in the `build.properties` files with descriptions and examples for each property. In the build.properties file, @ indicates a value that you should substitute. The scripts will use values from this property file.

> **Note:** Use forward slash characters to separate path elements regardless of platform.

The DBA user account must be specified in the `fuego.fdi.db.admin.id` setting of the `build.properties` file for the Ant tasks to correctly create the database schemas,

which will host the Oracle Enterprise Repository Workflows. This DBA User account is a new user which the Ant tasks create in the database.

Table A–1 build.properties File Property Values

Property	Description
bea.home	This property specifies the location of your Oracle Home directory. Example: c:/oracle or /Oracle/Middleware.
oer.uri	This property specifies the URI for your Oracle Enterprise Repository installation. Also, services/FlashlineRegistry should be present as part of the URI. Example: http://localhost:1111/oer/services/FlashlineRegistry
fuego.basedir	This property specifies the Oracle BPM installation location. Example: c:/oracle/obpm/enterprise or /opt/oracle/obpm/enterprise.
fuego.project	This property specifies the Oracle BPM project location. The following value is kept unchanged, if you followed the Oracle BPM install location, as recommended: `fuego.project=${basedir}/../workflow/aler_workflow` **Note**: {basedir} refers to the value of the `fuego.basedir` parameter in the `build.properties` file.
fuego.fdi.id	This is a Fuego directory property (FDI). Keep the value shown in the `build.properties` file. Example: `fuego.fdi.id=aler_fdi`.
fuego.fdi.organization	This is a Fuego directory property (FDI). Keep the value shown in the `build.properties` file. Example: `fuego.fdi.organization=BEA`.
fuego.fdi.admin.id	Create a new admin user for the Oracle BPM installation. This user account is used to access the Oracle BPM Webconsole application. Example: `oer_bpm_admin`
fuego.fdi.admin.password	Define a new admin password for the Oracle BPM installation. **Note**: The admin username and password should not be the same.
fuego.fdi.db.host	This property specifies the machine on which the Oracle BPM FDI (directory) database is to be installed. Example: `localhost`
fuego.fdi.db.port	This property specifies the Oracle BPM FDI database port.
fuego.fdi.db.admin.id	This property specifies the database admin user for the Oracle BPM FDI database. The installer uses this property to install the FDI schema (see below).
fuego.fdi.db.admin.password	This property specifies the database user's password for the Oracle BPM FDI database.
fuego.fdi.db.type	This property specifies the database type for the Oracle BPM FDI database. Example: oracle, mssqlserver, db2.
fuego.fdi.db.sid	This property specifies the SID for the Oracle BPM FDI database. This property is only applicable for the Oracle database type.

Table A–1 (Cont.) build.properties File Property Values

Property	Description
fuego.fdi.db.schema	This property specifies the database user name for the Oracle BPM FDI schema the database creates. This property is used by Oracle BPM at run time and is specific to Oracle database type. A new user database account will be created. Example: oer_bpm_fdi
fuego.fdi.db.schema.password	This property specifies the database password for the Oracle BPM FDI schema and is specific to Oracle database type.

fuego.server.db.host	These four properties define the Fuego engine properties for the database.
	This property specifies the system on which the Oracle BPM process engine database is to be installed.
	Example:
	`fuego.server.db.host=localhost`
	`fuego.server.db.port=1111`
	`fuego.server.db.admin.id=oer_bpm_engine`
	`fuego.server.db.admin.password=oracle`

fuego.server.db.port	This property specifies the Oracle BPM process engine database port.
fuego.server.db.admin.id	This property specifies the database admin user for the Oracle BPM process engine database. The installer uses this property to install the FDI schema (see below).
fuego.server.db.admin.password	This property specifies the database password user for the Oracle BPM process engine database.
fuego.server.db.type	This property specifies the database type for the Oracle BPM process engine database. Example: oracle, mssqlserver, db2.
fuego.server.db.sid	This property specifies the SID for the Oracle BPM process engine database. This property is only applicable for the Oracle database type. Example: XE
fuego.fdi.db.user.id	Only applicable for mssqlserver and db2 database types. This property specifies the database user name for the Oracle BPM FDI schema the database creates. This property is used by Oracle BPM at run time.
fuego.fdi.db.user.password	Only applicable for mssqlserver and db2 database types. This property specifies the database password for the Oracle BPM FDI schema.
fuego.fdi.db.database	Only applicable for mssqlserver and db2 database types. This property specifies the database name for the Oracle BPM FDI database.
fuego.server.db.user.id	Only applicable for mssqlserver and db2 database types. This property specifies the database user name for the process engine schema that is created. This property is used by Oracle BPM at run time.
fuego.server.db.user.password	Only applicable for mssqlserver and db2 database types. This property specifies the database password for the Oracle BPM process engine schema.

Table A–1 (Cont.) build.properties File Property Values

Property	Description
fuego.server.db.database	Only applicable for mssqlserver and db2 database types. This property specifies the database name for the Oracle BPM process engine database.
fuego.server.db.schema	This property specifies the database admin user name for the Oracle BPM process engine database. The installer uses this property to install the FDI schema for Oracle database.
fuego.server.db.schema.password	This property specifies the database password for the Oracle BPM process engine database for Oracle database type.

WARNING: Save a backup copy of your build.properties file. The build.properties file is deleted after you run Ant scripts.

A.3.5 Step 4: Configure the setenv File

Enter the correct values for the `ORACLE_HOME\obpm\OBPM_SetupScripts\setenv.bat` (Windows), or `/setenv.sh file` (Unix). Table A–2 describes the setenv file properties and its descriptions.

Note: Use the forward slash character (/) to separate path elements.

Tip: If you are running the setenv script in a Linux/Unix environment, run the dos2unix command for this script after editing it with the correct property values.

Table A–2 setenv File Property Values

Property	Description
`BEA_HOME`	This property specifies the location of the Oracle Home directory. Example: `/Oracle/Middleware`.
`OBPM_INSTALL_DIR`	This property specifies the parent directory where Oracle BPM is installed. Example: `BEA_HOME/obpm`.
`FUEGO_HOME`	This property specifies the location where Oracle BPM is installed. Example: `BEA_HOME/obpm/enterprise`.
`JAVA_HOME`	This property specifies the location of a Java runtime on the system. Example: `ORACLE_HOME/obpm/j2eewl/jre`.
`ANT_HOME`	This property specifies the location of a Apache ANT installation on your machine. Example: `C:/oracle/modules/org.apache.ant-1.6.5` or `/opt/oracle/modules/org.apache.ant-1.6.5`

A.3.6 Step 5: Generate (or Regenerate) the workflow.xml File

You have to generate a new workflow.xml file which you will edit with new user credentials. If you create new asset types or add users (*alerid* are users), you have to regenerate the workflow in order for workflow.xml to capture these changes. Regeneration also updates the OER URI in the workflow.xml file.

> **Note:** To run, you must have OER up and running.

You can generate the workflow.xml file using the Generate Workflow Config tool (config_gen.bat). This tool connects to Oracle Enterprise Repository and creates a bootstrapping file that can be customized.

Oracle delivers a workflow.xml file. Make a backup copy of this file. Navigate to the directory where you installed OER, run the script, and copy the generated script into the Setupscripts folder.

To generate or regenerate the workflow.xml file:

1. From a command prompt, run the Generate Workflow Config tool found in repository111/core/workflow-tools/ as follows:

   ```
   > config_gen.bat URI User Password ConfigDir
   ```

   where

   - URI = OER URI, using the following format:

     http://<host>:<port>/<oer web app name>/services/FlashlineRegistry

     For example: *http://localhost:7001/alerbuild/services/FlashlineRegistry*

   - User = Oracle Enterprise Repository user name
   - Password = Oracle Enterprise Repository password
   - ConfigDir = the directory where the **workflow.xml** file is created

2. Copy the newly generated workflow.xml file to the *<Oracle Business Process Management Enterprise Edition>/enterprise/server/aler_engine* directory.
3. Open the **workflow.xml** file using the XML editor of choice.

For more information about generating the workflow.xml file, see Section A.8.4, "Configuring Workflow".

A.3.7 Step 6: Edit the workflow.xml File

The first editing pass gets the right passwords and url into the file. You will configure the workflow file later to suit your organization's needs.

1. Edit the `workflow.xml` file located within the `ORACLE_HOME/obpm/OBPM_SetupScripts/workflow.xml` and modify the URI to match the URI of the Oracle Enterprise Repository installation.
2. Retain the `/services/FlashlineRegistry` at the end of the path reference.
3. Modify all the user credentials in the `workflow.xml` file to their correct values.

> **Note:** Ensure that you use the same JDK path that you had set while updating the BPM JDK to 1.6. For more information, see Section A.3.2, "Step 1: Install Oracle BPM 10.3.2".

A.3.8 Step 7: Encrypt the workflow.xml File

To encrypt the Oracle Enterprise Repository Workflow configuration file, workflow.xml:

1. Go to http://<host>:<port>/oer/diag/encryptstrings.jsp.
2. Run encryption on the workflow.xml file.
3. Copy the `workflow.xml` file into `/obpm/enterprise/server/aler_engine`.

For an alternative method of encrypting the workflow.xml file, see Section 5.3.3, "Workflow Configuration File".

A.3.9 Step 8: Copy the JDBC jar(s)

Copy the JDBC jar(s) for your database to the `$Oracle_HOME\OBPM_SetupScripts\ext` directory.

A.3.10 Step 9: Verify Permissions

Verify the permissions for the OBPM_SetupScripts and workflow folders to ensure they are writable.

A.3.11 Step 10: Run the Setup Script

> **Tip:** When the ANT scripts are run successfully, the build.properties file is deleted due to security concerns with passwords embedded in that file. You cannot prevent this, so the workaround is to take backups. The `fuego.fdi.admin.id` and `fuego.fdi.admin.password` properties must be recorded for future login to the Oracle BPM Web console application.

If you are using setenv.sh, you may need to run a dos2unix command to ensure that there are no DOS characters in the file. Any DOS in the file will cause a problem.

You may need to run setenv.sh in a BASH shell with the following syntax: . ./setenv.sh Verify that ANT_home is set.

To run the `setenv.bat` or `setenv.sh` file:

1. Open a command, or shell window.
2. Navigate to the `$Oracle_HOME\OBPM_SetupScripts` directory.
3. Run the following command:

   setenv.bat (Windows)

   setenv.sh (Unix) To use a bash shell, run . ./setenv.sh. Verify that ANT home is set.

   You execute `ant create-fdi` to initialize the OBPM database and then execute `ant install-workflow` to initialize the OBPM workflow engine (named `aler_engine`).
4. Run the following command:

   ```
   ant create-fdi
   ```
5. Run the following command:

   ```
   ant install-workflow
   ```

Do you need to regenerate the workflow here? Edit it?

A.3.12 Step 11: Verify the Setup Script

To validate the success of the setup script:

1. In the Enterprise/bin directory, launch obpmadmcenter.

   Open the Oracle BPM Admin Center application.

2. Click **Start BPM Applications**.

3. Click **Launch Process Administrator**.

4. Access the Web console Web interface and login using the credentials for the Oracle BPM Web console application.

   These credentials are specified in the `build.properties` file in the **fuego.fdi.admin.id** and **fuego.fdi.admin.password** properties.

5. Select the **Engines** link in the left side menu.

   The `aler_engine` engine is displayed with a status of **Not Running**.

6. Start the engine by clicking the left most icon in the Engine Actions column.

   Once the engine is running, the status of the engine displayed is **Ready**.

7. Validate that the Oracle BPM service endpoint is listening appropriately. Use a Web browser to connect to the Oracle BPM server at port 9000, for example, *http://localhost:9000/albpmServices/aler_engine/ws*.

8. Click the link for **albpmServices/oer_engine/ws** link.

   Two services are then listed, **StatusChangeEndpointServiceListener** and **RefreshConfigServiceListener**.

Tip: If any build failure or errors appear during the ANT deployment of the workflows, follow these steps:

1. Drop the FDI Schema user (as specified for fuego.fdi.db.schema in build.properties file) from Database.
2. Drop the FDI Engine user (as specified for fuego.server.db.schema in build.properties file) from Database.
3. Make sure the BPM Applications are not running and obpmadmcenter.exe is closed.
4. Run **setenv.bat** or **setenv.sh**.
5. Run the **ant create-fdi** command.
6. Run the **ant install-workflow** command.

Click the Oracle Enterprise Repository installation within the `WEB-INF/classes/EndPointEventSubscription.xml` file and modify the **<sub:host>** element to contain the IP address or the fully qualified host name of the server where Oracle BPM is installed.

Oracle Enterprise Repository workflows are now deployed.

To run workflows successfully, you must encrypt passwords in the xml file and then restart the Oracle Enterprise Repository server. If the `EndPointEventSubscription.xml` has clear text password, then the events do not get delivered to Oracle BPM.

If the password is not encrypted, then the following log message is displayed in the *<ORACLE_HOME>\user_projects\domains\<oer_domain>\eventing.log* file:

Insufficient subscription data. End point with name [<ENDPOINT_NAME>] requires an encrypted password...

A.4 Configuring the Oracle Enterprise Repository Event Manager

This section discusses the Event Manager configuration that must be completed before using the Automated Workflows.

This section contains the following topics:

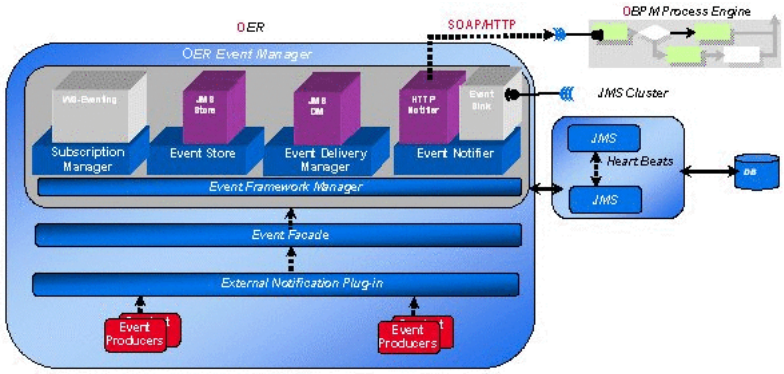
- Section A.4.1, "What is the Oracle Enterprise Repository Event Manager?"
- Section A.4.4, "Configuring the Event Manager's System Settings"
- Section A.4.5, "Configuring the Subscription Manager"
- Section A.4.6, "Configuring Logging of Event Manager Events"

A.4.1 What is the Oracle Enterprise Repository Event Manager?

The Event Manager is a component embedded within Oracle Enterprise Repository that manages event subscriptions, event persistence, event filtering, and event delivery. Web Service endpoints can subscribe to the Event Manager's Subscription Manager and the asset registration events that are generated within Oracle Enterprise Repository are delivered to the Web Service endpoints.

Figure A–3 shows the different components that are involved.

Figure A–3 Automated Workflow Components



The Event Manager uses an embedded version of Apache ActiveMQ JMS Server that is enabled by default. The embedded JMS server is configured to run out-of-the-box without any additional configuration. However, you can also configure the Event Manager to use an external JMS server, such as WebLogic Server or IBM WebSphere.

This section discusses the Event Manager configuration that needs to be completed before using the Automated Workflows.

For information on configuring the Automated Workflows, see Section A.4, "Configuring the Oracle Enterprise Repository Event Manager".

A.4.2 Configuring the Event Manager

To configure the Event Manager:

1. The Event Manager must be enabled in Oracle Enterprise Repository to allow the Event Manager to send events to external Web Service endpoints. You can either:
   - Enable the `cmee.eventframework.enabled`=true property in the eventing.properties file in the `<OER Domain>\WEB-INF\classes` directory.

     or
   - Enable using the Oracle Enterprise Repository Web-based console's System Settings, as explained in "Configuring the Event Manager's System Settings" on page 3-2.
2. The default Eventing `cmee.eventframework.delivery.sleep` and `cmee.eventframework.store.sleep` property values can also be tuned to control the overall performance of Oracle Enterprise Repository and the Web Service endpoints. These properties directly impact the number of events that get triggered per second by the Event Manager. For example, if a faster response is required for testing purposes, the default `cmee.eventframework.store.sleep` value of 7200 seconds should be changed to a reasonable testing amount.
3. The Event Manager uses the same logging framework as Oracle Enterprise Repository. By default, logging is enabled to go to a file, but you direct the debug statements to go to the console by appending the following categories to the `log4fl.properties` file in the `<OER Domain>\WEB-INF\classes` directory.

Example A–1 Directing Dbug Statements to the Console

```
# eventing subsystem
log4j.category.com.bea.infra.event.core= debug,eventingLog,stdout
log4j.category.com.bea.infra.event.dm= debug,eventingLog,stdout
log4j.category.com.bea.infra.event.facade= debug,eventingLog,stdout
log4j.category.com.bea.infra.event.notifier= debug,eventingLog,stdout
log4j.category.com.bea.infra.event.store= debug,eventingLog,stdout
log4j.category.com.bea.infra.event.sub= debug,eventingLog,stdout
```

4. Configure the Web Service subscriptions with the Event Manager's Subscription Manager.

   Note: As delivered the Subscription Manager is configured to work with the Oracle Business Process Management Process Engine if the Oracle Business Process Management Process Engine is running on the same machine as Oracle Enterprise Repository. You can skip this step if this is the case because the default settings are ready to run.

   You may need to change the following information within the EndPointEventSubscription.xml file under `<OER Domain>\WEB-INF\classes` directory, depending on your requirements:
   - **Host**: If the Web Service endpoint is running in a host other than Oracle Enterprise Repository.

     If it is the same host, then leave the default unchanged.
   - **Port**: Specify the port of the Web Service endpoint.

     If Oracle Business Process Management is used as the Process Engine, then leave the default unchanged.

- **URI**: Specify the URI of the Web Service.

  If Oracle Business Process Management is used as the Process Engine, then leave the default unchanged.

- **Operation Name**: If Oracle Business Process Management is used as the Process Engine, leave the default unchanged. Refer to the WSDL within the **modules.eventing-11.1.1.x.x.jar** file located in *<OER Webapp path>/WEB-INF/lib* for the available operations.

- **User Name/Password**: Used only if Oracle Business Process Management is used as the Process Engine.

  The default user name/password for OBPM is aler_workflow_user/*<encrypted_password>*. The default password text used is `aler_workflow_pass`.

- **Expression**: Default is empty, which means all the events are delivered.

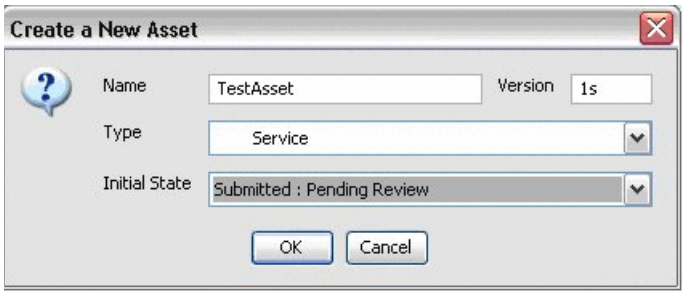
5. Restart Oracle Enterprise Repository for the configuration changes to take effect.

A.4.3 Triggering an Asset Event

To confirm that events are triggered after configuring the Event Manager:

1. Launch the Oracle Enterprise Repository Asset Editor from the Web-based console. For information on using the Oracle Enterprise Repository Asset Editor, see *Oracle Fusion Middleware User's Guide for Oracle Enterprise Repository*.
2. Create a new asset, as shown in Figure A–4.

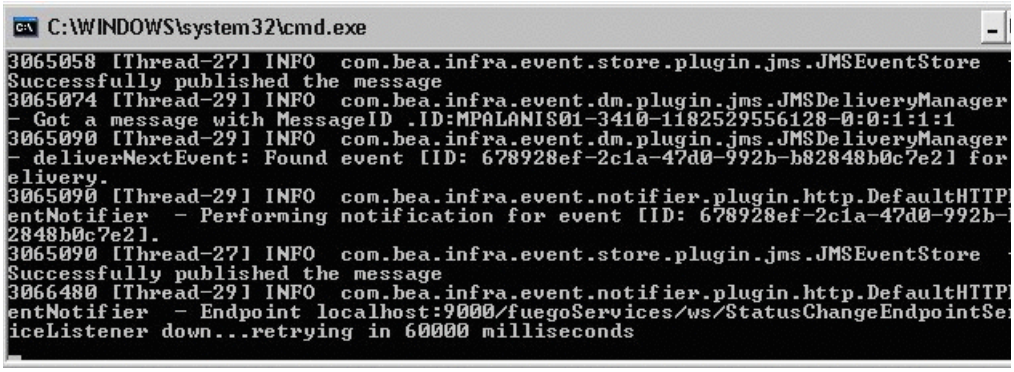
Figure A–4 Oracle Enterprise Repository Asset Editor - Create New Asset



Note: The Asset Type should be *Service*.

3. Click **OK** to submit the asset.
4. After the asset is submitted, switch to the Oracle Enterprise Repository console to verify the following logging statements printed to the console, as shown in Figure A–5.

Figure A–5 Event Monitoring Console



5. The Event Monitoring tool can be used to view the payload of the event that is delivered. For more information about how to monitor and manage events, see Section A.8, "Monitoring and Managing Events".

The `HarvesterSettings.xml` file can be optionally configured to set the `<triggerEvent>` tag turned to true or false so as to turn the BPM workflow eventing enabled or disabled respectively for the harvester tasks.

A.4.4 Configuring the Event Manager's System Settings

You use Oracle Enterprise Repository's system settings section to configure the basic Oracle Enterprise Repository operation and to enable or disable specific features. The Event Manager-related settings are under the "Eventing" group under the main "External Integrations" category.

For more information about system settings, see Section 17.5.5, "External Integration Settings".

Additional "Eventing" properties are available for configuring an external JMS server, such as WebLogic Server and IBM WebSphere, and are described in Section A.7, "Configuring JMS Servers for Oracle Enterprise Repository".

A.4.4.1 Enabling the Event Manager

The Event Manager must be enabled in Oracle Enterprise Repository to allow the Event Manager to send events to external Web Service endpoints.

1. Click **System Settings** in the sidebar on the Oracle Enterprise Repository Admin screen.
2. Enter *Event* in the System Settings Search box 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.

A.4.4.2 Configuring Optional Event Manager Settings

The following sections describe some optional "Eventing" properties that you can use to tune the Event Manager performance.

Note: You must restart Oracle Enterprise Repository after changing any Eventing property in order for the changes to take effect.

This section contains the following topics:

- Section A.4.4.2.1, "Eventing Manager Notifier Thread Sleep (seconds)"
- Section A.4.4.2.2, "Eventing Manager Store Thread Sleep (seconds)"
- Section A.4.4.2.3, "Eventing Manager Store Delivery Sleep (seconds)"
- Section A.4.4.2.4, "Batch Size for Event Manager Deliveries"

A.4.4.2.1 Eventing Manager Notifier Thread Sleep (seconds) If an endpoint is not available when one or more events should be delivered to that endpoint, the Event Manager notifier will retry delivering the event until the endpoint is available. The `cmee.eventframework.notifier.sleep` property configures in seconds how long the notifier should wait before trying to redeliver an event.

A.4.4.2.2 Eventing Manager Store Thread Sleep (seconds) As soon as an event is triggered, the Event Manager stores the event in memory before pushing it to the JMS server so that the Oracle Enterprise Repository thread is not blocked. The `cmee.eventframework.store.sleep` property specifies in seconds how long the Event Manager's Store Manager thread should sleep before polling for the next available batch of events stored in memory. The default polling delay is 16 seconds.

A.4.4.2.3 Eventing Manager Store Delivery Sleep (seconds) By default, the Event Manager delivers events in batches. The `cmee.eventframework.delivery.sleep` property specifies in seconds how long the Event Manager's Delivery Manager thread should sleep before selecting the next available event from the JMS server.The default delay between each batch is 13.

Tip: The default cmee.eventframework.store.sleep and `cmee.eventframework.delivery.sleep` property values can be tuned to control the overall performance of Oracle Enterprise Repository and the Web Service endpoints. These properties directly impact the number of events that get triggered per second by the Event Manager. For example, if a faster response is required for testing purposes, the default `cmee.eventframework.delivery.sleep` value of 13 seconds should be changed to a reasonable testing amount, if needed.

A.4.4.2.4 Batch Size for Event Manager Deliveries When the Event Manager delivers events in batches, the delivered batch size can be configured using the `cmee.eventframework.delivery.batch.size` property. The default batch size is 100 events. If the Event Manager finds fewer events to deliver, it will deliver the available events and then sleep for the time configured in the `cmee.eventframework.delivery.sleep` property.

A.4.5 Configuring the Subscription Manager

The Subscription Manager is responsible for managing the event subscriptions by the Web Service endpoints where the matched events are delivered.

The Subscription Manager configuration file is located in `<oer webapp name>\WEB-INF\classes\EndPointEventSubscription.xml`.

A.4.5.1 Configuring Web Service Endpoints

The Event Manager uses the EndPointEventSubscription.xml file to load information about the Web Service endpoints where events need to be delivered. The host, port, URI, user, and password of the predefined ALPBM endpoint, or user-defined Web Service endpoint, must be configured, as shown in Example A–2.

Example A–2 Web Service Endpoint Example

```
<sub:EventSubscriptionData
xmlns:sub="http://www.bea.com/infra/events/subscription"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
<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="ALBPMEndpoint">
<!--The host of the Webservice Endpoint -->
<sub:host>localhost</sub:host>
--The port of the Webservice Endpoint -->
<sub:port>9000</sub:port>
<!--The URI of the Webservice Endpoint -->
<!--If you are using ALBPM5.7 uncomment the following line and comment the line
 after -->
<!-- <sub:uri>fuegoServices/ws/StatusChangeEndpointServiceListener</sub:uri> -->
<sub:uri>albpmServices/aler_
engine/ws/StatusChangeEndpointServiceListener</sub:uri>
<!--Unless a custom WSDL Contract is used, the namepsace 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\modules.eventing-11.1.1.x.x.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 Webservice Endpoint.  -->
<sub:protocol>HTTP</sub:protocol>
<!--The user and password to authenticate the ALBPM Webservice. -->
<sub:authenticationData>
<sub:basicAuthentication>
<sub:username>aler_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 dafault is all the events are delivered. -->
<!--Example:- asset_id BETWEEN 50000 AND 50100 -->
<sub:expression></sub:expression>
</sub:eventSubscription>
</sub:EventSubscriptionData>
```

You can add as many endpoints as needed and the endpoints can be located in different hosts or ports and the events can be load balanced. The pre-defined Advanced Registration Flow has just one endpoint called "**StatusChangeEndpoint**".

A.4.5.2 Setting the Expression to Filter Events

Events can be filtered based on the value entered in the expression element. This section contains the following topics:

- Section A.4.5.2.1, "Delivering all Events to an Endpoint"
- Section A.4.5.2.2, "Delivering Events to an Endpoint Filtered by Event Type"
- Section A.4.5.2.3, "Delivering Events to an Endpoint Filtered Using a JMS Message Selector"
- Section A.4.5.2.4, "JMS Message Selector Examples"

A.4.5.2.1 Delivering all Events to an Endpoint The default setting is to deliver all events to an endpoint. All the events that are triggered within Oracle Enterprise Repository are delivered to the OOTB endpoint when the expression element is empty.

```
<sub:expression></sub:expression>
```

A.4.5.2.2 Delivering Events to an Endpoint Filtered by Event Type The following XML snippet shows how to deliver an event of type AssetSubmission to an endpoint:

```
<sub:expression> eventdata_name
='urn:com:bea.aler:events:type:AssetSubmission'</sub:expression>
```

You can also use the "OR" operator to filter more than one event type:

```
eventdata_name ='urn:com:bea.aler:events:type:AssetSubmission'
OR
eventdata_name ='urn:com:bea.aler:events:type:AssetAccepted'
```

These event types are supported:

- urn:com:bea:aler:events:type:AssetSubmission
- urn:com:bea:aler:events:type:AssetAccepted
- 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:PolicyAssertionChanged
- urn:com:bea:aler:events:type:MetaDataChange:name
- urn:com:bea:aler:events:type:AssetUnSubmission
- urn:com:bea:aler:events:type:AssetUnAccept
- urn:com:bea:aler:events:type:AssetUnregister
- urn:com:bea:aler:events:type:AssetStatusChanged
- urn:com:bea:aler:events:type:MetaDataChange:version
- urn:com:bea:aler:events:type:MetaDataChange:description
- urn:com:bea:aler:events:type:CategorizationChanged:assetLifecycleStage
- urn:com:bea:aler:events:type:CategorizationChanged:classification
- urn:com:bea:aler:events:type:MetaDataChange:supported
- urn:com:bea:aler:events:type:MetaDataChange:organizational ownership

- `urn:com:bea:aler:events:type:MetaDataChange:usagefee`

A.4.5.2.3 Delivering Events to an Endpoint Filtered Using a JMS Message Selector

Selectors are a way of attaching a filter to a subscription to perform content-based routing. Selectors are defined using SQL 92 syntax. Example A–3 is a complete list of fields that can be used to write a filter expression to filter the events. These fields are added to the JMS message as properties by the Event Manager and a JMS Message Selector that accesses the fields can be written to filter the events.

Example A–3 Fields Used to Write a Filter Expression

```
submittedby_emailaddress = mrsmith@bea.com
asset_description = Test Asset
submittedby_name = aler_workflow_user
submittedby_id = 99
asset_community = Java
eventdata_description = new aler event
eventsource_componentname = OER
asset_name = TestAsset
eventsource_componenttype = OER 10.3
asset_typeid = 154
eventdata_eventid = d0cdac55-c78f-4a29-8aec-6ea9ba8d31f1
eventdata_name = urn:com:bea:aler:events:type:MetaDataChange:name
asset_activestatus = ACTIVE
eventsource_location = ALERCore
asset_id = 50100
eventdata_version = ver1.0
asset_version = 1
```

For more information about JMS Message Selectors, see the following web sites:

- *http://java.sun.com/j2ee/1.4/docs/api/javax/jms/Message.html*
- *http://activemq.apache.org/selectors.html*

A.4.5.2.4 JMS Message Selector Examples

Here are some sample usages of JMS message selectors:

- `asset_id` BETWEEN 50000 AND 50100
- `eventdata_name` = 'urn:com:bea:aler:events:type:AssetSubmission' AND asset_id BETWEEN 50000 AND 50100
- `asset_name` LIKE 'Inventory'
- `asset_id` > 500

> **Tip:** Symbols, such as "< >" used for less than/greater than, are not valid XML content. Because the expression is written in an XML file and parsed by the Event Manager, the XML unfriendly characters need to be managed using the XML Rules. For example, you must use "id > 500", which is equivalent to "asset_id > 500".

A.4.6 Configuring Logging of Event Manager Events

The Event Manager uses the same logging framework as Oracle Enterprise Repository. By default, logging is enabled to go to a file, but you direct the debug statements to go to the console by appending the categories shown in Example A–4 to the **log4fl.properties** file in the *<OER Domain>\WEB-INF\classes* directory.

Example A–4 Configuring Logging of Event Manager Events

```
# eventing subsystem
log4j.category.com.bea.infra.event.core= debug,eventingLog,stdout
log4j.category.com.bea.infra.event.dm= debug,eventingLog,stdout
log4j.category.com.bea.infra.event.facade= debug,eventingLog,stdout
log4j.category.com.bea.infra.event.notifier= debug,eventingLog,stdout
log4j.category.com.bea.infra.event.store= debug,eventingLog,stdout
log4j.category.com.bea.infra.event.sub= debug,eventingLog,stdout
```

A.5 Administering Oracle Business Process Management Processes

This section discusses the administration tasks for Oracle Business Process Management processes.

This section contains the following topics:

- Section A.5.1, "Overview"
- Section A.5.2, "Administering Oracle Business Process Management Web Applications"
- Section A.5.3, "Tuning the Oracle Business Process Management Automated Workflows Engine"
- Section A.5.4, "Configuring a Standalone Process Engine for Failover"
- Section A.5.5, "Using the Oracle Business Process Management Log Viewer"

A.5.1 Overview

Once the Event Manager is ready to send events, the Process Engine must be configured to process the events. When Oracle Enterprise Repository is installed, it provides an option to install and configure the Oracle Business Process Management Process Engine. This section assumes that the Oracle Business Process Management Process Engine was successfully installed.

A.5.2 Administering Oracle Business Process Management Web Applications

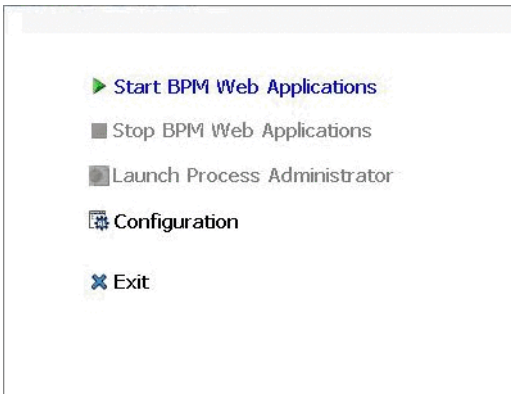
To start the Oracle Business Process Management Process engine and define the participants, you must launch the Oracle Business Process Management Admin Center.

A.5.2.1 Starting the Oracle Business Process Management Admin Center

Follow these steps to launch the Oracle Business Process Management Admin Center:

1. Navigate to the *<ORACLE_HOME>\obpm\enterprise\bin* directory and double-click one of the following files:
   - **obpmadmcenter.exe** (Windows or UNIX GUI-based)
   - `./startwebconsole.sh` (UNIX console-based). Then point your browser to *http://<host>:8585/webconsole* (for example, *http://localhost:8585/webconsole*).
2. On the Admin Center page, click the **Start BPM Web Applications** option, as shown in Figure A–6.

Figure A–6 Oracle Business Process Management Admin Center



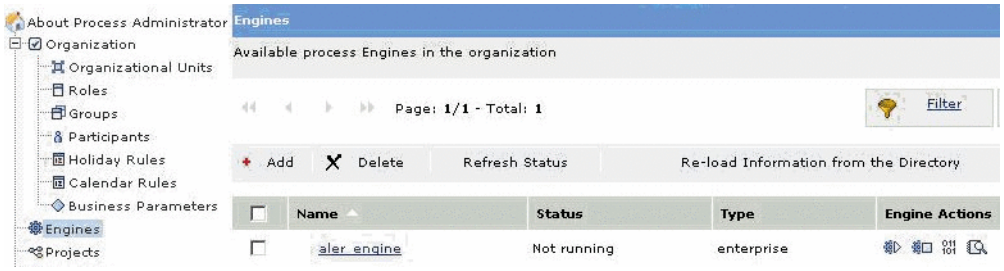
3. When it becomes available, click the **Launch Process Administrator** option to launch the Process Administrator.
4. When prompted to enter the required credentials, enter the BPM admin user name and password that was used on the FDI User Credentials panel during the installation process. The recommended example for these credentials is bpm_admin for the user name and password.

A.5.2.2 Starting the Oracle Business Process Management Process Engine

Follow these steps to start the Oracle Business Process Management Process Engine.

1. On the Oracle Business Process Management Process Administrator page, open the `aler_engine` Process Engine by clicking the **Engine** link on the left side of the page, as shown in Figure A–7.

Figure A–7 Oracle Business Process Management Process Administrator - Start / Stop



2. Start the `aler_engine` by clicking the Start icon under Engine Actions on the right side of the page. Starting the engine may take several minutes to complete. Make sure that the status of the engine is Ready.

Once you the Oracle Business Process Management Process Engine is running, you can stop it and then restart it to load your latest workflow.xml changes.

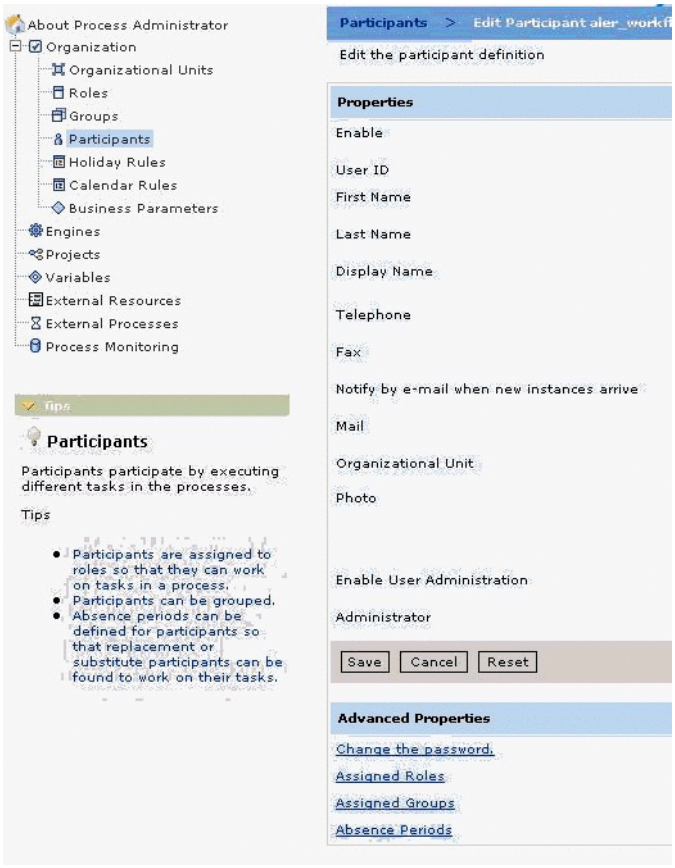
A.5.2.3 Defining the Oracle Business Process Management Participants

This section explains how to define the Oracle Business Process Management Process Engine participants.

A.5.2.3.1 Oracle Business Process Management Administrators Using the FDI User Credentials, Oracle Business Process Management Process Administrator can log into the Process Administrator, start/stop the process engine, and create other users.

A.5.2.3.2 Advanced Registration Flow Participant When the Oracle Business Process Management Process Engine is installed by Oracle's Products installer, it creates `aler_workflow_user` as the Advanced Registration Flow user. By default, the password is set as `aler_workflow_pass`, but the password can be changed in the Process Administrator by selecting Participants in the navigator and clicking **Change the password** in the Advanced Properties section, as shown in Figure A–8.

Figure A–8 Oracle Business Process Management Process Administrator - Change Password



A new participant can also be created for the role of "administrator" and this new participant can be configured in the Event Manager's Subscription Manager file. For more information, see Section A.4.5, "Configuring the Subscription Manager".

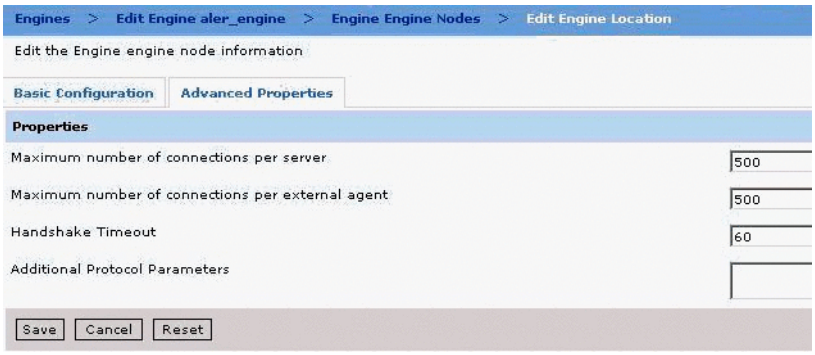
A.5.3 Tuning the Oracle Business Process Management Automated Workflows Engine

The following parameters need to be tuned using the Oracle Business Process Management Process Administrator.

A.5.3.1 Advanced Properties

Go to the Engines, <Engine Name>, Engine Nodes, Advanced Properties page. Figure A–9 illustrates the Oracle Business Process Management Process Administrator - Advanced Properties page.

Figure A–9 Oracle Business Process Management Process Administrator - Advanced Properties



A.5.3.2 Database Runtime Properties

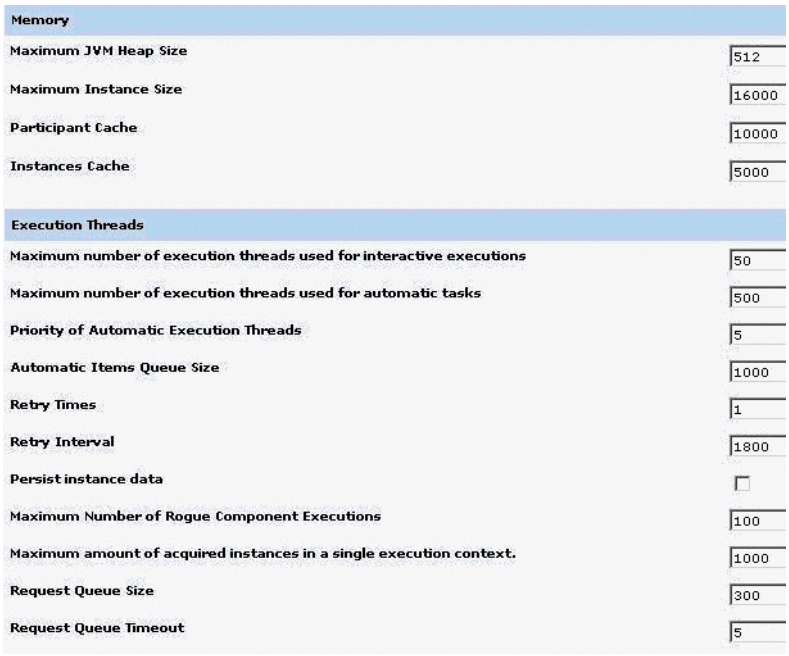
Go to the Engines, <Engine Name>, Edit Engine Database Configuration page. Figure A–10 illustrates the Oracle Business Process Management Process Administrator - Database Runtime page.

Figure A–10 Oracle Business Process Management Process Administrator - Database Runtime

A.5.3.3 Memory and Execution Thread Properties

Go to the Engines, <Engine Name>, Execution page. Figure A–11 illustrates the Oracle Business Process Management Process Administrator - Memory and Threads page.

Figure A–11 Oracle Business Process Management Process Administrator - Memory and Threads



A.5.4 Configuring a Standalone Process Engine for Failover

To support failover of Oracle Business Process Management standalone process engines, you can configure a backup engine(s) in your environment. One of the engines in this federation is marked as PRIMARY and the others are backups for this primary engine. Multiple engines can be configured to serve as backups. Any of these backup engines will take the role of the primary if the designated primary fails. When the server that has failed comes back online, it will join in as a backup to the one acting as primary.

For detailed instructions on configuring backup engines, see the section on configuring engine failover in the Oracle Business Process Management Administration Guide at:

http://download.oracle.com/docs/cd/E13165_01/bpm/docs65/admin_guide/index.html

A.5.5 Using the Oracle Business Process Management Log Viewer

The Oracle Business Process Management Log Viewer enables you to read information logged by the Process Execution Engine. A set of log files is created for each project you define. The Studio Log Viewer reads the files and displays them to help you monitor and trace Engine execution.

To launch the Log Viewer, double-click the **obpmlogviewer.exe** file in the `<Oracle Business Process Management Enterprise Home>\bin` directory.

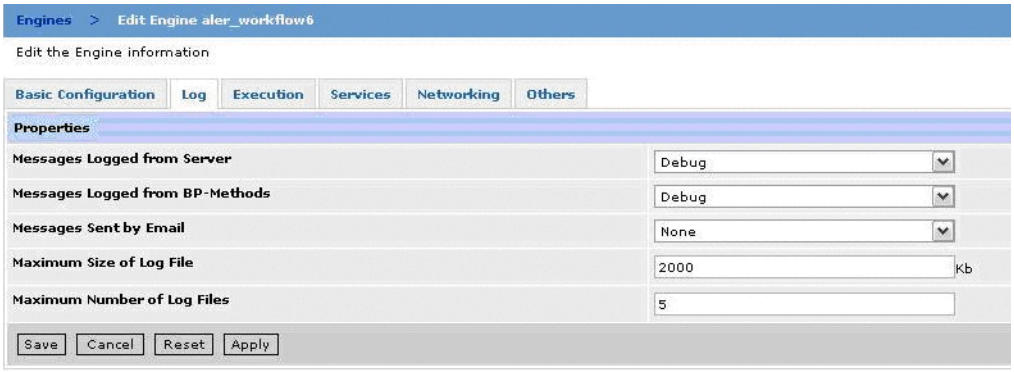
A.5.5.1 Filtering Event Log Messages for Oracle Enterprise Repository Flows

You can filter log messages so that the Automated Workflows log Info, Debug, and Fatal messages.

Turn on the "Debug" level on the Log page of the Process Engine using the Process Administrator preference settings. By default, the level is set to "Warning".

Go to the Engines, <Engine Name>, Log page. Figure A–12 illustrates the Oracle Business Process Management Process Administrator - Logging Preferences page.

Figure A–12 Oracle Business Process Management Process Administrator - Logging Preferences

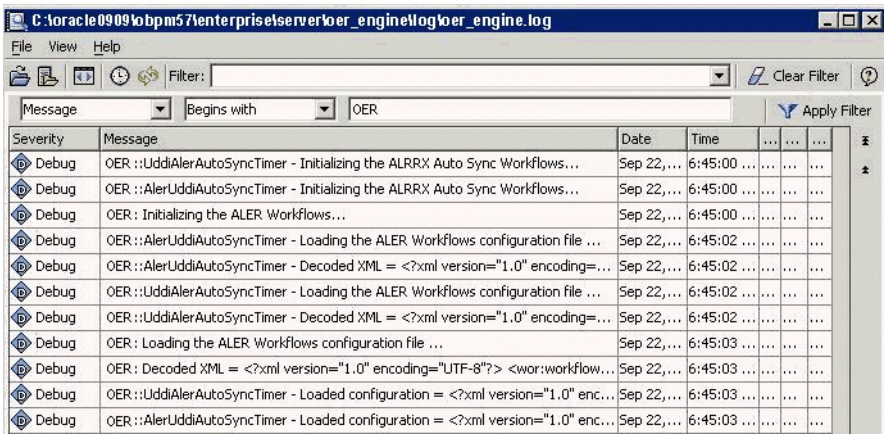


When you turn on the Debug level though you will notice that the Process Engine prints a lot of information, not just for the Oracle Enterprise Repository Automated Workflows, but other Process Engine information as well. To filter the debug logging to show only the Oracle Enterprise Repository flow-related information, follow these steps, as shown in Figure A–13:

1. Within the Log viewer, select **Message** in the left-most list box.
2. Select **Begins With** in the next list box.
3. Type *OER*: in the text box.
4. Click the **Apply Filter** button.

The Oracle Enterprise Repository Event Logging prints a prefix of ALER: for all logged event messages, as shown here.

Figure A–13 Log Viewer With OER Filter



A.6 Configuring Automated Workflows

This section describes how to get started with OER automated workflows and explains the use of workflows in various high-level use cases.

This section contains the following topics:

- Section A.6.1, "Overview of Automated Workflows"
- Section A.6.2, "Creating and Customizing a Workflow Configuration File"
- Section A.6.3, "Wiring Asset Events to Flows"
- Section A.6.4, "Automatic Asset Registration Flows"
- Section A.6.5, "Multi-tier Automatic Assignment Flows"
- Section A.6.6, "Metadata Change Flows"
- Section A.6.7, "Time-based Escalation Flows"
- Section A.6.8, "Validation Expiration Flows"
- Section A.6.9, "Customizing Email Notification Templates"
- Section A.6.10, "Email Notification Use Cases"
- Section A.6.11, "Known Issues"

A.6.1 Overview of Automated Workflows

> **Tip:** Before you begin, read Section A.2, "Introduction to Oracle Enterprise Repository Automated Workflows" to quickly get started using the Advanced Registration Flow feature using the bundled Oracle Business Process Management Web Service endpoint that is configured to work with the Oracle Business Process Management Process Engine.

Oracle Enterprise Repository bundles pre-built Oracle Business Process Management flows that attempt to automate Oracle Enterprise Repository asset submission, acceptance, registration and other governance process. This section discusses the configuration that is required before starting the Oracle Business Process Management Process Engine to process the asset events that are triggered by Oracle Enterprise Repository. For more information about configuring the Process Engine to trigger flows, see Section A.4, "Configuring the Oracle Enterprise Repository Event Manager".

The flows are also designed to be flexible and can be customized using either the Workflow Configuration file (workflow.xml) or Oracle Business Process Management. This section also discusses each flow in detail and gives examples of how they can be tailored to suit your environment.

This section describes how to configure an advanced registration flow. To create a new workflow, follow these steps:

1. Open an existing Oracle Business Process Management project in an IDE.
2. Add a new workflow. See the Oracle Business Process Management documentation for details on this step.
3. Undeploy and deploy the project.
4. Wire the workflow to the events by following the instructions in Section A.6.3, "Wiring Asset Events to Flows".

A.6.2 Creating and Customizing a Workflow Configuration File

This section explains how to create and customize a Workflow Configuration XML file.

A.6.2.1 Defining the Oracle Enterprise Repository Connection and Registrar

The Workflow Configuration file loads the Oracle Enterprise Repository connection and registrar information from the following XML data.

```
<alerconnection>
  <uri>http://localhost.7001/aler/services/FlashlineRegistry</uri>
  <registrar>
    <user>admin</user>
    <password>******</password>
  </registrar>
</alerconnection>
```

A.6.2.2 Encrypting the Registrar User Password

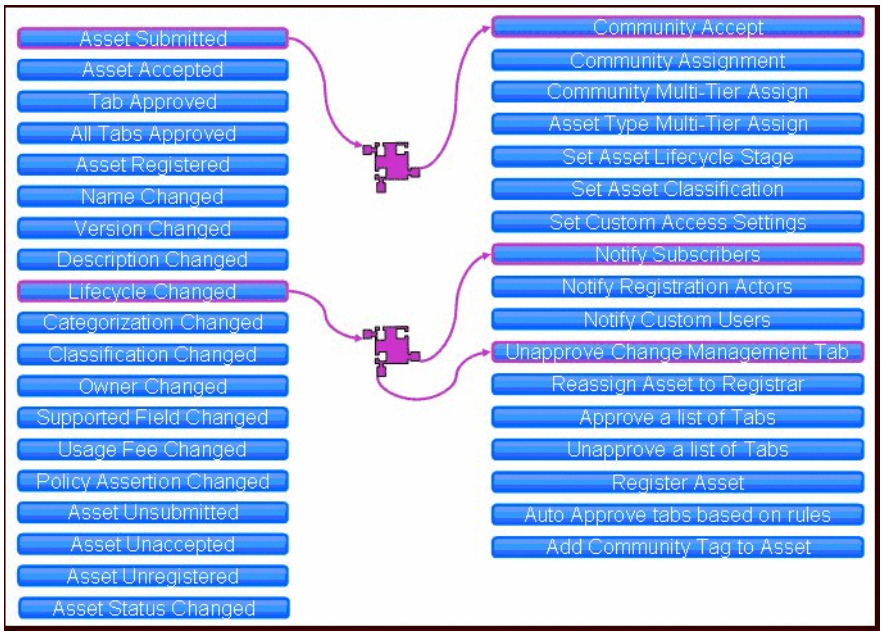
The Security Encrypt Password tool (**runWfSecurity.bat**) allows you to encrypt the registrar passwords that are stored in the Workflow Config file. The tool recursively scans the file and encrypts all the password elements it encounters.

For more information, see Section A.8.5, "Encrypting Your Passwords".

A.6.3 Wiring Asset Events to Flows

The Automated Workflows are designed with a flexible framework where asset events can be wired to one or more flows that are executed when an event is triggered, as illustrated in Figure A–14.

Figure A–14 Wiring Asset Events to Flows



Note:

- All the events are wired to pre-defined flows out-of-the-box. The wirings only need to be changed if customizations or new flows are designed.
- This diagram does not include all the events and actions. The list of all the actions or events is described in Section A.6.6.2.1, "Available Metadata Change Events/States" and Section A.6.6.2.2, "Available Flows That Can Be Wired to Actions".
- For more information about community workflows, see Section A.6.4.1, "Configuring Community Flows".

The wiring of asset events to flows is configured within the Workflow Configuration file. For example, the following configuration snippet shows that when an "Asset Submitted" event is triggered, it in turn triggers a flow to automatically accept the asset based on rules that are configured in the Workflow Configuration file.

```
<!--Community Flows-->
<state name="urn:com:bea:aler:events:type:AssetSubmission">
  <action>CommunityAccept</action>
</state>
<!--The Multi_tier Flows-->
<state name="urn:com:bea:aler:events:type:AssetAccepted">
  <action>MultiTier_Tier1_Assign</action>
</state>
<state name="urn:com:bea:aler:events:type:AssetTabApproved">
  <action>MultiTier_NextTier_Assign</action>
</state>
<!--Asset Registration Status Flows-->
<state name="urn:com:bea:aler:events:type:AssetAllTabApproved">
  <action>AllTabApproved_Register</action>
</state>
```

This example configuration wires the following events to various flows. The <action> element contains the name of the flow that are executed.

1. When an asset "submitted" event is triggered, execute the Community Accept flow.
2. When an asset "accepted" event is triggered, execute the MultiTier1 flow.
3. When a tab "approved" event is triggered, execute the Multi-Tier Next Tier flow.
4. When "all the tabs approved" event is triggered, execute the Automatic Registration flow.

Some of the flows take parameters that are needed as input. Different parameters are passed to different flows. For example, the ChangeCAS (Change Custom Access Settings) flow takes <customAccessSettings> as a parameter. Here is a sample wiring when an asset is registered, where the flow automatically assigns MyCAS and MyCAS2 custom access settings.

```
<state name="urn:com:bea:aler:events:type:AssetRegister">
  <action>ChangeCAS</action>
<customAccessSettings>
  <customAccessSetting>MyCAS</customAccessSetting>
  <customAccessSetting>MyCAS2</customAccessSetting>
</customAccessSettings>
</state>
```

A.6.4 Automatic Asset Registration Flows

This section describes how the Automated Workflows can automate the manual asset acceptance and registration process done using the Oracle Enterprise Repository Asset Editor. For information on using the Oracle Enterprise Repository Asset Editor and the asset registration process, see *Oracle Fusion Middleware User's Guide for Oracle Enterprise Repository*.

> **Note:** Do not enable the "Community Acceptance" or the "Automated Acceptance" flows if repository users submit assets via the "Submit an Asset" link. This configuration is not currently supported in Oracle Enterprise Repository.

A.6.4.1 Configuring Community Flows

The Community flow provides a way to automate the asset acceptance, assignment, and registration process by allowing the configuration of automated assignment rules and also introduces the notion of federated registrars among different authorities. Rather than spamming many registrars across all communities (through the system registrar notification), you could limit the system registrar to one or a few individuals, and let the Automatic Acceptance flow accept assets on behalf of a registrar-of-record for the community. The Community flow feature can distribute asset submissions to those with the authority to approve them for the community.

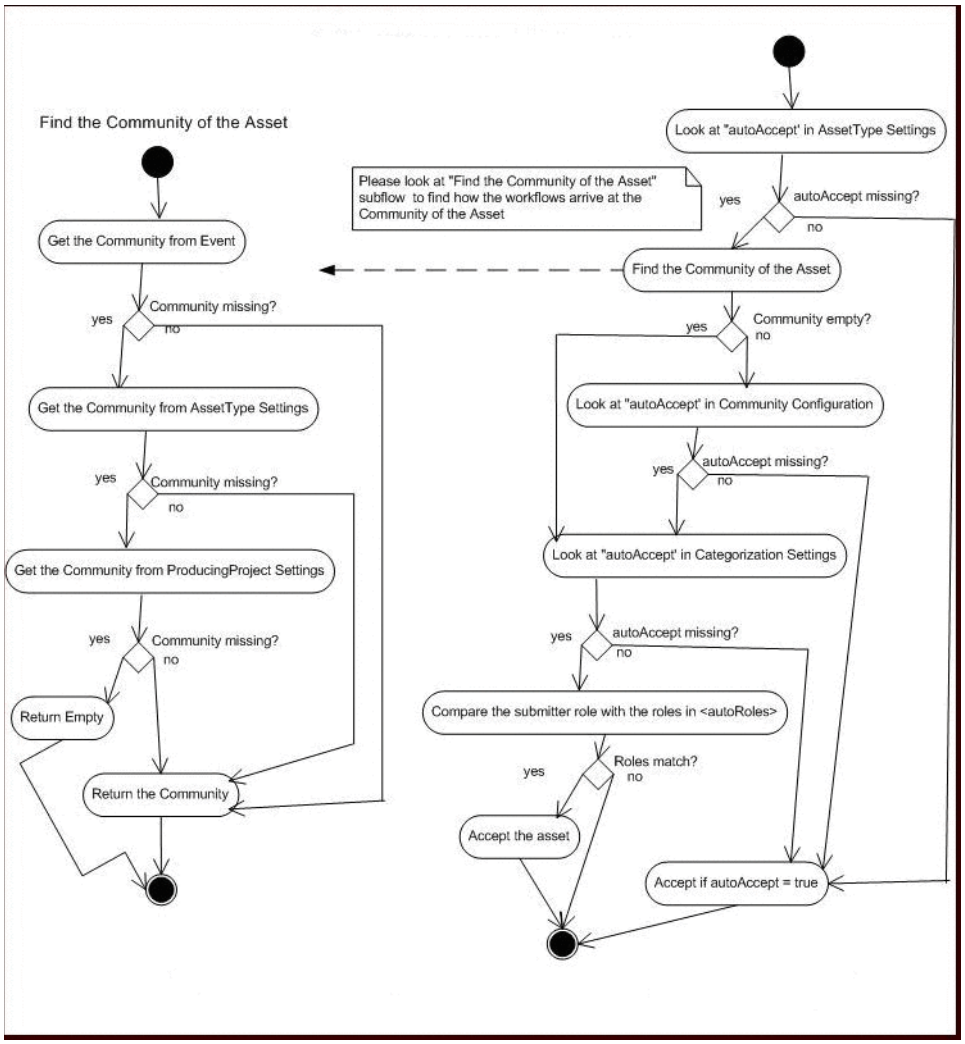
The Community flow can be used to address the following scenarios:

- Automatic federated registrars support for acceptance as opposed to a single registrar getting many notifications about newly submitted assets.
- Even if asset acceptance is manual, the Community flow can be used to automate the assignment of the asset approvals to pre-defined approvers. Creating pre-defined approvers can be achieved in two ways:
  - Creating a list of pre-defined approvers for all the tabs in that asset.
  - Using multi-tier assignment (this is the same as the Multi-Tier flow but it operates within the Community).
- Automation of the registration process. The flows will automatically register the assets if the following conditions happen:
  a. When all the tabs approved
  b. When the last tier in a Multi-tier process is completed
  c. Or whichever happens first.

  The Communities are configured within the flow configuration and Asset Types, Producing Projects, etc., can point to a Community.

  Figure A–14 demonstrates how a Community for an asset is located by the flow, as well as how the rules for automatic acceptance are located by the flow.

Figure A–15 Automatic Asset Acceptance Flowchart



Note: The same flowchart applies for automatic Registration. Simply substitute autoRegister for autoAccept.

A.6.4.1.1 Setting the Community for an Oracle Enterprise Repository Project Define the community for a project using the <producingProjectSettings> element. The following example demonstrates creating a project named "Registry" for the "SOA Center of Excellence" community, and with an ID of "40000".

```
<producingProjectSettings>
   <producingProject name="Registry" community="SOA Center of Excellence
   id="40000"/>
</producingProjectSettings>
```

A.6.4.1.2 Setting the Community for an Asset Type Define the community for an Asset Type using the <assetType> element. The following example demonstrates creating an asset type named "Application" for the "SOA Center of Excellence" community, and with an ID of "158".

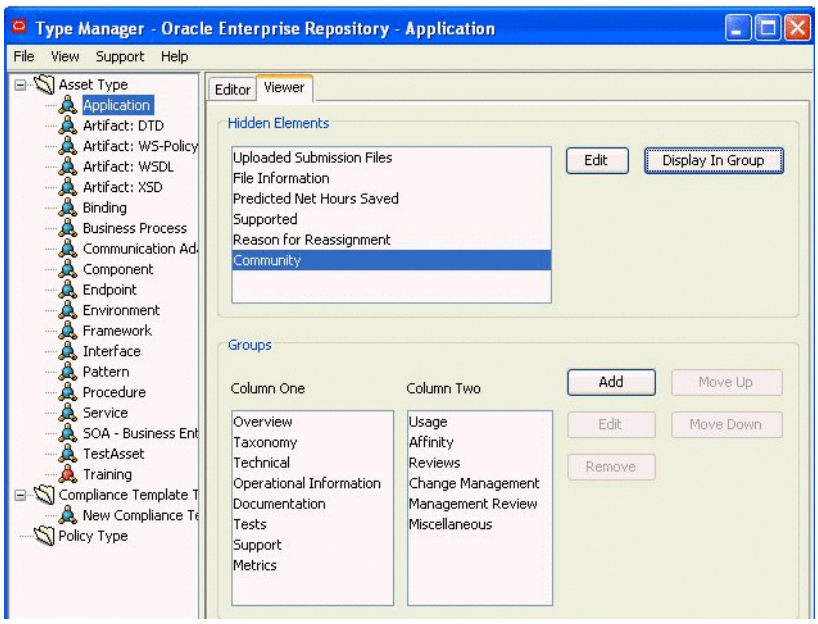
```
<assetType name="Application" community="SOA Center of Excellence
id="158">
  <allTabs>
```

A.6.4.1.3 Setting the Community for an Asset using the Type Manager and Asset Editor Instead of setting the community for an Asset in workflow.xml, you can set the community for the Asset Type using the Type Manager and Asset Editor.

In the Type Manager, follow these steps:

1. Select the Asset Type for which you want to enable the Community field, and click the **Viewer** tab.
2. Click the **Display in Group** button, as shown in Figure A–16.

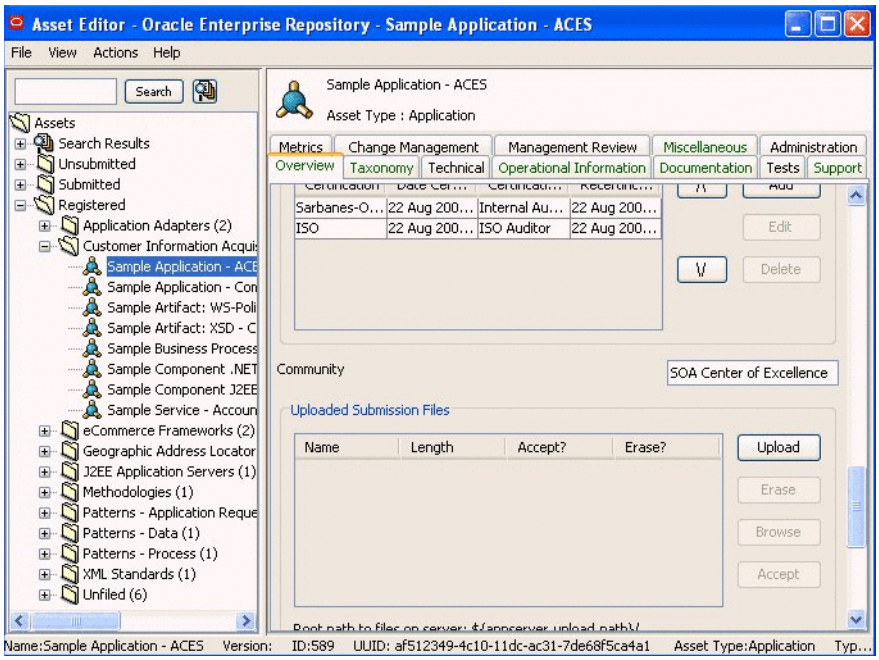
Figure A–16 Setting the Community for an Asset Type in Type Manager



Then, in Asset Editor, follow these steps:

1. Select an asset of the Asset Type you selected in the Type Manager.
2. Set the community name to use for that asset in the Community field of the Overview tab, as shown in Figure A–17.

Figure A–17 Setting the Community Name in the Community Field in Asset Editor



If you followed the instructions for setting a community in Section A.6.4.1.2, "Setting the Community for an Asset Type"and you then set a Community name for an asset in Asset Editor, the Community name you set for the asset in Asset Editor overrides the Community name set in the **workflows.xml** file.

A.6.4.1.4 Configuring a Community to Automatically Accept an Asset The following example demonstrates how to set the "SOA Center of Excellence" community to automatically accept assets.

```
<communities name="SOA Center of Excellence autoAccept="true">
```

Note: Do not enable the "Community Acceptance" or the "Automated Acceptance" flows if repository users submit assets via the "Submit an Asset" link. This configuration is not currently supported in Oracle Enterprise Repository.

A.6.4.1.5 Configuring a Community to Assign Assets for Tab Approval If the AssetSubmitted event is wired to the Community flow, then the <approvers> element lists the approvers that are assigned by the Community flow automatically.

```
<communities name="Java" autoAccept="true">
  <approvers>
    <alerid>5003</alerid>
    <alerid>5004</alerid>
  </approvers>
```

For instructions on using the <alerid> in Tab Approval flows, see Section A.6.5.2, "Using an <alerid> for Tab Approvals".

A.6.4.1.6 Configuring a Community to Assign Assets for Tab Approval Using Multi-tier

Multi-tier assignment is the same as the Multi-Tier flow but it operates within the Community. For more information about the Multi-tier flow, see Section A.6.5, "Multi-tier Automatic Assignment Flows".

> **Note:** The tabs that are provided within the Multi-tier configuration of a community should be the common tabs that exist in all the asset types.

A.6.4.1.7 Configuring a Community to Automatically Register an Asset

The following example demonstrates how to set the "SOA Center of Excellence" community to automatically accept and register assets.

```
<communities name="SOA Center of Excellence autoAccept="true"
 autoRegister="true">
```

A.6.4.1.8 Configuring a Community to Have a Dedicated Registrar

The Registrar user name and password is required to accept, assign, and register assets. The Community flow will load the registrar information from the Community that the asset belongs to. If an asset does not belong to a community or if the registrar information is not found in the community, then the global registrar is used by the Community flow.

The following is the order of precedence in getting the Community tag by the Community flows, as illustrated in Figure A–13:

- Community Tag in the incoming event
- Community Tag in the Asset Type that the incoming asset belongs to
- Community Tag in the Producing Project that the incoming asset belongs to

A.6.4.2 Configuring Automated Acceptance and Automated Registration Flows

Besides using the Community flows to automatically accept and register assets, the following rules can be used to accept and register assets, as illustrated in Figure A–13.

> **Note:** Do not enable the "Community Acceptance" or the "Automated Acceptance" flows if repository users submit assets via the "Submit an Asset" link. This configuration is not currently supported in Oracle Enterprise Repository.

A.6.4.2.1 Asset Type

The autoAccept and autoRegister flag within the AssetType element can be used to automatically accept or register assets.

```
<assetType name="Application" autoAccept="true" autoRegister="true"
id="158">
  <allTabs>
    <tab name="Overview"/>
    <tab name="Application Lifecycle"/>
  </allTabs>
```

A.6.4.2.2 Categorization Settings

By default the flows do not look for the autoAccept and autoRegister flags, since the look-up may affect performance. However, this can be enabled by using the <action> flag.

As shown in this example, the <action> flag must be set to true if the flows should use the Categorization settings. If not, the Categorization settings are ignored.

```
<catgorizationTypeSettings action="true">
  <catgorizationType name="AssetFunction" type "100">
    <catgorizations name="Application Adapters" autoAccept="false"/>
    <catgorizations name="Customer Information Acquisition" autoAccept="false"/>
    <catgorizations name="eCommerce Frameworks" autoAccept="false"/>
  </catgorizationType>
```

A.6.4.2.3 Submitter Role The submitter role can be used to automatically accept or register the asset. If the role specified in the following configuration matches the submitter role, then the asset is automatically accepted.

```
<automation>
  <autoRoles>
    <role>admin</role>
    <role>accesAdminstrator</role>
  </autoRoles>
  <autoApprovalTabs>
    <tab name="Documentation"/>
  </autoApprovalTabs>
</automation>
```

A.6.4.2.4 Conflict Resolution and Precedence In some cases, there are more than one rule that matches for a given event trigger, so there is a hierarchy for how each rule is evaluated by the Automated Acceptance and Automated Registration flows for acceptance, registration, etc., as illustrated in Figure A–13. The flow will scan for the following piece of metadata and as soon as it encounters the one in the following precedence, it will break and use the settings in that metadata.

- AssetType settings in the Flow configuration file
- Community Tag found in the incoming asset
- Community Tag found in the AssetType settings in the Flow configuration file
- Community Tag found in the ProducingProject settings in the Flow configuration file
- Categorization settings in the Flow configuration file
- Categorization settings in the Flow configuration file
- SubmitterRole settings in the Flow configuration file

A.6.5 Multi-tier Automatic Assignment Flows

Multi-tier flows structure the asset tab approval process in multiple steps called tiers. Asset approval tabs can be grouped in tiers, and the Mult-tier flow tracks each tier to verify whether all the tabs are approved by the designated approvers. As soon as the last tab in a tier is approved, the Mult-tier flow starts the next tier by assigning the asset to the next level of designated approvers.

A.6.5.1 Use Cases

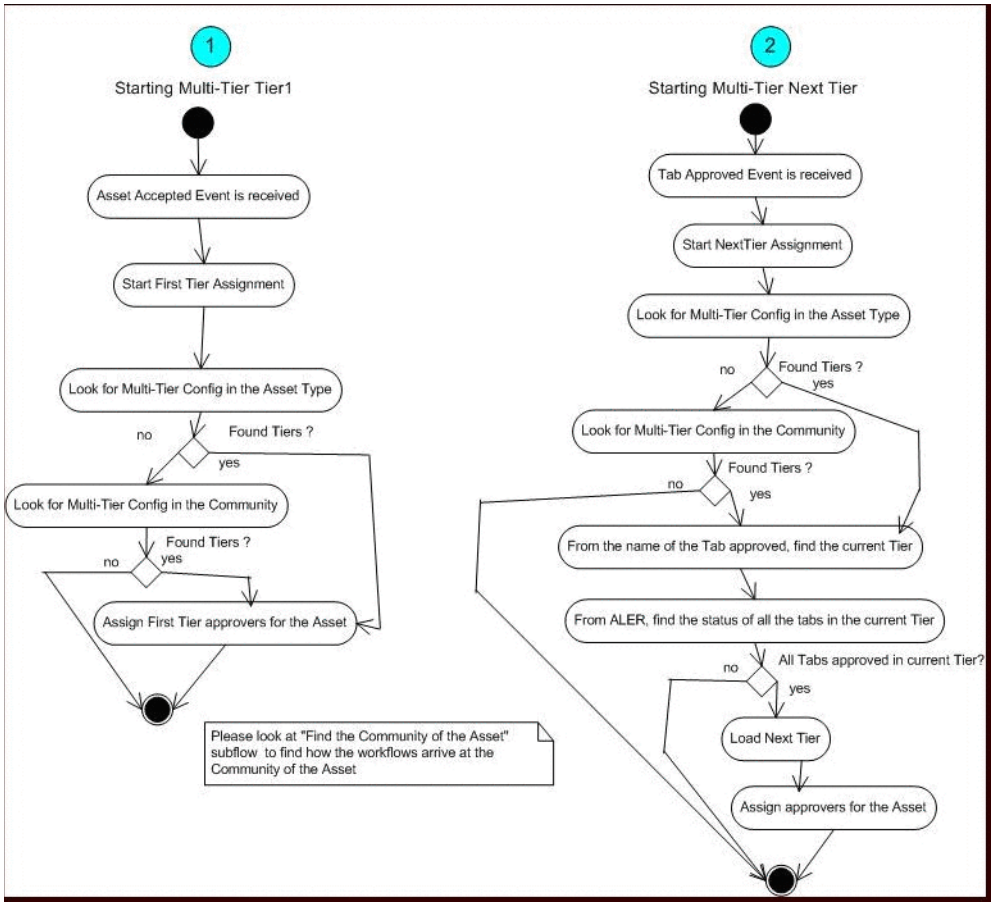
- In some cases, it may be desired to assign tabs for Tab Approval in multiple steps called Tiers. For example, it may be desirable to approve the Architecture tab first before approving the Documentation tab. This is because any architectural issue

needs to be corrected first before it comes to the attention of the Documentation expert.

- In previous releases, Tab Approval was done manually by the registrar by manually tracking the status of each tab approval and then assigning the tabs for the next tier level approvals. With the Multi-tier flows, this process is automated by the flows.

Figure A–18 demonstrates the flow of the Mult-tier process.

Figure A–18 Multi-tier Automatic Assignment Flowchart



A.6.5.2 Using an <alerid> for Tab Approvals

When the workflow.xml file is generated, the following XML section is created under the <allAssetSettings> section. These are all the users that are created in Oracle Enterprise Repository.

```
<alerUsers>
  <user name="admin" alerid="99"/>
  <user name="allpriv" alerid="50000"/>
  <user name="nopriv" alerid="50001"/>
  <user name="tier1" alerid="50002"/>
  <user name="tier2" alerid="50003"/>
  <user name="mrsmith" alerid="50004"/>
</alerUsers>
```

As the Workflow Administrator, you need to identify the user(s) by name that you want to use for approving the asset tabs and use the corresponding <alerid>. Then you can use that <alerid> in the Workflow XML, such as in the following Multi-tier approval flow:

```
<tiers>
  <tier name="Tier1">
    <approvers>
      <alerid>50001</alerid>
    </approvers>
    <tabs>
      <tab name="Overview"/>
      <tab name="Technical"/>
      <tab name="Documentation"/>
    </tabs>
</tier>
```

A.6.5.3 Setting Up a Community for Multi-tier Tab Approval

The following example demonstrates how the Multi-tier flow is configured for tab approvers in the "SOA Center of Excellence" community to automatically accept tabs.

```
<communities name="SOA Center of Excellence autoAccept="true">
    <tiers>
      <tier name="Tier1">
        <approvers>
          <alerid>5002</alerid>
        </approvers>
        <tabs>
          <tab name="Overview"/>
          <tab name="Taxonomy"/>
        </tabs>
      </tier>
      <tier name="Tier2">
        <approvers>
          <alerid>5003</alerid>
        </approvers>
        <tabs>
          <tab name="Architecture/">
        </tabs>
      </tier>
    </tiers>
  </communities>
```

> **Note:** Tabs that are provided within the Multi-tier configuration of a Community should be the common tabs that exist in all the Asset Types.

A.6.5.4 Setting Up an Asset Type for Multi-tier Tab Approval

The following example demonstrates how the tabs of an asset type of "Application" are configured for multi-tier approval.

```
<assetType name="Application" id="158">
    <allTabs>
      <tab name="Oveview"/>
      <tab name="Application Lifecycle"/>
      <tab name="License Information"/>
```

```
  <tab name="Certification Tracking"/>
  <tab name="Taxonomy"/>
  <tab name="Documentation"/>
  <tab name="Relationships"/>
  <tab name="Support"/>
  <tab name="Cost Categories"/>
  <tab name="Ownership"/>
  <tab name="Technology Stack"/>
  <tab name="Operational Information"/>
  <tab name="Miscellaneous"/>
</allTabs>
<tiers>
  <!--Please change "_CHANGE_TIER1_NAME_" to the name of the Tier-->
  <!--Example:- "Tier1"-->
  <tier name="Tier1">
    <approvers>
      <alerid>99</alerid>
    </approvers>
    <tabs>
    <!--Please change "_CHANGE_TABNAME_" to the name of the Tab-->
    <!--Example:- "Documentation"-->
      <tab name="Overview"/>
      <tab name="Taxonomy"/>
    </tabs>
  </tier>
</tiers>
```

A.6.6 Metadata Change Flows

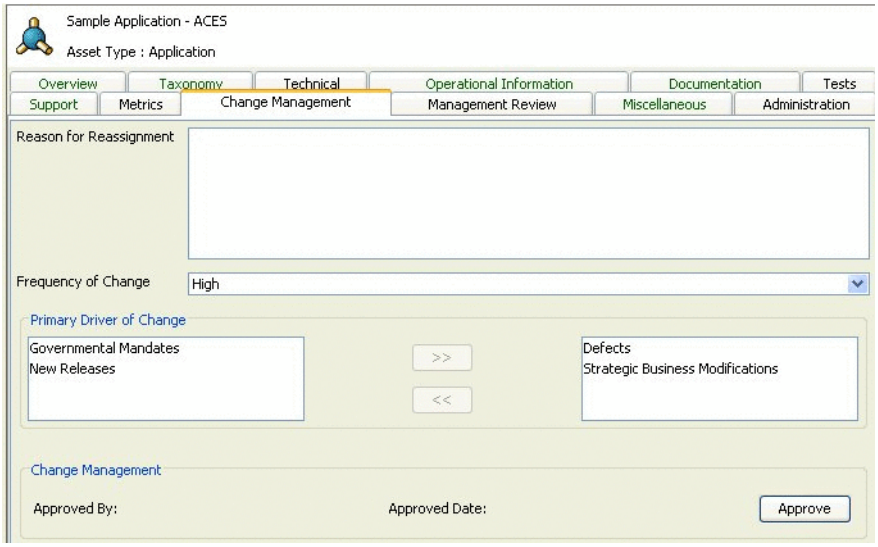
Metadata flows are a group of flows that take one or more actions when a metadata element of an asset changes. The Metadata element that changes will trigger an event that is wired to one or more flows. For instructions on how to wire an event to a flow, see Section A.6.3, "Wiring Asset Events to Flows".

A.6.6.1 Use Cases

These are some of the use cases where Metadata Change Flows may apply:

- When the "Asset Lifecycle Stage" metadata element of an asset changes from "Build" to "Release,", you may want to change Custom Access Settings to have more restricted access control to the asset.
- When the "Name" of an asset changes, you may want to notify the subscribers.
- When any metadata element of an element changes, you may want the asset to go through a "Change Management" approval process. The "Change Management" will involve the following:
  - Unapprove a tab named "Change Management." The Change Management tab in Asset Editor is shown in Figure A–19.

Figure A–19 The Asset Editor Change Management Tab



- Assign the asset to the registrar
- Append the kind of change to a field called "Reason for reassignment" to assist the registrar

A.6.6.2 Configuring Metadata Change Flows

This section describes the configuring metadata change flows. This section contains the following topics:

- Section A.6.6.2.1, "Available Metadata Change Events/States"
- Section A.6.6.2.2, "Available Flows That Can Be Wired to Actions"
- Section A.6.6.2.3, "Example Metadata Change Configuration"
- Section A.6.6.2.4, "Example Metadata Change Configuration That Checks for Metadata Value"
- Section A.6.6.2.5, "ChangeClassification"
- Section A.6.6.2.6, "ChangeCAS"
- Section A.6.6.2.7, "ChangeAssetLifecycle"
- Section A.6.6.2.8, "ApproveTabAction"
- Section A.6.6.2.9, "UnapproveTabAction"
- Section A.6.6.2.10, "AutoApproveTabAction"
- Section A.6.6.2.11, "UnapproveChangeManagementTab"
- Section A.6.6.2.12, "ResetChangeManagementTab"
- Section A.6.6.2.13, "NotifyCustomUser"
- Section A.6.6.2.14, "Invoking Flows Based on Approval of Named Tabs"

A.6.6.2.1 Available Metadata Change Events/States Following are the states that are available that can be wired to Metadata Change flows.

Note: Besides these events, any categorization changes can be wired, including the custom categorization.

```
<state name="urn:com:bea:aler:events:type:MetaDataChange:name">
 <state name="urn:com:bea:aler:events:type:MetaDataChange:version">
 <state name="urn:com:bea:aler:events:type:MetaDataChange:description">
 <state name="urn:com:bea:aler:events:type:CategorizationChanged:
 assetLifecycleStage"/>
 <state name="urn:com:bea:aler:events:type:CategorizationChanged:classification">
 <state name="urn:com:bea:aler:events:type:MetaDataChange:supported">
 <state
name="urn:com:bea:aler:events:type:MetaDataChange:organizationalOwnership">
 <state name="urn:com:bea:aler:events:type:MetaDataChange:usageFee">
```

For most asset types, the usageFee field is found on the Miscellaneous tab of the Asset Editor.

A.6.6.2.2 Available Flows That Can Be Wired to Actions These are the pre-defined flows that can be wired to actions. These flow names should appear as content inside the <action> element to indicate that this is the action that should take place when the event occurs. Note that any element other than <action> are parameters used by specific flows.

- 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
- AddCommunityTag: saves the "Community" of an asset to Oracle Enterprise Repository
- NotifySubscriber: notifies the Subscribers about the Metadata Change
- NotifyRegistrationActors: notifies the Registrar, Subscribers, Owners, etc., about the Metadata Change
- NotifyCustomUser: notifies configured custom users about the Metadata Change
- UnapproveChangeManagementTab: triggers the Change Management process
- ResetChangeManagementTab: resets the "Reason for reassignment" field in the Change Management tab as soon as the Change Management tab is approved
- CommunityAccept: invokes the Community Accept Flow used when an asset is submitted
- CommunityAssign: invokes the Community Assign Flow used when an asset is accepted
- MultiTier_Tier1_Assign: invokes the Multi-Tier Flow used when an asset is accepted
- MultiTier_NextTier_Assign: invokes the Multi-Tier Flow used when a tab is approved
- ApproveTabAction: approves one or more tab
- UnapproveTabAction: unapproves one or more tab

- AutoApproveTabAction: approves one or more configured tab based on the role of the submitter
- AllTabsApproval_Register: invokes the flow to register the asset when all the tabs are approved
- ReAssignAssetToRegistrar: Assigns the asset to the Registrar for approval. The flow uses the Community Registrar if one is configured. If not, it uses the Global Registrar.
- ResetFlowState: Resets the State information used by the Timer based flows. This is useful in cases where a Timer flow is tracking the Unsubmitted assets and when the state changes from Unsubmitted to submitted, so the State information can be reset. If not reset, then if the asset goes back to Unsubmitted, the workflows use the same state that was previously set. This is not always desirable and the ResetFlowState action can be used in appropriate events or states to reset the state information.
- UnRegisterAssetAction: Unregisters the Asset if the asset is in registered state.
- autoSyncAlerToUddi: Timer-based workflow that moves services from Oracle Enterprise Repository to Oracle Service Registry.
- autoSyncUddiToAler: Timer-based workflow that moves services from Oracle Service Registry to Oracle Enterprise Repository.
- 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.

For more information about this workflow, see Section 10.3.1.1, "Invoking the Oracle Registry Repository Exchange Utility Using Workflows".

A.6.6.2.3 Example Metadata Change Configuration This sample configuration specifies that when an asset is registered, it invokes two flows by the names of "NotifySubscriber" and "ChangeCAS." Note that the element `<customAccessSettings>` is a parameter to the flow ChangeCAS, which tells the flows the names of the CAS that should be applied.

```
<state name="urn:com:bea:aler:events:type:AssetRegister">
    <action>NotifySubscriber</action>
    <action>ChangeCAS</action>
    <customAccessSettings>
      <customAccessSetting>MyCAS</customAccessSetting>
      <customAccessSetting>MyCAS2</customAccessSetting>
    </customAccessSettings>
  </state>
  <state name="urn:com:bea:aler:events:type:AssetUnAccept">
    <action>NotifySubscriber</action>
    <action>ChangeClassification</action>
    <classification>Approved</classification>
  </state>
```

A.6.6.2.4 Example Metadata Change Configuration That Checks for Metadata Value It is also possible to invoke a flow not only when a metadata element changes, but also when it takes a specific value. For example, when the "Asset Lifecycle Stage" metadata element of an asset changes from "Build" to "Release," you may want to apply one set of Custom Access Settings, where as when the value changes from "Plan" to "Build," you may want to apply a different set. Here is an example:

```
<state
name="urn:com:bea:aler:events:type:CategorizationChanged:AssetLifecycleStage"
 value="Stage 4 - Release">
    <action>ChangeCAS</action>
    <customAccessSettings>
      <customAccessSetting>MyCAS</customAccessSetting>
    </customAccessSettings>
  </state>
  <state
 name="urn:com:bea:aler:events:type:CategorizationChanged:AssetLifecycleStage"
 value="Stage 3 - Build">
    <action>ChangeCAS</action>
    <customAccessSettings>
      <customAccessSetting>MyCAS2</customAccessSetting>
    </customAccessSettings>
  </state>
```

A.6.6.2.5 ChangeClassification Sets the classification of an asset. ChangeClassification uses the following element to set the classification.

```
<state name="urn:com:bea:aler:events:type:AssetRegister">
    <action>ChangeClassification</action>
    <classification>Approved</classification>
  </state>
```

A.6.6.2.6 ChangeCAS Applies one or more Custom Access Settings to an asset. ChangeCAS uses the following element to set the custom access settings.

```
<state name="urn:com:bea:aler:events:type:AssetRegister">
    <action>ChangeCAS</action>
    <customAccessSettings>
      <customAccessSetting>MyCAS</customAccessSetting>
      <customAccessSetting>MyCAS2</customAccessSetting>
    </customAccessSettings>
  </state>
```

A.6.6.2.7 ChangeAssetLifecycle Sets the Asset Lifecycle stage of an asset. ChangeAssetLifeCycle uses the following element to set the asset life cycle.

```
<state name="urn:com:bea:aler:events:type:AssetRegister">
    <action>ChangeAssetLifeCycle</action>
    <assetLifeCycle>Stage 3 - Build</assetLifeCycle>
  </state>
```

A.6.6.2.8 ApproveTabAction The ApproveTabAction flow approves one or more tabs of an asset. The following configuration approves the "Overview" and "Taxonomy" tabs.

```
<state name=?urn:com:bea:aler:events:type:MetaDataChange:name?>
    <action>ApproveTabAction</action>
    <approveTabs>
      <tab name="Overview"/>
      <tab name="Taxonomy"/>
   </approveTabs>
  </state>
```

A.6.6.2.9 UnapproveTabAction The following element configures the list of tabs to be unapproved by the UnapproveTabAction flow.

```
<state name="urn:com:bea:aler:events:type:MetaDataChange:name">
    <action>UnApproveTabAction</action>
    <unapproveTabs>
      <tab name="Overview"/>
      <tab name="Taxonomy"/>
    </unapproveTabs>
  </state>
```

A.6.6.2.10 AutoApproveTabAction The AutoApproveTabAction flow approves tabs based on the role of the submitter. For example, the following element under <allAssetSettings> configures the list of tabs that need to be automatically approved based on the role of the submitter. The roles that are acceptable are also configured.

```
<automation>
    <autoRoles>
      <role>admin</role>
      <role>accesAdminstrator</role>
    </autoRoles>
    <autoApprovalTabs>
      <tab name="Documentation"/>
    </autoApprovalTabs>
  </automation>
```

Here is the configuration for invoking the flow:

```
<state name="urn:com:bea:aler:events:type:AssetRegister">
    <action>AutoApproveTabAction</action>
  </state>
```

A.6.6.2.11 UnapproveChangeManagementTab When any metadata element of an element changes, you may want the asset to go through a "Change Management" approval process, which involves following.

- Unapprove a tab by name "Change Management." The Change Management tab in Asset Editor is shown in Figure A–19.
- Assign the asset to the registrar.
- Append the kind of change to a field called "Reason for reassignment" to assist the registrar

```
<state name="urn:com:bea:aler:events:type:MetaDataChange:name">
   <action>UnApproveChangeManagementTab</action>
</state>
```

A.6.6.2.12 ResetChangeManagementTab This flow resets the "Reason for reassignment" field in the Change Management tab as soon as the Change Management tab is approved.

```
<state name="urn:com:bea:aler:events:type:AssetTabApproved">
    <action>MultiTier_NextTier_Assign</action>
    <action>ResetChangeManagementTab</action>
</state>
```

A.6.6.2.13 NotifyCustomUser Notifies configured custom users about the metadata change. The E-mail addresses of the users are configured inside the <customNotification> element under <allAssetSettings>, as shown below:

```
<allAssetSettings>
    <notification timerInterval="id">
      <customNotification>
        <emailAddress>smith@bea.com</emailAddress>
      </customNotification>
    </notification>
```

A.6.6.2.14 Invoking Flows Based on Approval of Named Tabs A metadata change flow can be executed based on the approval of a specific tabs, as follows:

```
<state name="urn:com:bea:aler:events:type:AssetTabApproved" value ="Overview">
    <action>MultiTier_NextTier_Assign</action>
    <action>ChangeAssetLifecycle</action>
    <assetLifecycle>Stage 3 - Build</assetLifecycle>
  </state>
```

A.6.7 Time-based Escalation Flows

The Time-based Escalation flows track assets in various states and notifies all interested parties. The following section explains how to configure the Time-based Escalation flows. There are four different kinds of Time-based Escalation flows and each one can be configured individually, as described in the following sections.

Open the workflow.xml configuration file and locate the <notification> element.

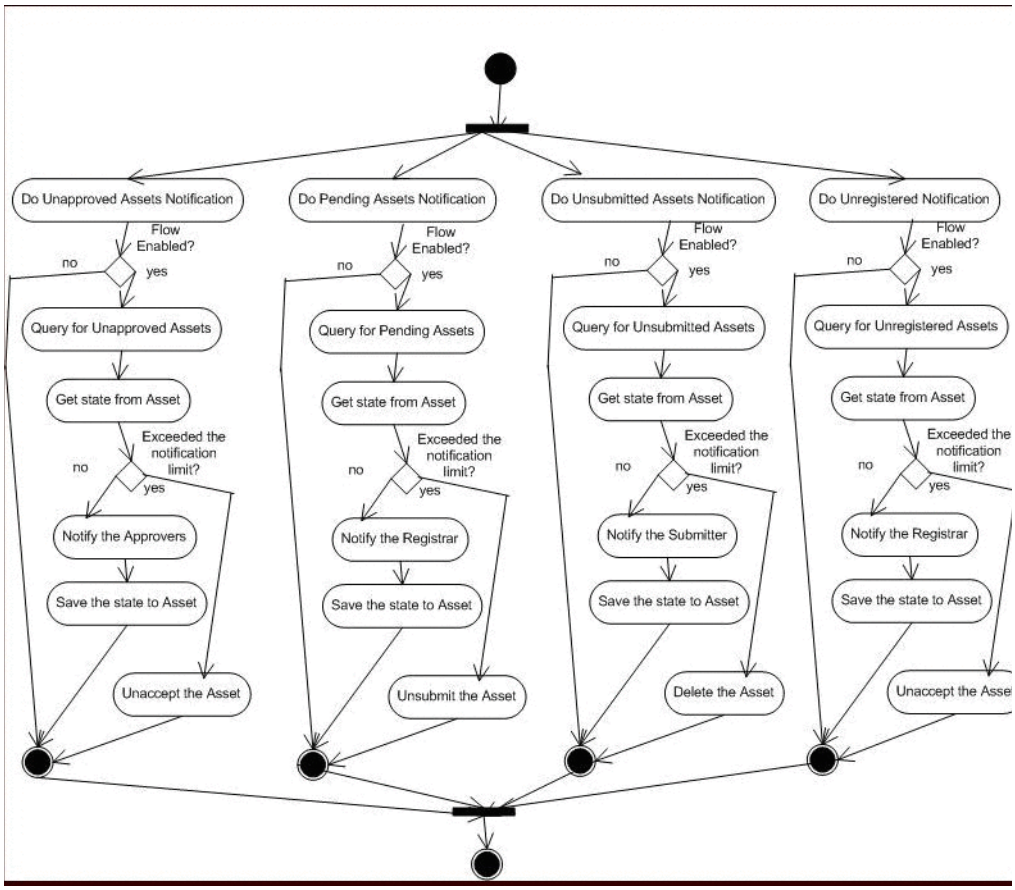
```
<notification timerInterval="1d">
    <numTimesNotify>10</numTimesNotify>
    <daysBeforeNextNotification>2</daysBeforeNextNotification>
```

- The timerInterval element specifies the time interval after which the flows are triggered. In a production environment, this should be set to "1d", which means the flows are triggered once a day. However for testing purposes, you can set it to "1m" or "5m" to trigger the flows every minute or every five minutes. Also, each time this field is changed, the Event Engine needs to be restarted, unlike the other field changes that can be refreshed using the refresh tool.
- The numTimesNotify element specifies how many times the notifications should be sent by the Time-based Escalation flows.
- The daysBeforeNextNotification element specifies how many days need to elapse in between the notifications.

> **Note:** If the timerInterval element is configured in minutes to trigger flows in minute intervals for testing purposes, then the specified interval for daysBeforeNextNotification will also be interpreted in minutes.

Figure A–20 demonstrates the flow of the Time-based Escalation flows.

Figure A–20 Time-based Escalation Flowchart



A.6.7.1 Tracking Unsubmitted Assets

This flow tracks assets that are in an "unsubmitted" status and sends notification to the owners to take action.

```
<owner_resubmit action="false" days="0" regressOnInaction="true"
queryOperator="eq"/>
```

- action="true" enables the flow and action="false" disables the flow.
- days="10" tracks the assets that reached unsubmitted status 10 days ago. The Time-based Escalation flows use the current date and subtracts the value from this attribute.
- regressOnInaction="true" regresses the asset on inaction. For example, unsubmitted assets may be deleted.
- queryOperator="eq" uses the equals operator when the date is used for querying. Other possible values are "lte", "gte", etc.

A.6.7.2 Tracking Unaccepted Assets

This flow tracks assets that are in an "unaccepted" status and sends notification to the registrar to take action.

```
<registrar_accept action="false" days="0" regressOnInaction="true"
queryOperator="eq"/>
```

- action="true" enables the flow and action="false" disables the flow.
- days="10" tracks the assets that reached unsubmitted status 10 days ago. The Time-based Escalation flows use the current date and subtracts the value from this attribute.
- regressOnInaction="true" regresses the asset on inaction. For example, submitted assets may be unsubmitted.
- queryOperator="eq" uses the equals operator when the date is used for querying. Other possible values are "lte", "gte", etc.

A.6.7.3 Tracking Unapproved Assets

This flow tracks assets that are in an "unapproved" status and sends notification to the approvers to take action.

```
<assignees_approve action="false" days="0" regressOnInaction="true"
queryOperator="eq"/>
```

- action="true" enables the flow and action="false" disables the flow.
- days="10" tracks the assets that reached unsubmitted status 10 days ago. The Time-based Escalation flows use the current date and subtracts the value from this attribute.
- regressOnInaction="true" regresses the asset on inaction. For example, accepted assets may be unaccepted.
- queryOperator="eq" uses the equals operator when the date is used for querying. Other possible values are "lte", "gte", etc.

A.6.7.4 Tracking Unregistered Assets

This flow tracks the assets that are in an "unregistered" status and sends notification to the approvers to take action.

```
registrar_register action="false" days="0" regressOnInaction="true"
queryOperator="eq"/>
```

- action="true" enables the flow and action="false" disables the flow.
- days="10" tracks the assets that reached unsubmitted status 10 days ago. The Time-based Escalation flows use the current date and subtracts the value from this attribute.
- regressOnInaction="true" regresses the asset on inaction. For example, registered assets may be unregistered.
- queryOperator="eq" uses the equals operator when the date is used for querying. Other possible values are "lte", "gte", etc.

A.6.8 Validation Expiration Flows

The Validation Expiration flows track the expired assets prior to the expiration date, as well as on the date of expiration, and sends warning notifications to all interested parties. After X number of days of expiration, the flows unregister the assets. After Y number of days of expiration, the flows deactivate the assets. After Z number of days of expiration, the flows delete the assets.

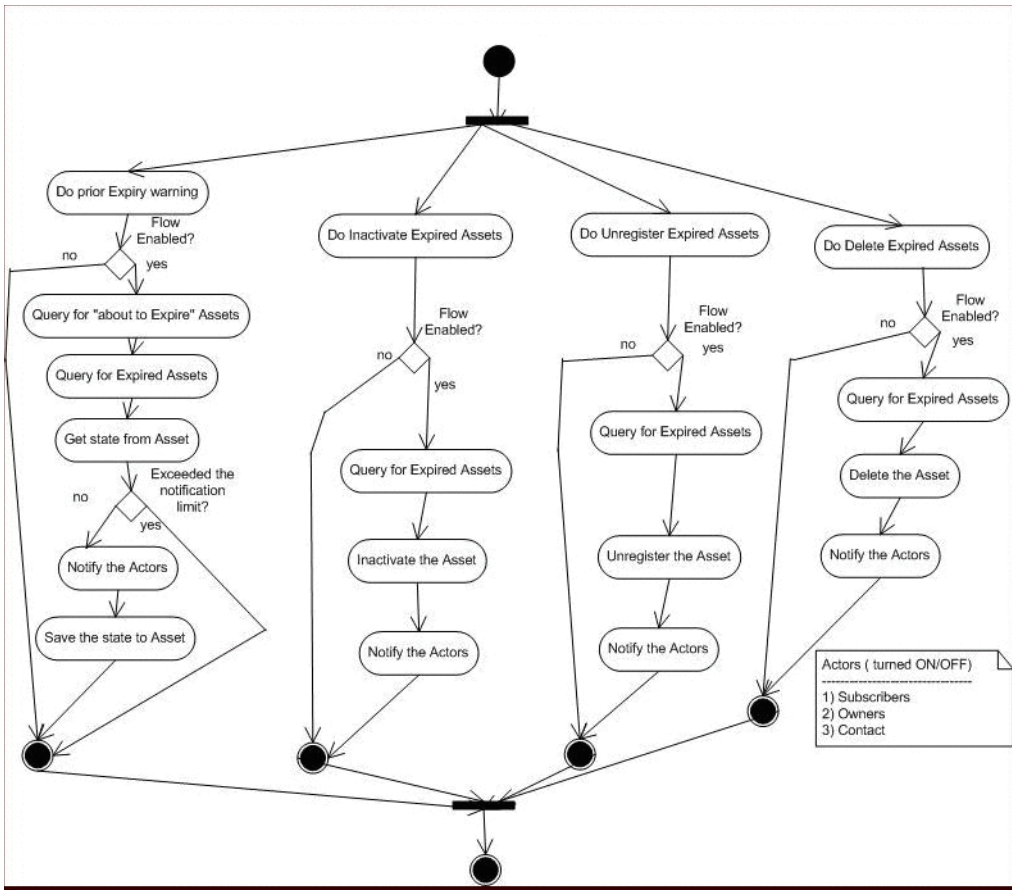
```
<notification timerInterval="1d">
    <numTimesNotify>10</numTimesNotify>
    <daysBeforeNextNotification>2</daysBeforeNextNotification>
```

- The timerInterval attribute configures the time interval that the flows are triggered. This should be set to "1d", which means the interval is one day. However for testing, this can be set to "1m" or "5m" to trigger every minute or every 5 minutes. Also, every time this field is changed, the Event Engine needs to be restarted, unlike the other field changes that can be refreshed using the refresh tool.
- The numTimesNotify element specifies how many times the notifications should be sent by the Validation Expiration flow.
- The daysBeforeNextNotification element specifies how many days need to elapse in between the notifications.

```
<expiration>
    <expiration_warning action="false" days="10" owner="false"
subscriber="false" contact="99"/>
    <unregister_after_expire action="true" days="10" queryOperator="eq"/>
    <inactive_after_expire action="true" days="10" queryOperator="eq"/>
    <delete_after_expire action="true" days="10" queryOperator="eq"/>
  </expiration>
```

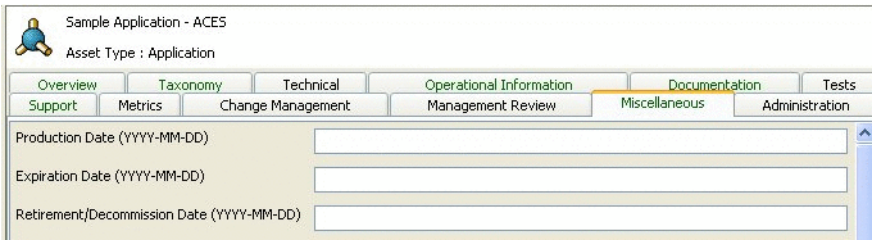
Figure A–21 demonstrates the flow of the Validation Expiration flows.

Figure A–21 Validation Expiration Flowchart



To set the expiration date for an asset, specify the date in the Expiration Date (YYYY-MM-DD) field on the Miscellaneous tab of the Asset Editor, as shown in Figure A–22.

Figure A–22 The Expiration Date (YYYY-MM-DD) Field on the Asset Editor Miscellaneous Tab



A.6.8.1 Asset Expiration Warning Notification

The following line enables the warning notification and determines who should receive the notifications.

```
<expiration_warning action="false" days="10" owner="false" subscriber="false"
contact="99"/
```

> **Note:** The days element configures the number of days prior to the expiration that the warning should be sent.

A.6.8.2 Unregister Assets After Expiration

The following line enables the Metadata Change flow to unregister the asset after 10 days of expiration.

```
<unregister_after_expire action="true" days="10" queryOperator="eq"/>
```

A.6.8.3 Inactivate After Expiration

The following line enables the Metadata Change flow to inactivate the asset after 10 days of expiration.

```
<inactive_after_expire action="true" days="10" queryOperator="eq"/>
```

A.6.8.4 Delete Assets After Expiration

The following line enables the Metadata Change flow to delete the asset after 10 days of expiration:

```
<delete_after_expire action="true" days="10" queryOperator="eq"/>
```

A.6.9 Customizing Email Notification Templates

The Automated Registration Flows automatically send email notifications under many circumstances. There are five new email templates for the new flows. The email templates are stored within Oracle Enterprise Repository and the flows invoke an Oracle Enterprise Repository API by passing name/value pairs that are then substituted by Oracle Enterprise Repository.

Administrators can customize the email subject, body, etc., the same way as other email templates. The following are the templates that are used by the Automated Workflows:

- Metadata of asset has changed: Notifies the registrar and the users assigned to the asset that the metadata has changed.
- Registration status unchanged: Notifies the registrar and the users assigned to the asset that the registration status <%asset.reg.status%> has remained unchanged for more than <%action.pending.days%> days.
- Status of expired asset has changed: Notifies the registrar and the users assigned to the expired asset that the status has changed.
- Prior to expiration: Notifies the registrar and the users assigned to the asset that it is due for expiration.
- Asset has been expired: Notifies the registrar and the users assigned to the asset that it has been expired.

For more information about email templates, see Chapter 8.4, "Working with E-Mail Templates".

A.6.10 Email Notification Use Cases

This section describes the Email Notification use cases and explains how the following email templates can be triggered:

- **Asset has been expired**: Notifies the registrar and the users assigned to the asset that it has been expired.

  This email template is triggered whenever the asset gets expired and this email template is created as part of the advanced workflow configuration. To set the expiration date for an asset, specify the date in the Expiration Date (YYYY-MM-DD) field in the Miscellaneous tab of the Asset Editor. There is an element in `workflow.xml`, where you can specify what needs to happen once the asset expires.

- **Asset regressed**: Notifies the registrar and the users assigned to the asset that the asset registration status has remained unchanged for more than `<%action.pending.days%>` days.

  This email template is also created as part of advanced workflow configuration. For example, under `<notification timerInterval="1d">`, there is `<registrar_accept action="false" days="5" regressOnInaction="true" queryOperator="eq"/>`. This flow tracks assets that are in an "accepted" status for the number of days specified and sends notification to the registrar to take action.

  When regressOnInaction="true", regresses the asset for inaction. For example, the accepted assets may become unaccepted.

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

  This email template is triggered whenever asset is being used in a project. This email template sends notification to the email address specified in Notification Email text field, under the Miscellaneous tab of the asset.

- **Compliance Template Applied**: Notifies project leaders when a compliance template has been applied to a project.

This email template is triggered whenever a compliance template type has been assigned to a project. Compliance Template type is part of the archetype for an asset type.

- **Metadata of asset has changed**: Notifies the registrar and the users assigned to the asset that the metadata has changed.

  This email template is also part of the advanced workflow configuration and is triggered whenever metadata has changed. There are multiple metadata change flows. For more information, see Section A.6.6, "Metadata Change Flows".

- **New Asset Version Under Development**: Notifies subscribers when a new version of an asset is being developed.

  This email template notifies subscribers when a new asset version is under development. New version is identified based on the relationship "next version" or "previous version".

- **New Version Registered**: Notifies subscribers that a new version of the asset is registered.

  This email template notifies subscribers when a new asset version is registered. New version is identified based on the relationship "next version" or "previous. version".

- **Prior to expiration**: Notifies the registrar and the users assigned to the asset that it is due for expiration.

  This email template is also used with advanced workflow configuration and is triggered when the asset is due for expiration. For example, `<expiration_warning action="false" days="5" owner="false" subscriber="false" contact="admin@example.com" queryOperator="eq"/>`.

- **Registration status unchanged**: Notifies the registrar and the users assigned to the asset that the registration status '`<%asset.reg.status%>`' has remained unchanged for more than `<%action.pending.days%>` days.

  This email template is also used with workflow configuration and is triggered when registration status is not changed for the number of days specified in workflow configuration. For example, `<registrar_register action="false" days="5" regressOnInaction="false" queryOperator="eq"/>`.

- **Status of expired asset has changed**: Notifies the registrar and the users assigned to the expired asset that the status has changed.

  This email template is also used with advanced workflow configuration and is triggered when expired asset status gets changed.

- **Usage Reassigned**: Notifies the user to whom an asset usage record has been reassigned.

  This email template is triggered from couple of places.

  From Project Detail:

  1. Go to project detail.
  2. Click **reassign users/usage**.
  3. Select reassign usage only.
  4. Select current or different project.
  5. Click **Next**. The list of users corresponding to that project is displayed.
  6. Select a user to whom you want to transfer the usage.

This email template sends notification to the user to whom usage was transferred.

From Edit Project:

1. Click **Edit Project**.
2. Click **Edit Users**.
3. Remove the user with usage record. The Assign option to whom the usage can be transferred is displayed.

   This email template triggers the notification to the user to whom usage was transferred.

A.6.11 Known Issues

This section contains the following known issues in Oracle Enterprise Repository workflow:

- The `workflow-tools/refresh_workflows.sh` file has the path to JDK hard-coded and is invalid. If you are using a different JDK, then you should manually edit the refresh_workflows.sh file to map to that JDK.
- When you modify a file on Windows and save it with an editor that does not understand how to handle CR/NL characters in Unix, then you must run the following command to remove the LF characters from the <file.sh> file:

  ```
  cat filename.sh | tr -d '\015' > filename.sh.tmp; mv filename.sh.tmp
  filename.sh
  ```

A.7 Configuring JMS Servers for Oracle Enterprise Repository

This section describes how to configure JMS servers for Oracle Enterprise Repository.

This section contains the following topics:

- Section A.7.1, "Overview of JMS for the Event Manager"
- Section A.7.2, "Configuring Connectivity Properties for External JMS Servers"
- Section A.7.3, "Configuring the Embedded ActiveMQ JMS Server to Use a Database"
- Section A.7.4, "Configuring JMS Servers in an Oracle Enterprise Repository Cluster"
- Section A.7.5, "Configuring a JMS Provider In WebSphere 6.1.0.5"

A.7.1 Overview of JMS for the Event Manager

The Event Manager uses an embedded version of Apache ActiveMQ JMS Server that is enabled by default. The embedded JMS Server is configured to run out-of-the-box without any additional configuration. However, if an external JMS server is preferred, such as Oracle's Weblogic Server JMS or IBM WebSphere Application Server, then a number of Oracle Enterprise Repository system settings must be configured.

Note: When Oracle Enterprise Repository is deployed on WebSphere 6.x, the embedded Apache ActiveMQ JMS Server cannot be used due to conflicts in the classes used by ActiveMQ and Oracle Enterprise Repository. Therefore, WebSphere 6.x customers should use the default JMS implementation that comes with WebSphere 6.x. See Section A.7.5, "Configuring a JMS Provider In WebSphere 6.1.0.5".

A.7.2 Configuring Connectivity Properties for External JMS Servers

Oracle Enterprise Repository's System Settings section allows administrators to configure the basic Oracle Enterprise Repository operation and to enable/disable specific features. The Event Manager's JMS-related settings are under the "Eventing" group under the main "External Integrations" category. For more information about System Settings, see *Oracle Fusion Middleware Integration Guide for Oracle Enterprise Repository*. Additional "Eventing" properties are described in Section A.4, "Configuring the Oracle Enterprise Repository Event Manager"

A.7.2.1 Enabling and Configuring an External JMS Server

The internal Apache ActiveMQ JMS Server needs to be disabled in order to configure an external JMS product. You must also configure JNDI and JMS properties for the external JMS.

Note: These steps are for configuring a single external JMS server. For instructions on configuring multiple JMS servers in a cluster, see Section A.7.4, "Configuring JMS Servers in an Oracle Enterprise Repository Cluster".

1. Click **System Settings** in the sidebar on the Oracle Enterprise Repository Admin screen.
2. Enter `Event` in the System Settings Search box to view all the Event Manager related settings.
3. Disable the internal JMS server property, `cmee.eventframework.embedded.jms.enabled`, by clicking False next to the Event Manager Embedded JMS Enable property. This forces the Event Manager to use an external JMS server.

   Note: The `cmee.eventframework.embedded.jms.enabled` property is not visible, by default. To enable this property, enter this property name in the Enable System Setting box, and then click **Enable**.

4. In the `eventing.properties` file, configure the required JNDI properties that are:
   - **JNDI URL**: `cmee.eventframework.jndi.provider.url`

     Specifies the JNDI URL. For example, t3://localhost:7001.
   - **JNDI User Name**: `cmee.eventframework.jndi.user`

     Specifies the JNDI user name.
   - **JNDI Password**: `cmee.eventframework.jndi.password`

     Specifies the password for the JNDI User Name.

- **JNDI Context Factory**: `cmee.eventframework.jndi.context.factory`

  Specifies the JNDI initial context factory. For example, `weblogic.jndi.WLInitialContextFactory`.

  Note: The JNDI properties are not visible, by default. To enable these properties, enter each property name in the Enable System Setting box, and then click **Enable**.

5. In the `eventing.properties` file, configure the following JMS properties:
   - **JMS Connection Factory**: `cmee.eventframework.jms.connection.factory`

     Specifies the JMS connection factory to enable JMS clients to create JMS connections. For example, `weblogic.examples.jms.TopicConnectionFactory`.
   - **JMS Topic**: `cmee.eventframework.jms.topic`

     Specifies the JMS topic, which is a publish/subscribe destination type for a JMS server. For example, `weblogic.examples.jms.TopicConnectionFactory`.

     Note: The JMS properties are all hidden, by default. To enable these properties, enter each property name in the Enable System Setting box, and then click **Enable**.
6. Click **Save**.
7. Restart Oracle Enterprise Repository for the configuration changes to take effect.

A.7.2.2 Configuring JMS Message Header Properties

Every JMS message contains a standard set of header fields that is included by default and available to message consumers. The Message Expiration and Delivery Mode headers can be configured using the Oracle Enterprise Repository System Settings.

1. Access the "Eventing" System Settings, as described in Section A.7.2.1, "Enabling and Configuring an External JMS Server".
2. In the `eventing.properties` file, configure the JMS message header properties:
   - **JMS Message Expiration**: Sets the JMS message expiration time in seconds. If set, unprocessed events will expire in the specified number of seconds. The default is 0 seconds, which means that messages will never expire. However, some environments have policies that require that JMS messages cannot be stored forever if they are not selected for some reason. Enable this property and set the property value to `cmee.eventframework.jms.expiration`.
   - **JMS Delivery Mode**: Sets the JMS message delivery mode to either PERSISTENT or NON-PERSISTENT values. If set to PERSISTENT, the JMS server will write the events to the underlying store. Although more reliable, persisting events to a store can affect performance. The default is PERSISTENT. Enable this property and set the property value to `cmee.eventframework.jms.deliverymode`.
3. Click **Save**.
4. Restart Oracle Enterprise Repository for the configuration changes to take effect.

A.7.2.3 Miscellaneous JMS Properties

Note: You must restart Oracle Enterprise Repository after changing any Eventing property in order for the changes to take effect.

The following miscellaneous System Settings can also be configured:

- **Event Manager JMS Subscribers Enabled**: If set to False, then the internal JMS subscribers will not be enabled. This is to make sure that the embedded JMS server is started, but an external tool can be used to connect to the embedded server using the given durable subscriber name and the stored events can be cleaned up.
- **JMS Subscribers Client ID**: Specifies the JMS durable subscriber ID. For example, ALER_JmsSubscriber. The property name is `cmee.eventframework.jms.subscribers.client.id`.
- **JMS Producers Client ID**: Specifies the JMS producer's client ID. For example, ALER_DeliveryManager. The property name is `cmee.eventframework.jms.producers.client.id`.
- **Lazy Initialize Event Engine**: When enabled, the Event Manager is initialized when an event is produced for the first time. This property should be enabled for either of the following reasons:
  - If there is a large number of events stored by the JMS server and if it is required that these events should not be processed as soon as Oracle Enterprise Repository is started.
  - There are startup issues that occur because of the timing of initializing the embedded JMS server.

  The property name is `cmee.eventframework.lazy.load`.

A.7.2.4 Configuring External JMS Jar Files

If an external JMS server is being used, then the external JMS server-related JAR files should be copied to the WEB-INF\lib directory.

A.7.3 Configuring the Embedded ActiveMQ JMS Server to Use a Database

By default, the ActiveMQ JMS server uses a file-based store to store events. However, you can specify to have events stored in a database. Simply, configure the activemq.xml file in the *WEB-INF\classes* directory to use your database parameters.

For example:

```
<persistenceAdapter>
   <journaledJDBC journalLogFiles="5" dataDirectory="../activemq-data"
 dataSource="#oracle-ds" />
   <!-- To use a different datasource, use the following syntax : -->
   <!-- <journaledJDBC journalLogFiles="5" dataDirectory="../activemq-data"
 dataSource="#postgres-ds"/> -->
 <!--  Oracle DataSource Sample Setup  -->
- <bean id="oracle-ds" class="org.apache.commons.dbcp.BasicDataSource"
 destroy-method="close">
  <property name="driverClassName" value="oracle.jdbc.driver.OracleDriver" />
  <property name="url" value="jdbc:oracle:thin:@localhost:1521:AMQDB" />
  <property name="username" value="scott" />
  <property name="password" value="*****" />
  <property name="poolPreparedStatements" value="true" />
```

```
</bean>
```

A.7.3.1 Configuring JMS Durable Subscribers for Web Service Endpoints

The Event Manager creates one durable subscriber for each Web Service endpoint it encounters in the Subscription Manager XML file. This ensures that events are stored if the endpoints are not online and that they can be reliably delivered once the endpoints are online again. As per the JMS Specification, the durable subscriber name should be unique across the JMS server. The Event Manager gets the durable subscriber name from the name field found in the `EndPointEventSubscription.xml` file, as shown in this example:

```
<sub:EventSubscriptionData
 xmlns:sub="http://www.bea.com/infra/events/subscription"
  <sub:eventSubscription>
    <sub:endPoint name="ALBPMEndpoint">
```

Note: JMS servers associate the durable subscriber name with the message selectors. Therefore, if the message selector is changed, either a new durable subscriber name should be provided or the existing one should be deleted. You can use the Oracle Enterprise Repository "Event Cleanup" tool, as described in "Cleaning Up Stored Events" on page 7-4. You can also use a JMS-specific tool to accomplish this.

A.7.4 Configuring JMS Servers in an Oracle Enterprise Repository Cluster

Note: For information about configuring Oracle Enterprise Repository in a clustered environment, see *Oracle Fusion Middleware Installation Guide for Oracle Enterprise Repository*.

A.7.4.1 Enabling JMS Clustering Mode

If Oracle Enterprise Repository is deployed on cluster mode, you must enable clustering on each Oracle Enterprise Repository instance regardless of which type of JMS server being used (embedded or external).

1. Click **System Settings** in the sidebar on the Oracle Enterprise Repository Admin screen.
2. Enter `cmee.eventframework.clustering.enabled` in the Enable New System Setting box and click **Enable** to reveal this hidden property.
3. Set the **Clustering Enabled** property to `True`.
4. Set other required properties based on the type of JMS server, as described in the following sections.

A.7.4.2 Configuring Embedded JMS Servers for Clustering

In a clustered environment, each member Oracle Enterprise Repository instance in the cluster will have one embedded JMS server. For example, in case of two-node cluster, there are two Oracle Enterprise Repository instances, such as server01 and server02, with each having one embedded JMS server. Once clustering is enabled for the embedded JMS servers, you then need to specify the connection URL information for the embedded JMS servers on server01 and server02.

1. Click **System Settings** in the sidebar on the Oracle Enterprise Repository Admin screen.
2. Enter `cmee.eventframework.embedded.jms.url` in the Enable New System Setting box and click **Enable** to reveal this hidden property.
3. In the Embedded JMS Server URL property, supply the connection URL for the embedded JMS servers on the clustered Oracle Enterprise Repository servers, using the following format.

   ```
   failover:(tcp:// $SERVER_DNS_NAME_OR_IP$:61700,tcp://$SERVER_DNS_NAME_OR_
   IP$:61700, …)
   ```

   where

   `$SERVER_DNS_NAME_OR_IP$` are replaced by actual server DNS name or IP address. The entries should be repeated for each Oracle Enterprise Repository server in a given cluster.

   Using the example above, this could be set to:
   failover:(*tcp://server01:61700,tcp://server02:61700*)

   Caution: Port 61700 is the default port for the embedded JMS server, and therefore should not be used by any other application on the Oracle Enterprise Repository server unless another port is configured for the embedded JSM server.

4. Click **Save**.
5. Repeat steps 1-4 for each Oracle Enterprise Repository instance in a given cluster. Using the example above, the Embedded Broker URLs could be set to:
   failover:(*tcp://server01:61700,tcp://server02:61700*)

   Tip: Ensure that each embedded JMS server is enabled by setting the cmee.eventframework.embedded.jms.enabled property to `True`.

A.7.4.3 Configuring External JMS Servers for Clustering

For external JMS servers, no additional configuration is required. However, you must make sure that the embedded JMS server is disabled, as follows:

1. Click **System Settings** in the sidebar on the Oracle Enterprise Repository Admin screen.
2. Set the **Event Manager Embedded JMS Enable** property to `False` (i.e., `cmee.eventframework.embedded.jms.enabled` is `False`.

A.7.5 Configuring a JMS Provider In WebSphere 6.1.0.5

When Oracle Enterprise Repository is deployed on WebSphere Application Server 6.1.0.5, the embedded Apache ActiveMQ JMS server cannot be used. Therefore, WebSphere 6.1.0.5 implementations must use the default JMS provider that comes with WebSphere 6.1.0.5.

To configure a JMS provider for Oracle Enterprise Repository in WebSphere 6.1.0.5, complete the following steps in the WebSphere administration console and in your Oracle Enterprise Repository application.

1. Create a new Service Integration Bus:

a. In the navigation pane, expand Service Integration, and then click **Buses**.

b. On the Buses page, click **New**.

c. On the Create a new bus page, enter `alerbus` as the name for the new bus.

d. Clear the **Bus security** option.

e. Click **Next**, and then click **Finish**.

2. Add a Bus member to the newly created alerbus:

a. On the Buses page, click the **alerbus** link.

b. Under the Topology category, click **Bus members**.

c. On the Bus members page, click **Add**.

d. On the Add a new bus member > Select Server, Cluster or WebSphere MQ server page, accept the default Server option and click **Next**.

e. On the Add a new bus member > Select the type of message store page, accept the default File store option and click **Next**.

f. On the Add a new bus member > Provide the message store properties page, accept the default values and click **Next**.

g. On the Add a new bus member > Confirmation page, click **Finish**.

h. On the Buses page, click **Save**.

3. Create a JMS Topic Connection Factory in the default message provider:

a. In the navigation pane, expand JMS, and then click **JMS providers**.

b. Click the **Default messaging provider** option, with a Scope of Node=<nodename>, server=server1.

c. On the JMS providers > Default messaging provider page, click the **Topic connection factories** option under Additional Properties.

d. On the JMS providers > Default messaging provider > Topic connection factories page, click **New**.

e. On the Administration page, configure the topic connection factory as follows:

*) **Name**: **alerEventingTopicCFDefault**

***) JNDI name**: **jms.alerEventingTopicCFDefault**

***) Bus name**: **alerbus**

***) Client identifier**: **ALER_JmsProducer**

***) Durable subscription home**: **<nodename>.server1-alerbus**

f. Click **Apply**, and then click **Save**.

4. Create a JMS Topic in the default message provider:

a. Re-navigate to the JMS providers > Default messaging provider page.

b. Click the **Topics** option under Additional Properties.

c. On the JMS providers > Default messaging provider > Topics page, click **New**.

d. On the Administration page, configure the topic as follows:

***) Name**: `alerEventingTopicCFDefault`

***) JNDI name**: `jms.alerEventingTopicCFDefault`

*) **Topic name**: `alerEventingTopicDefault`

*) **Bus name**: `alerbus`

*) **Topic space**: `Default.Topic.Space`

e. Click **Apply**, and then click **Save** to save your changes.

5. Deploy the `aler.ear` application file, as follows:
   a. In the navigation pane, expand Applications, and then click Enterprise Applications.
   b. On the Enterprise Applications page, click **Install**.
   c. On the Preparing for the application install page, click **Browse**, specify the aler.ear file in the path, and then click Next.
   d. Click **Next** on the Select installation options page.
   e. Click **Next** on the Map modules to servers page.
   f. On the Map resources to resource references page, click **Browse** in the Target Resource JNDI Name column.
   g. On the Enterprise application > Available resources page, select **alerEventingTopicCFDefault**, and then click **Apply**.
   h. Click **Next** in the ensuing Map resources to resource references page.
   i. On the Map resource environment entry references to resources page, enter jms/aler/alerEventingTopicDefault in Target Resource JNDI Name and then click **Next**.
   j. Click **Finish** on the Summary page.
   k. After the application is installed, click **Save** to save it to the Master Configuration.
6. Follow the "Manually Installing the Oracle Business Process Management Process Engine and Automated Workflows" steps in *Oracle Fusion Middleware Installation Guide for Oracle Enterprise Repository* to deploy additional files in the web-inf/classes directory and the database drivers required by the Oracle Enterprise Repository application.
7. Configure the Oracle Enterprise Repository eventing.properties file for the WebSphere settings:
   a. Navigate to the `<OER Domain>\WEB-INF\classes` directory.
   b. Use a text editor to modify the eventing.properties file as follows:

```
cmee.eventframework.jms.topic=jms.alerEventingTopicDefault
cmee.eventframework.jndi.provider.url=iiop\://localhost:2809
cmee.eventframework.embedded.jms.enabled=false
cmee.eventframework.jndi.context.factory=com.ibm.websphere.naming.WsnIniti
alContextFactory
cmee.eventframework.jms.connection.factory=jms.alerEventingTopicCFDefault
```

   c. Save the file.
8. Restart the WebSphere application server to enable the modified settings.
9. Check the WebSphere logs for possible errors:
   `\WebSphere\AppServer\profiles\AppSrv01\logs\server1`

A.8 Monitoring and Managing Events

This section discusses the steps to monitor and manage events in Oracle Enterprise Repository.

This section contains the following topics:

- Section A.8.1, "Overview"
- Section A.8.2, "Monitoring Events"
- Section A.8.3, "Cleaning Up Stored Events"
- Section A.8.4, "Configuring Workflow"
- Section A.8.5, "Encrypting Your Passwords"

A.8.1 Overview

This section describes how to use the administrative tools that are shipped as part of Oracle Enterprise Repository. The Advanced Registration Flow administrative tools are used to:

- Monitor events using a command-line interface
- Clean up the events and unsubscribe the JMS durable subscriber
- Generate the Workflow Configuration file
- Refresh the Oracle Business Process Management Engine with the latest Workflow Configuration file
- Encrypt the passwords stored in the Workflow Configuration and Subscription Manager files

The administrative tools are installed under the following directory:

```
<ORACLE_HOME>/repositoryXXX/core/workflow-tools
```

A.8.2 Monitoring Events

The Event Manager has a tool for monitoring the events that are generated by the Event Manager. The tool peeks into the event traffic and prints information, such as the Event Body and Event Properties, as shown in this section.

A.8.2.1 Prerequisites

The following prerequisites apply before starting the monitoring tool:

- If the default embedded JMS server is used, then Oracle Enterprise Repository needs to be running with the cmee.eventframework.enabled system setting set to true. This is to make sure that the JMS broker that is embedded within Oracle Enterprise Repository is running so that the monitoring tool can connect to it and monitor the events.
- If an external JMS server is used, then the external JMS Server needs to be running and the JNDI-related eventing.properties that are required to connect to the external JMS server must be configured.

For more information, see Section A.7.2, "Configuring Connectivity Properties for External JMS Servers".

A.8.2.2 Usage

From a command prompt, run the Event Monitoring tool as follows:

```
> event_monitor.bat <Path of WEB-INF\classes>
```

For example, if Oracle Enterprise Repository is deployed to a domain named oerdomain under: `D:\bea816\user_projects\domains\oerdomain`

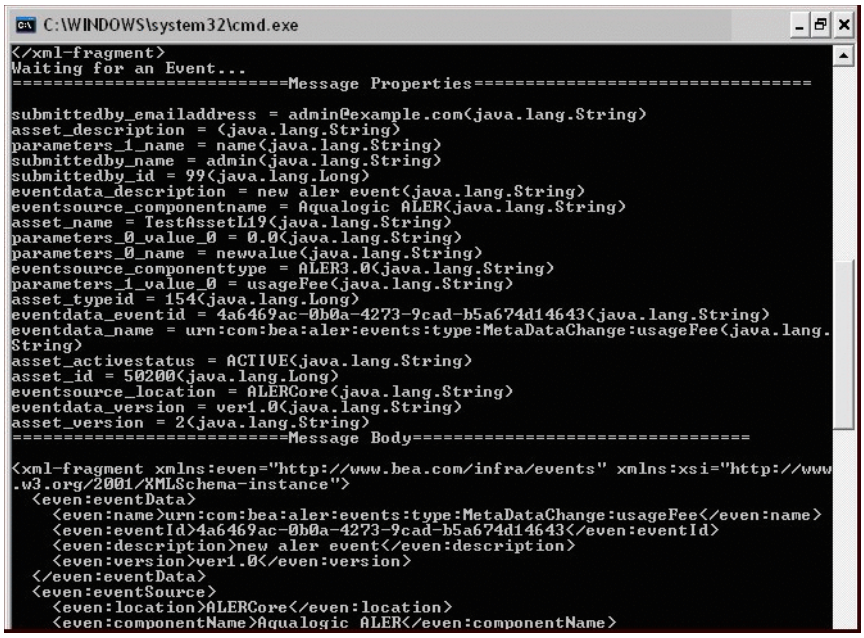
Then the <Path of WEB-INF\classes> is:

<Oracle_HOME>\user_projects\applications\base_domain\applications\oer_11.1.1.4.0\oer-app\WEB-INF\classes

where `Oracle_HOME` represents the `OER_HOME` install location.

This path is needed to get the JMS configuration from the `eventing.properties` file so that the tool can connect to the JMS server.

Figure A–23 Event Monitor Console



A.8.3 Cleaning Up Stored Events

Sometimes it may be required to remove all the events that are stored by the Event Engine and also unsubscribe the durable subscription. The Event Cleanup tool can be used for this purpose.

A.8.3.1 Prerequisites

The following prerequisites apply before starting this tool:

- Set the Oracle Enterprise Repository `cmee.eventframework.jms.subscribers.enabled` system setting to `false` so that the Oracle Enterprise Repository Event Manager does not start the durable subscriber because this is unsubscribed by the Clean Event tool.
- Restart Oracle Enterprise Repository with the `cmee.eventframework.jms.subscribers.enabled` property set to `false`.

A.8.3.2 Usage

From a command prompt, run the Event Cleanup tool as follows:

```
 > event_clean.bat <Path of WEB-INF\classes> <Name of Durable Subscriber> <Message
Selector>
```

For example, if Oracle Enterprise Repository is deployed to a domain named oerdomain under: `D:\bea816\user_projects\domains\oerdomain`

Then the <Path of WEB-INF\classes> is:

<Oracle_HOME>\user_projects\applications\base_domain\applications\oer_11.1.1.4.0\oer-app\WEB-INF\classes

This path is needed to get the JMS configuration from eventing.properties so that the tool can connect to the JMS Server.

The `<Name of Durable Subscriber>` can be found in the name attribute inside the endpoint that requires event cleanup within the EndPointEventSubscription.xml as follows:

```
<sub:eventSubscription>
  <!--The name should be unique within this file since
<sub:endPoint name="ALBPMEndpoint">
```

The `<Message Selector>` can be found in the expression attribute inside the endpoint that requires cleanup within the `EndPointEventSubscription.xml` file.

> **Note:** The parameter can be omitted if the Message selector is not set or empty.

A.8.3.3 Sample Event Cleanup

Using the example above, navigate to the workflow-tools directory:

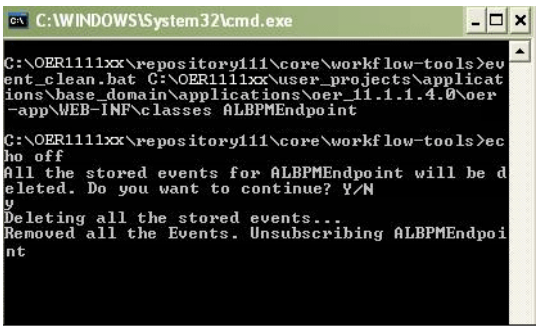
```
> cd D:\bea816\repositoryXXX\core\workflow-tools>
```

From the command prompt, type:

```
> event_clean.bat <ORACLE_HOME>\user_projects\applications\base_
domain\applications\oer_11.1.1.4.0\oer-app\WEB-INF\classes ALBPMEndpoint
```

The following is the output printed by the Event Cleanup tool to the console.

Figure A–24 Event Cleanup Console



A.8.4 Configuring Workflow

The Generate Workflow Config tool is used to generate the Workflow Configuration file (`workflow.xml`) by connecting to Oracle Enterprise Repository. The tool populates the `workflow.xml` with configuration for asset types, categorizations, etc. by reading these entities from Oracle Enterprise Repository. The Workflow Config file can then be customized as per your requirements. For example, you may need to configure and customize flows to add new asset types, projects, categorizations, etc.

For more information about configuring Automated Workflows, see Section A.6, "Configuring Automated Workflows".

From a command prompt, run the Generate Workflow Config tool as follows:

```
> config_gen.bat URI User Password ConfigDir
```

where

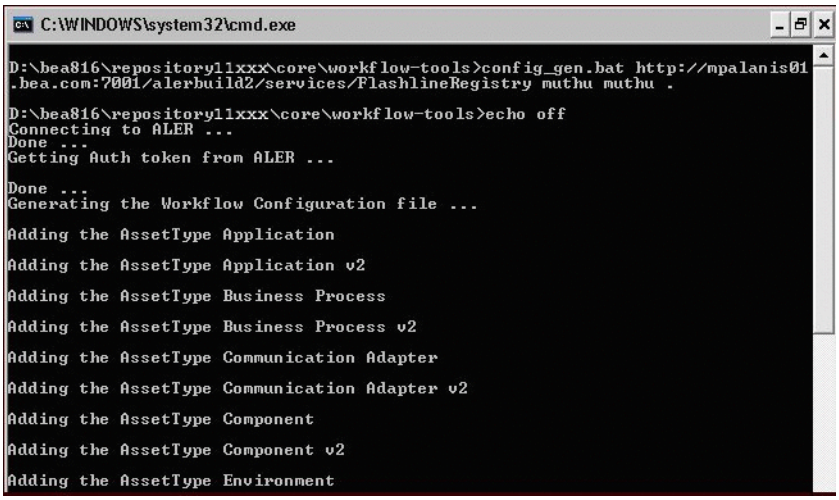
URI = OER URI (for example: *http://localhost:7001/alerbuild/services/FlashlineRegistry*)

User = OER user name

Password = OER password

ConfigDir = the directory where the Config XML file is created. If the file exists, it is renamed to `workflow.xml.bak`.

Figure A–25 Generate Workflow Configuration Tool



The `workflow.xml` file needs to be generated to the following directory:

<OBPM Enterprise Home>/server/<OER Workflows Project>/workflow.xml

A.8.4.1 Refreshing the Workflow Config File

The Refresh Workflow Config XML tool lets you to refresh a Workflow Config file without restarting the Oracle Business Process Management Engine. For example, if the Workflow Config XML file is updated during development, running this tool allows the Oracle Business Process Management Engine to use the updated version without restarting the engine.

Note: The Oracle Business Process Management Engine must be running when running this tool.

From a command prompt, run the Refresh Workflow Configuration tool as follows:

```
> refresh_workflows.bat URI User Password
```

where

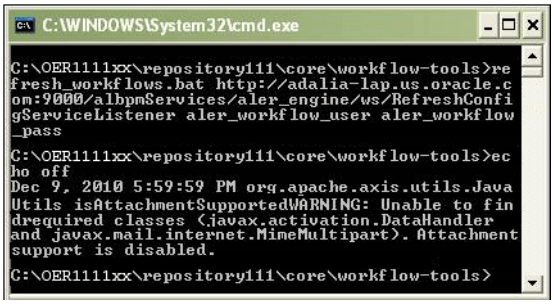
URI = Oracle Business Process Management URI (for example, *http://localhost:9000/albpmServices/aler_engine/ws/RefreshConfigServiceListener*)

User = Oracle Business Process Management user name (for example, aler_workflow_user)

Password = Oracle Business Process Management password (for example, aler_workflow_pass)

Note: aler_workflow_user is created by Oracle Products Installer and is the default user that can be used with this tool.

Figure A–26 Refresh Workflow Configuration Tool



A.8.5 Encrypting Your Passwords

For enhanced security, the Security Encrypt Password tool (`runWfSecurity.bat`) allows you to encrypt passwords that are stored in the Workflow Configuration and Subscription Service files.

From a command prompt, run the Security Encrypt Password tool as follows:

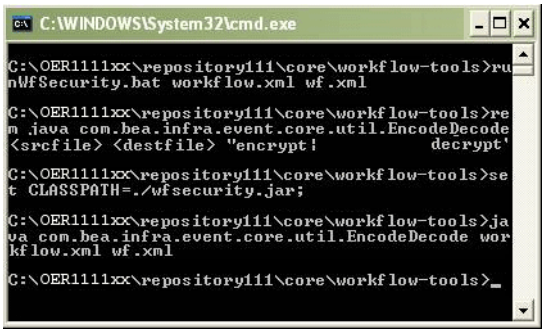
```
> runWfSecurity.bat srcFileName destFileName
```

where

srcFileName = source config file with clear password.

destFileName = destination config file with encrypted password.

Figure A–27 Security Encrypt Password Tool



A.9 Extending the Event Manager for Web Service Endpoints

This section describes how to extend the event manager for Web Service endpoints.

This section contains the following topics:

- Section A.9.1, "Overview"
- Section A.9.2, "Developing a Web Service Endpoint"
- Section A.9.3, "Web Service Operations"
- Section A.9.4, "Developing an Endpoint with an Incompatible Contract"

A.9.1 Overview

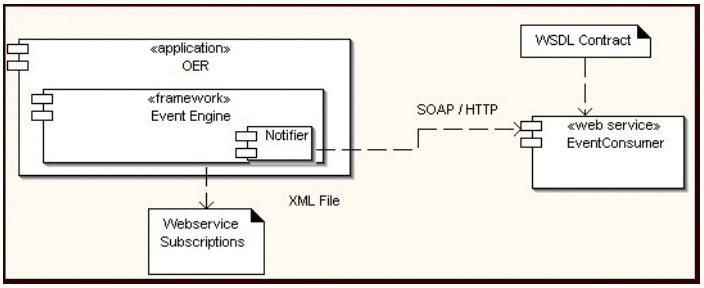
This section explains how to develop a new Web Service endpoint to consume the events that are emitted by the Event Manager and also explains how to extend the Event Manager to use other notifier plug-ins.

For information about configuring the Event Manager, see Chapter A.4, "Configuring the Oracle Enterprise Repository Event Manager".

A.9.2 Developing a Web Service Endpoint

Figure A–28 shows how a Web Service endpoint can be plugged-in to receive the Events emitted by the Oracle Enterprise Repository Event Manager.

Figure A–28 Web Service Endpoint Plug-in



Perform the following steps to create a new Web Service endpoint and start getting events:

1. Pick up the WSDL contract defined by the Event Manager. This is bundled with the `modules.eventing-11.1.1.x.x.jar` file located in the `<oer Webapp path>/WEB-INF/lib` directory.
2. Open the jar file and locate a WSDL named "**EventListener.WSDL**" and extract the WSDL to the file system. This WSDL is the abstract contract defined by the Event Manager and the new Web Service endpoint needs to implement the operation defined in the WSDL.

   Figure A–29 shows a snapshot of the WSDL file:

Figure A–29 Sample WSDL File

```
<?xml version="1.0" encoding="UTF-8" ?>
<!-- edited with XMLSpy v2007 sp2 (http://www.altova.com) by X Y (XYZ)  -->
<wsdl:definitions xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/" xmlns:tns="http://www.bea.com/aler/events/alerE
  xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:aler="http://w
  xmlns:ns="http://www.bea.com/infra/events" xmlns:ns1="http://schemas.xmlsoap.org/soap/encoding/"
  xmlns:wfStatus="http://www.bea.com/infra/events/workflow/status" name="AlerEventsListener"
  targetNamespace="http://www.bea.com/aler/events/alerEventsListenerWsdl/">
- <wsdl:types>
  - <xs:schema targetNamespace="http://www.bea.com/infra/events" xmlns:commonEvents="http://www.bea.com/infra/eve
      xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns="http://www.w3.org/2001/XMLSchema" elementFormDefault=
      attributeFormDefault="unqualified">
    - <xs:annotation>
        <xs:documentation>Defines the common events structure</xs:documentation>
      </xs:annotation>
      <xs:element name="Event" type="commonEvents:EventType" />
    - <xs:complexType name="EventType">
      - <xs:sequence>
          <xs:element name="eventData" type="commonEvents:EventInfoType" minOccurs="1" maxOccurs="1" />
          <xs:element name="eventSource" type="commonEvents:EventSourceType" minOccurs="1" maxOccurs="1" />
          <xs:element ref="commonEvents:ExtendedData" minOccurs="0" maxOccurs="1" />
        </xs:sequence>
      </xs:complexType>
    - <xs:complexType name="EventInfoType">
      - <xs:sequence>
          <xs:element name="name" type="xs:string" />
          <xs:element name="eventId" type="xs:string" />
          <xs:element name="description" type="xs:string" />
          <xs:element name="version" type="xs:string" />
        </xs:sequence>
```

3. Complete the Web Service endpoint development using the tool or technology, as per the requirement. For example, you could develop a Proxy Service using Oracle Service Bus, which provides a feature where you can create a Web Service-based proxy service by pointing to a WSDL file. Make the Web Service running by completing the development of the Web Service.
4. Configure the Event Manager so that the Web Service endpoint's host, port, and URI, etc., are entered in the Subscription Manager file. For more information about configuring the Event Manager, see Chapter A.4, "Configuring the Oracle Enterprise Repository Event Manager".
5. Start Oracle Enterprise Repository and trigger events using the Asset Editor and the Web Service endpoint will start getting the Events.
6. You can use the Event Monitoring tool that is bundled with Oracle Enterprise Repository for debugging and monitoring the Events that are generated by the Event Manager.

A.9.3 Web Service Operations

This section describes the available operation for a new Web Service endpoint, and how to specify operations in the Event Manager.

A.9.3.1 Available Web Service Operations

The Oracle Enterprise Repository Event Manager supports the following operations:

newEventRequestResponse

This operation takes the event object that is defined in the XML schema section as an input and returns the status as the output. The status is defined as string type. Additionally, if the status string starts with Failure, then the Event Manager will throw an exception and will try to re-deliver the event until it succeeds. If not, it will log the response and will deliver the next event unless there is a transport exception.

newEventRequestResponseString

This operation takes the event data in string form as an input and returns the status as the output. The status is defined as string type. Additionally, if the status string starts with Failure, then the Event Manager will throw an exception and will try to re-deliver the event until it succeeds. If not, it will log the response and will deliver the next event unless there is a transport exception.

newEventRequest

This operation takes the event object that is defined in the XML schema section as an input and is defined as a one-way operation.

newEventRequestString

This operation takes the event data in string form as an input and is defined as a one-way operation.

newEvent

This operation should be used only if the Process Engine is Oracle Business Process Management. This operation internally invokes the startSession operation to start session to authenticate with Oracle Business Process Management. It will also call discardSession after the invocation.

A.9.3.2 Selecting a Web Service Operation

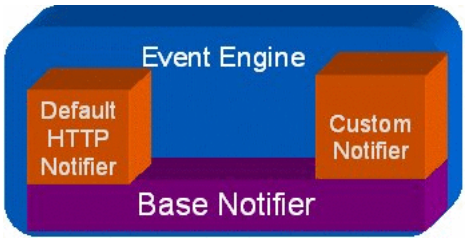
The preferred Web Service operation can be selected by configuring the Event Manager's Subscription Manager the following way, as specified in the operationName element.

```
<sub:EventSubscriptionData
xmlns:sub="http://www.bea.com/infra/events/subscription"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
      <sub:eventSubscription>
           <sub:endPoint name="ALBPMEndpoint">
            <sub:host>localhost</sub:host>
            <sub:port>9000</sub:port>
            <sub:uri>albpmServices/aler_
                engine/ws/StatusChangeEndpointServiceListener</sub:uri>
        <sub:targetNamespace>http://www.bea.com/infra/events</sub:targetNamespace>
        <sub:operationName>newEvent</sub:operationName>
        <sub:protocol>HTTP</sub:protocol>
        <sub:authenticationData>
          <sub:basicAuthentication>
                <sub:username>aler_workflow_user</sub:username>
                <sub:password>*****</sub:password>
          </sub:basicAuthentication>
       </sub:authenticationData>
     </sub:endPoint>
<sub:notifierClass>com.bea.infra.event.notifier.plugin.http.DefaultHTTPEventNotifi
er</sub:notifierClass>
<sub:expression></sub:expression>
 </sub:eventSubscription>
 </sub:EventSubscriptionData>
```

A.9.3.3 Developing a Notifier Plug-in

The Oracle Enterprise Repository Event Manager includes a default SOAP/HTTP notifier. A new plug-in can be developed and plugged in if there are additional requirements, as shown in Figure A–30.

Figure A–30 Notifier Plug-in



Follow these steps to make the new plug-in work with the Event Manager.

1. Develop a new Notifier Plug-in by extending the Java Class AbstractEventNotifier that is bundled with the Oracle Enterprise Repository Event Manager. This class is bundled with the modules-eventing-<oer.version>.jar located in the <oer Webapp path>-app/WEB-INF/lib directory. The init() and sendNotification() methods need to be overridden.

   Refer to the Javadoc (EventManagerPublicAPIdoc.zip) for more information about these methods. The handle() method passes the event data in an XML Beans format, which can be sent to an external Web Service.

2. Configure the Subscription Manager file to point to the developed class. Modify the notifierClass element as follows:

```
<sub:EventSubscriptionData
xmlns:sub="http://www.bea.com/infra/events/subscription"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
      <sub:eventSubscription>
           <sub:endPoint name="ALBPMEndpoint">
            <sub:host>localhost</sub:host>
            <sub:port>9000</sub:port>
            <sub:uri>albpmServices/aler_
                engine/ws/StatusChangeEndpointServiceListener</sub:uri>

<sub:targetNamespace>http://www.bea.com/infra/events</sub:targetNamespace>
        <sub:operationName>newEvent</sub:operationName>
        <sub:protocol>HTTP</sub:protocol>
        <sub:authenticationData>
          <sub:basicAuthentication>
                <sub:username>aler_workflow_user</sub:username>
                <sub:password>*****</sub:password>
          </sub:basicAuthentication>
       </sub:authenticationData>
     </sub:endPoint>
<sub:notifierClass>com.bea.infra.event.notifier.plugin.http.DefaultHTTPEventNot
ifier</sub:notifierClass>
<sub:expression></sub:expression>
 </sub:eventSubscription>
 </sub:EventSubscriptionData>
```

3. Bundle the classes in a JAR file and copy it to *<oer Webapp path>/WEB-INF/lib* directory so that it is picked up by the classpath.
4. Restart the Event Manager and trigger an event using the Asset Editor.
5. The Event Manager will call the init() and handle() methods of the new notifier plug-in.

A.9.4 Developing an Endpoint with an Incompatible Contract

It is possible that there may be an endpoint with an Interface or Contract that is not compatible with Oracle Enterprise Repository Event Manager. This is because the tool that is used to develop the endpoint may have restrictions to use the WSDL provided by Oracle Enterprise Repository Event Manager, or there may be other inter-operability issues. The following approach can be used under those circumstances:

- Develop an event notifier plug-in to receive the event XML data and register with the Subscription Manager.
- Write the code in the new notifier plug-in that transforms the event data into the format that the remote Web Service expects.
- Invoke the remote Web Service by whatever API is supported by the remote endpoint.

A.9.5 Creating Custom Workflows in OBPM 11g

Users who wish to write their own custom workflows can interact with OER's REX API. In the workflow designer in Jdeveloper, when invoking the REX web service, use

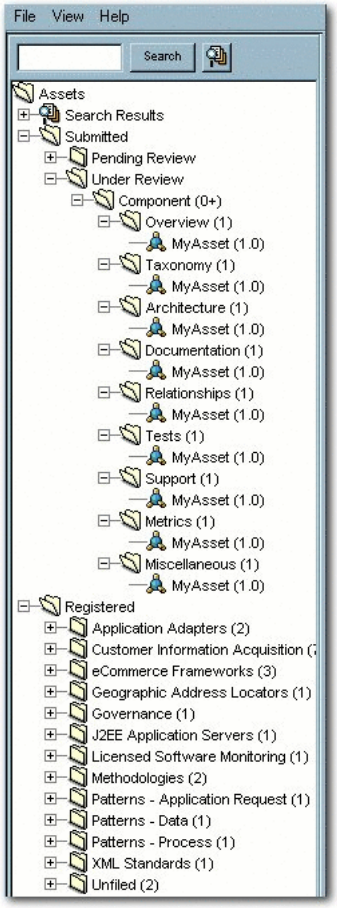
```
http://<host>:<port>/<oer web app name>/services/RexAPI
```

A.10 Working with OER Workflow

The Asset Editor is used to administer assets and configure the resources associated with assets, such as value lists, categories, relationships, rejection reasons, artifact stores, and vendors. The Asset Editor is typically launched from the Web console.

Assets moving through the registration process are organized and managed via several folders, as displayed in the file tree in the Asset Editor shown in Figure A–31.

Figure A–31 Asset Editor - File Tree



An asset begins the registration process in the Pending Review folder, under the Submitted folder. 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 accept it, which enters it into the work queues. The Registrar may also choose to reject the asset.

Rejection

- If the asset is rejected, the registrar enters a reason for the rejection.

- When an asset is rejected from the submission folder, it is removed from the Asset Editor and marked as rejected in the submitter's My Submissions folder in My Stuff.

Acceptance

Once an asset is accepted, it can be assigned to one or more Registrars for processing and approval. Registrars can view assets assigned to them from My Stuff. 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 contains the following topics:

- Section A.10.1, "Automated Asset Registration Using Automated Workflows"
- Section A.10.2, "Accepting a Submission"
- Section A.10.3, "Registering an Asset"
- Section A.10.4, "Completing the Tab Approval Process"
- Section A.10.5, "Audit Log, Reviews, and Notes"

A.10.1 Automated Asset Registration Using Automated Workflows

The Oracle Enterprise Repository bundles pre-built BPM asset registration flows that attempt to automate the registration and governance processes defined in Understanding the Registration Process. For ease of use, you can use the predefined OBPM endpoint or create your own Web Service endpoints to subscribe to Oracle Enterprise Repository events. You may use the OBPM Designer to create new Governance workflows. There are also event monitoring and logging tools for troubleshooting and tuning purposes.

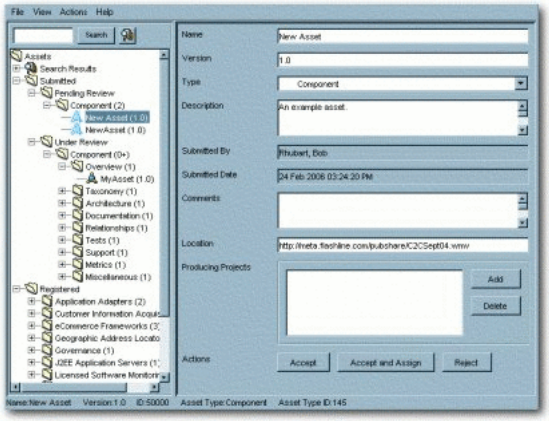
For more information about using the Automated Workflows feature, see Section A.6, "Configuring Automated Workflows".

A.10.2 Accepting a Submission

This procedure is performed in the Asset Editor. (Requires appropriate permission.)

1. Open the Submitted folder.
2. Open Pending Review.
3. Select the asset to be registered, as shown in Figure A–32.

Figure A–32 Asset Editor



4. Options:

   a. 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.

   b. Click **Accept and Assign**. The Assigning Users dialog is displayed, as shown in Figure A–33.

Figure A–33 Assigning Users Dialog

5. Use the << and >> buttons to move items between the Available Users and Selected Users columns.

6. 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. The assignees also may receive a notification e-mail that lets them know they are assigned to this asset.

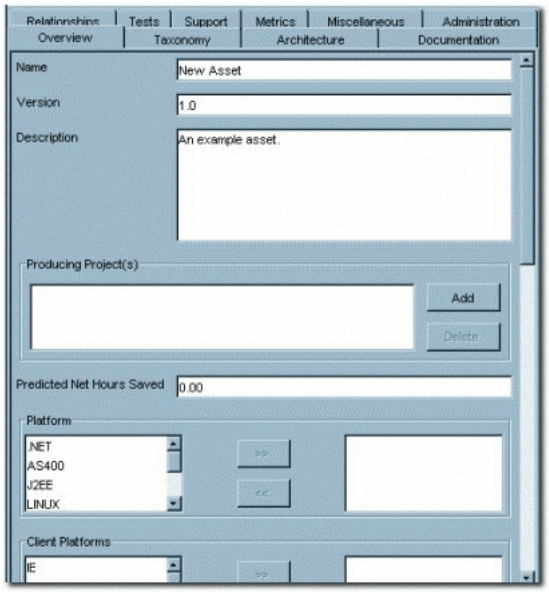
A.10.3 Registering an Asset

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.

Overview Tab

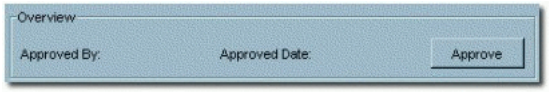
1. Click the **Overview** tab. The Overview tab is displayed, as shown in Figure A–34.

Figure A–34 Overview Tab



2. Enter the appropriate information in each of the fields.
3. Click **Approve**, as shown in Figure A–35

Figure A–35 Overview Tab - Approve Button



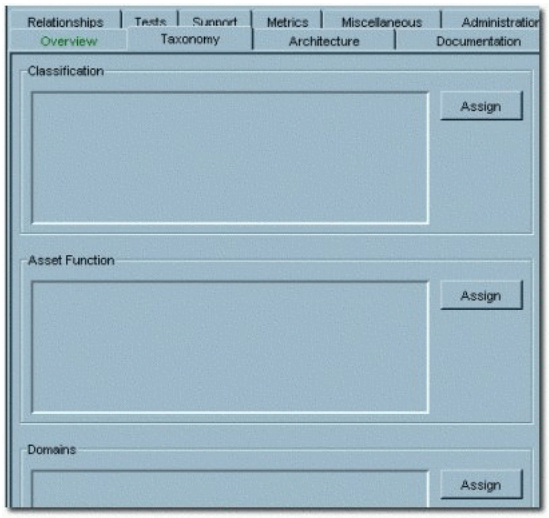
The text in the Overview tab changes color and the Approve button changes to Unapprove.

Note: Approval buttons in the Asset Editor are visible only to users with the appropriate permissions.

Taxonomy Tab

1. Click the **Taxonomy** tab. The taxonomy tab is displayed, as shown in Figure A–36.

Figure A–36 Taxonomy Tab



Several Categorizations are displayed.

2. Click the **Assign** button in the Classification section. The Assign Classifications dialog is displayed.
3. Use the options to select the appropriate classification.
   - Approved

     Approved by the Registrar for use in projects.
   - Educational

     For educational/informational purposes only. Not approved for use in projects.
   - 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.).
   - Raw

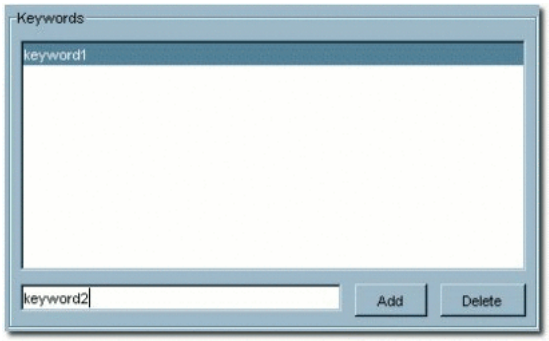
     No assurance of quality or completeness.
   - Recommended

     Approved and successfully deployed in at least one project.
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 an appropriate keyword in the text box in the Keywords section.
7. Click **Add**. The new keyword appears in the Keywords list, as shown in Figure A–37.

Figure A–37 Keywords List



Add other keywords as necessary.

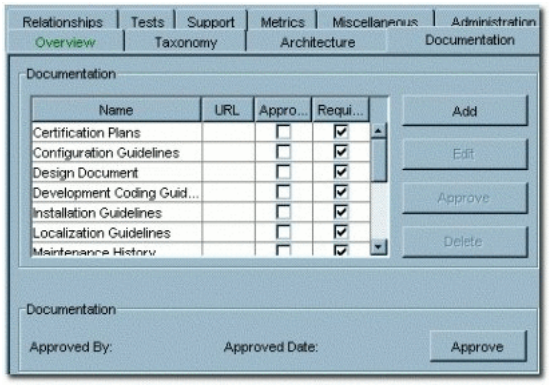
8. When the Taxonomy tab is completed, click **Approve**.

   The text in the Overview tab changes color and the Approve button changes to Unapprove.

Documentation Tab

1. Click the **Documentation** tab. The documentation tab is displayed, as shown in Figure A–38.

Figure A–38 Documentation Tab

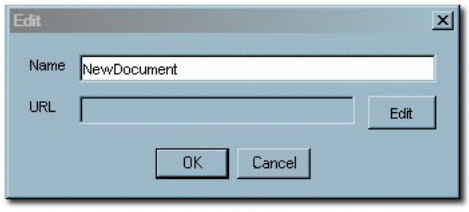


A number of suggested document titles are listed in the Documentation window. Appropriate documentation may be associated with each of these titles, and new documents may be added to the list.

To add a new document:

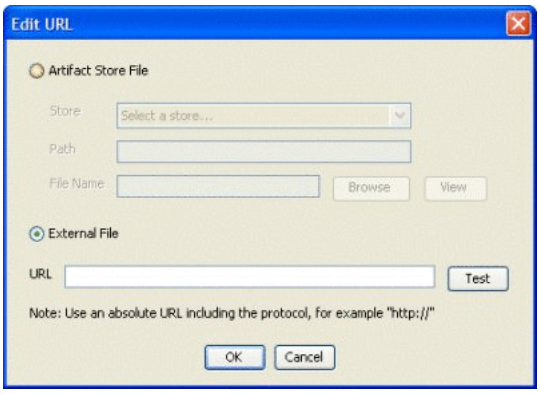
2. Click **Add**. The Edit dialog is displayed, as shown in Figure A–39.

Figure A–39 Edit Dialog



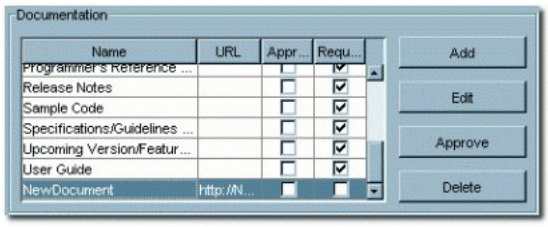
3. Enter the appropriate information in the Name text box.
4. Click the **Edit** button next to the URL text box. The Edit URL dialog is displayed, as shown in Figure A–40.

Figure A–40 Edit URL Dialog



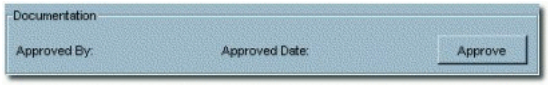
5. Select **Artifact Store File** or **External File**, as appropriate.
6. Enter all necessary information in the available text boxes.
7. Click **OK**. The new document appears in the list, as shown in Figure A–41.

Figure A–41 Documentation Tab - NewDocument



8. To edit file information for an existing document, select the document, click **Edit** and repeat Steps 4 to 7.
9. When finished, click **Approve**, as shown in Figure A–42.

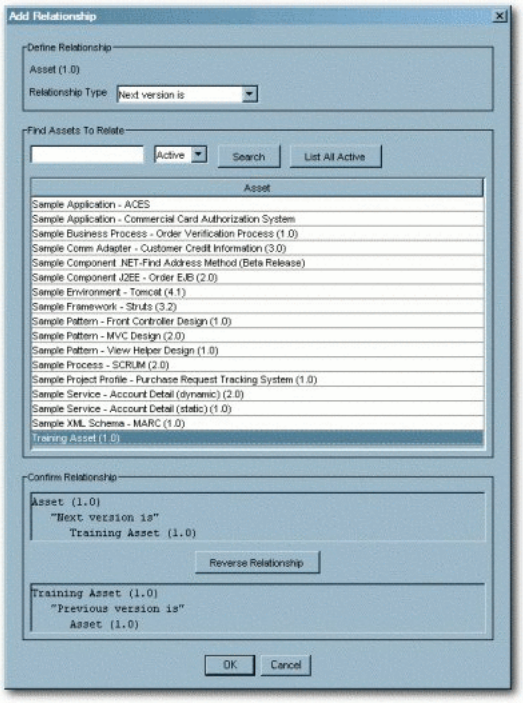
Figure A–42 Documentation Tab - Approve Button



Relationships Tab

1. Click the **Relationships** tab.
2. Click the **Add** button. The Add Relationship dialog is displayed, as shown in Figure A–43.

Figure A–43 Add Relationship Dialog



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.

   Note: If necessary, click the **Reverse Relationship** button to reverse the relationship.

6. Click **OK** when finished.
7. Repeat as necessary to add other asset relationships.
8. When finished, click **Approve**.

   See the Oracle Enterprise Repository Administration Guide for information on configuring relationships.

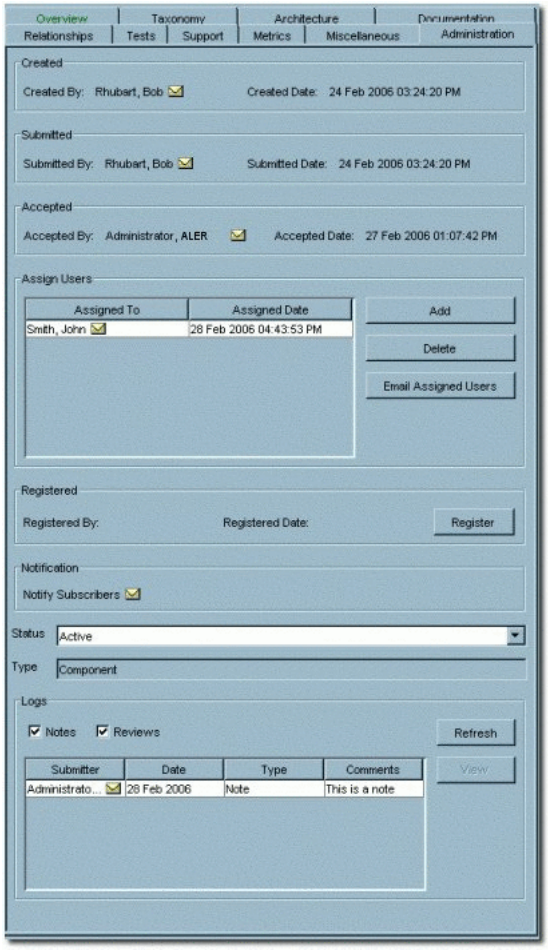
Administration Tab

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:
  - Active
  - Inactive

  - Retired
  - Deleted
- View asset notes and reviews.
- Complete the registration process by clicking the **Register** button, as shown in Figure A–44.

Figure A–44 Administration Tab



A registered asset can include unapproved tabs.

Whenever a tab is approved, a TabApprovedEvent is triggered. This event along with the tab name and certain other metadata is sent to the workflows for processing. This event is used by the Multi-Tier workflows where depending on the tab approved and depending on the configuration, the workflows will decide if the next tier in the approval process needs to be assigned or not.

Also the users can wire the TabApprovedEvent to any of the pre-defined workflows such as ChangeAssetLifecycle etc. So, this is also tied to the Tab names.

A.10.4 Completing the Tab Approval Process

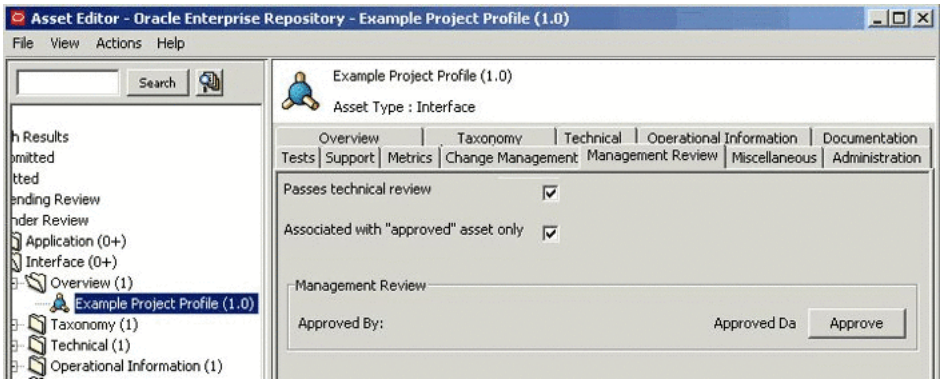
While certain tabs are common to all asset types, the specific Asset Editor tabs for any asset are determined by the configuration of the Type template to which the asset is assigned on submission. Similarly, the metadata elements that appear on any tab are

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. Each time a you click the Approve on a tab, you are prompted to save the changes.

Process to approve a tab for a specific type configuration

1. In the Oracle Enterprise Repository screen, click the **My Stuff** link. The My Stuff section is displayed.
2. Select Assigned Assets and then select the asset for which you would want to approve the tab(s).
3. Click the **Edit** button in the right-bottom pane of the page. The Asset Editor dialog is displayed.
4. Select the tab that you want to approve.
5. Review the information in that tab.

Figure A–45 Asset Editor Window



6. Click the **Approve** button. You are prompted to save the changes you made.
7. Click **OK** in the Select an Option dialog.
8. Go back to the Assigned Assets page and refresh to note the changes. In the Tab Approval Status pane, you will notice that Approved has one item against it, as shown in Figure A–46.

Figure A–46 Approved Status Pane

A.10.5 Audit Log, Reviews, and Notes

When an Asset is updated, a record of the User, date and action appears in the Audit Log. Logged changes include:

- Asset Creation
- Asset Update

- Changes in an Asset's Registration Status
- Changes in an Asset's Active Status
- Completion of an approval tab

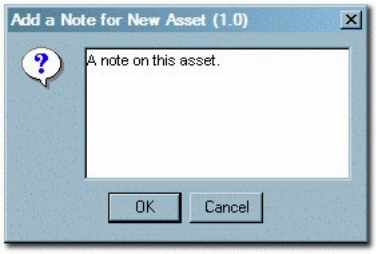
The log entry is added to the list in the Logs section on the asset's Administration tab. (It may be necessary to click the **Refresh** button in the Logs section.)

> **Note:** When changes are made to a custom data field, a log is created indicating that a field has changed. However, the specific details of that change are empty.

Adding a Note to the Asset

1. Click **Add Note** in the File menu. The Add a Note for... dialog is displayed, as shown in Figure A–47.

Figure A–47 Add a Note for ... Dialog



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.)

A.11 Getting Started with Automated Workflows

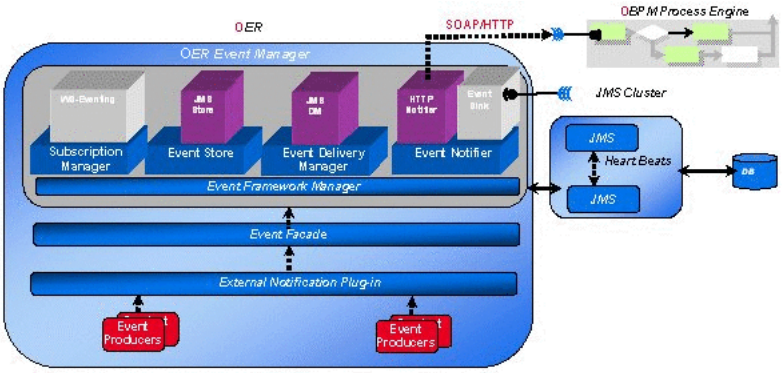
This section will help you to quickly get started using the Advanced Registration Flow feature using the bundled Oracle Business Process Management Web Service endpoint that is configured to work with the Oracle Business Process Management Process Engine. However, this feature is highly extensible and can be tailored to suit your environment.

A.11.1 Steps to Configure the Oracle Enterprise Repository Event Manager

The Event Manager is a component embedded within Oracle Enterprise Repository that manages event subscriptions, event persistence, event filtering, and event delivery. Web Service endpoints can subscribe to the Event Manager's Subscription Manager and the events that are generated within Oracle Enterprise Repository are delivered to the Web Service endpoints.

Figure A–48 shows the different components that are involved.

Figure A–48 Advanced Registration Flow Components



The Event Manager uses an embedded version of Apache ActiveMQ JMS Server that is enabled by default. The embedded JMS server is configured to run out-of-the-box without any additional configuration. However, you can also configure the Event Manager to use an external Java-based message broker, such as Weblogic Server JMS or IBM MQSeries.

For more information about configuring the Event Manager, see Section A.4, "Configuring the Oracle Enterprise Repository Event Manager".

A.11.2 Use Cases

This section describes the use cases for configuring Oracle Enterprise Repository Event Manager:

- Oracle Enterprise Repository features pre-bundled Oracle Business Process Management flows and a Web Service endpoint that is by default registered with the Event Manager's Subscription Manager. All the triggered events are delivered to this Oracle Business Process Management endpoint, which then attempts to automate Oracle Enterprise Repository processes, such as the asset registration process, tracking metadata changes, and taking pre-defined actions.
- You can also write your own Web Service endpoints, subscribe them with the Event Manager, and start getting the events to solve your specific business needs.

A.11.3 Steps to Configure and Run the Oracle Business Process Management Process Engine

When Oracle Enterprise Repository is installed, you are directed to install and configure Oracle Business Process Management. This section assumes that Oracle Business Process Management was successfully installed.

After the Event Manager is ready to send events, the Oracle Business Process Management Process Engine needs to be configured and be ready to process the events. When Oracle Enterprise Repository is installed, it provides an option to install and configure the Process Engine. This section assumes that the Process Engine was successfully installed before following these steps.

To launch the Oracle Business Process Management Admin Center, double-click the **obpmadmcenter.exe** file in the *<OBPM Enterprise Home>\bin* directory.

A.11.3.1 Use Cases

Oracle Enterprise Repository features pre-bundled Automated Workflows that are deployed to the Process Engine. When events are triggered within Oracle Enterprise

Repository, they are delivered to the Process Engine and execute the Automated Workflows that attempt to automate Oracle Enterprise Repository processes, such as asset submission, acceptance, registration, and other governance process.

For more information about the available Automated Workflows, see Section A.6, "Configuring Automated Workflows".

A.11.3.2 Configuring the Automated Workflows to Process a Submission Event

Follow these steps after the Oracle Business Process Management Process Engine is installed.

1. Generate the Workflow Configuration (`workflow.xml`) file using the Generate Workflow Config tool (`config_gen.bat`). This tool connects to Oracle Enterprise Repository and creates a bootstrapping file that can be customized. For more information about generating the workflow.xml file, see Section A.8.4, "Configuring Workflow".
2. Encrypt the Workflow Configuration (`workflow.xml`) file. For more information, see Chapter 5.3.3, "Workflow Configuration File".
3. Copy the newly generated `workflow.xml` file to the `<OBPM Enterprise Edition>/enterprise/server/aler_engine` directory.
4. Open the `workflow.xml` file using the XML editor of choice.
5. Ensure that the Oracle Enterprise Repository Connection information, such as the URI and the registrar user name/password, are configured correctly as shown here.

```
<alerconnection>
 <uri>http://server01.amer.bea.com:7005/aler/services/FlashlineRegistry </uri>
     <registrar>
       <user>admin</user>
       <password>*****</password>
     </registrar>
   </alerconnection>
```

   The URI must use the following format:

   http://<host>:<port>/<oer web app name>/services/FlashlineRegistry

6. Within the `workflow.xml` file, locate the assetType settings for the "Service" asset type, as shown here.

```
<assetType name="Service" community="_CHANGE_COMMUNITY_" id="154">
     <allTabs>
     <allTabs>
       <tab name="Oveview"/>
       <tab name="UDDI: Business Entity"/>
       <tab name="Taxonomy"/>
       <tab name="Architecture"/>
     </allTabs>
```

7. Add the autoAccept attribute and set the value to true, as shown here.

```
<assetType name="Application" community="_CHANGE_COMMUNITY_" id="154"
 autoAccept="true">
     <allTabs>
     <allTabs>
       <tab name="Oveview"/>
       <tab name="UDDI: Business Entity"/>
       <tab name="Taxonomy"/>
```

```
    <tab name="Architecture"/>
  </allTabs>
```

Now the Oracle Business Process Management Process Engine is configured to automatically accept any asset of type "Service."

8. If the Oracle Business Process Management Process Engine is running, stop it and then restart it to load the latest `workflow.xml` changes.

9. The Refresh Workflow Configuration tool can be used to refresh the workflow.xml file without restarting the Oracle Business Process Management Process Engine. For more information about refreshing the `workflow.xml` file, see Section A.8.4.1, "Refreshing the Workflow Config File".

A.11.4 Triggering an Asset Submission Event

Once the Oracle Business Process Management Process Engine is configured and running, follow these steps to process an asset submission event.

1. Launch the Oracle Enterprise Repository Asset Editor from the Web console.

   For information about using the Oracle Enterprise Repository Asset Editor, see *Oracle Fusion Middleware User's Guide for Oracle Enterprise Repository*.

2. Create a new asset from File ->New, as shown in Figure A–31.

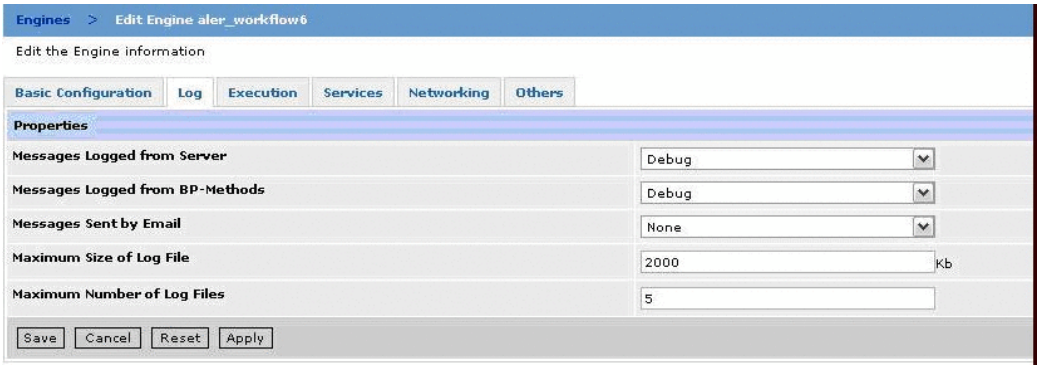
   Note: The Asset Type should be Service.

3. Click **OK** to submit the asset.

4. After the asset is submitted, switch to the Oracle Business Process Management Log Viewer to ensure that the event is processed. To launch the Log Viewer, double-click the **obpmlogviewer.exe** file in the *<OBPM Enterprise Home>\bin* directory.

5. Turn on the "Debug" level on the Log page of the Process Engine using the Process Administrator preference settings, as shown in Figure A–49. By default, the level is set to "Warning."

Figure A–49 Oracle Business Process Management Process Administrator - Logging Preferences



6. When you turn on the Debug level though you will notice that the Process Engine prints a lot of information, not just for the Oracle Enterprise Repository Automated Workflows, but other Process Engine information as well. To filter the Oracle Enterprise Repository logging, follow these steps, as shown in Figure A–50:

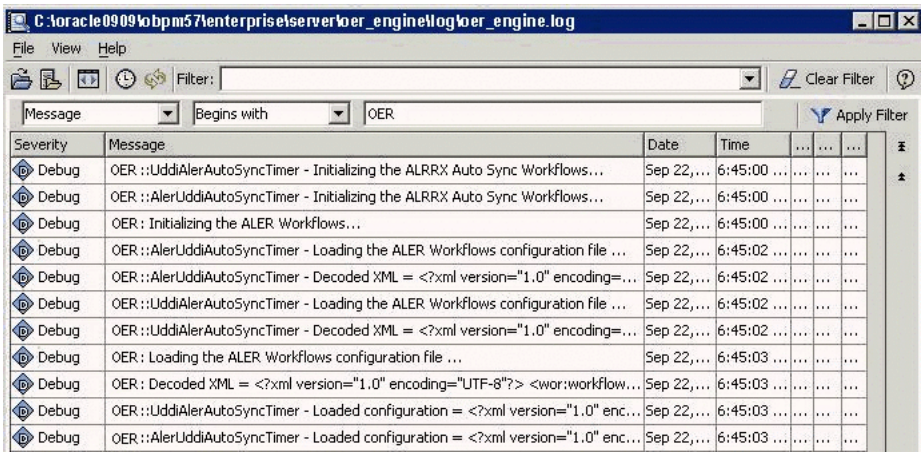
a. Within the Log viewer, select Message in the left-most list box.

b. Select Begins With in the next list box.

c. Type OER: in the text box.

d. Click the **Apply Filter** button.

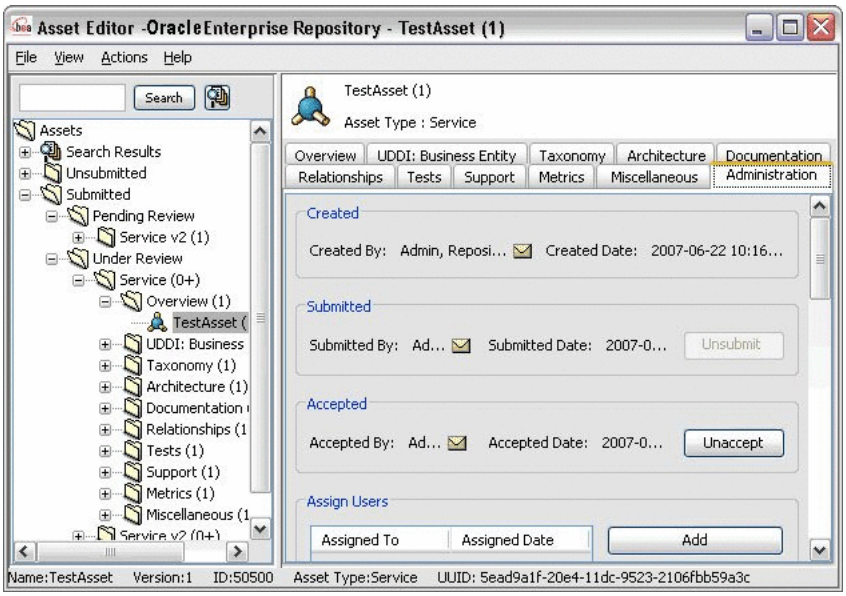
Figure A–50 Oracle Business Process Management Log Viewer



7. After the "OER: Done accepting the asset" message is displayed in the Log Viewer, switch back to the Asset Editor, and then refresh the Administration tab using the View -> Refresh Tree command.

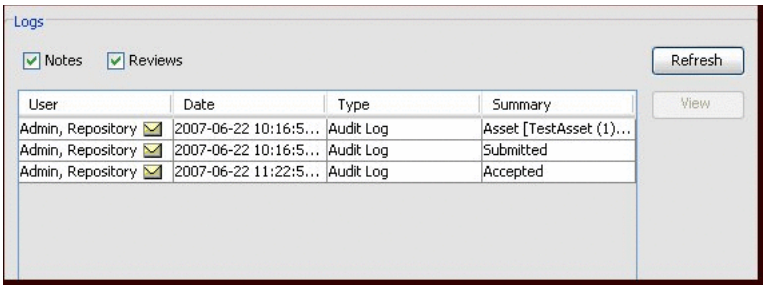
8. Verify that the "Accepted" section is updated with the latest data, as shown in Figure A–51.

Figure A–51 Oracle Enterprise Repository Asset Editor - Administration Tab



9. Also verify that the Audit Log on the Administration tab is updated, as shown in Figure A–52.

Figure A–52 Oracle Enterprise Repository Asset Editor - Audit Log



A.12 "Community Flow" Use Case

The Community flow provides a way to automate the asset acceptance, assignment, and registration process by allowing the configuration of automated assignment rules and also introduces the notion of federated registrars among different authorities. Rather than spamming many registrars across all communities (through the system registrar notification), you can limit the system registrar to one or a few individuals, and let the Automatic Acceptance flow accept assets on behalf of a registrar-of-record for the community. The Community flow feature can distribute asset submissions to those with the authority to approve them for the community.

For example, you can add two communities and configure two different registrars responsible for each community. Then, depending on the producing projects or asset types, certain assets can belong to a community. The Community flow automatically accepts such assets in the same way it would be manually accepted by a registrar.

B

Other Configurations

This appendix describes customization and configuration options not covered previously.

This appendix contains the following topics:

- Section B.1, "Customize the Oracle Enterprise Repository Help System"
- Section B.2, "Creating and Configuring an Upload Submission Directory"
- Section B.3, "Creating an Asset Audit Log"
- Section B.4, "Plug-ins and Adapters"
- Section B.5, "Oracle Enterprise Repository Default Artifact Store Integration"
- Section B.6, "Maintaining XSLT Templates for use with the Export-to-PDF Feature"
- Section B.7, "Support for UTF-8/Unicode Characters"
- Section B.8, "Asset Type Inventory"
- Section B.9, "Data Rules"
- Section B.10, "Harvester Properties"

B.1 Customize the Oracle Enterprise Repository Help System

In standard Oracle Enterprise Repository installations, Help links point to Oracle-hosted Help files. In some situations, however, it may be more appropriate to host the Help files locally.

This section contains the following topics:

- Section B.1.1, "Download the Help File"
- Section B.1.2, "Configure the Custom Help URL Property"

B.1.1 Download the Help File

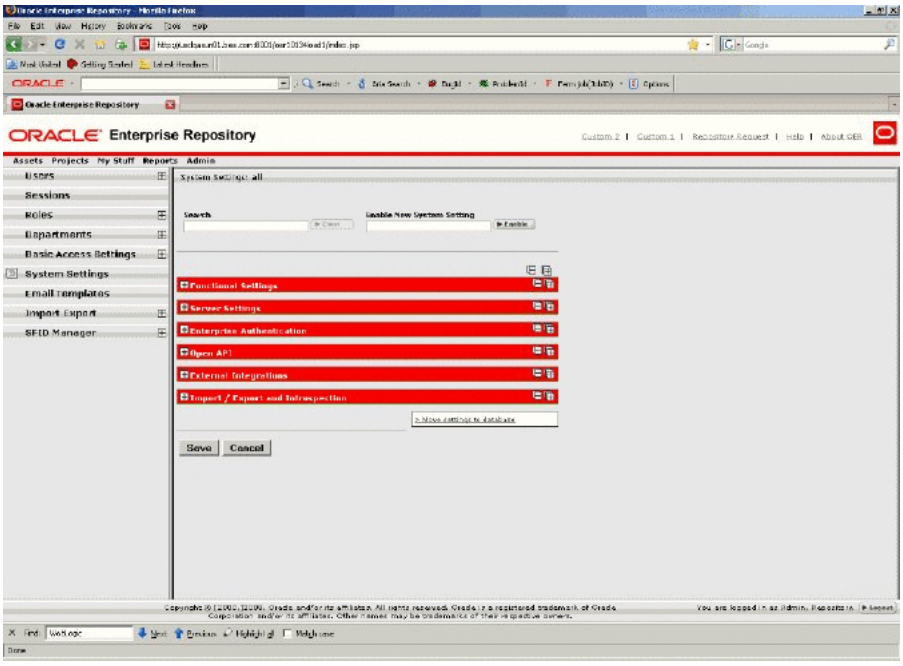
1. Download the compressed Oracle Enterprise Repository Help file. (Contact Oracle support for the correct version for your system.)
2. Extract the contents of the compressed Help file to the desired URL-accessible directory on the appserver/web server.

B.1.2 Configure the Custom Help URL Property

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

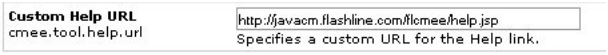
1. In the left panel, click **System Settings**. The System Settings section is displayed, as shown in Figure B–1.

Figure B–1 Oracle Enterprise Repository System Settings Section



2. Type `cmee.tool.help.url` into the System Settings Search text box. The Custom Help URL property is displayed, as shown in Figure B–2.

Figure B–2 Custom Help URL Property



3. Enter the appropriate URL for the Help files in the Custom Help URL property text box. The proper end-point to the URL should be wwhelp.htm.

   The primary Help URL must reflect the newly created file location. For example:

   http://host.domain.name:port/aler-web/help/wwhelp.htm

4. Click **Save**.
5. Refresh the browser.
6. Click **Help** in the Oracle Enterprise Repository menu bar to verify that the link points to the appropriate URL.

B.2 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 B.2.1, "Asset Upload System Settings"
- Section B.2.2, "Creating and Configuring an Artifact Store"
- Section B.2.3, "Selecting a New Artifact Store"

B.2.1 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 B–3 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: `<ORACLE_HOME>/user_projects/domains/<oerdomain>/autodeploy/<oer-app-name>/oer-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/Sharity.
- 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.oer.com`
  - Path: `upload/`
  - Username: `user_oer`
  - Password: `flash`
  - URL: `file:////unc.oer.com/upload/`
- HTTP
  - Hostname: `http.oer.com`
  - Path: `oer-web/upload`
  - Username: `user_oer`
  - Password: `flash`
  - URL: `http://user_oer:flash@http.oer.com/oer-web/upload/`
- FTP
  - Hostname: `meta.oer.com`
  - Path: `upload`
  - Username: `user_oer`

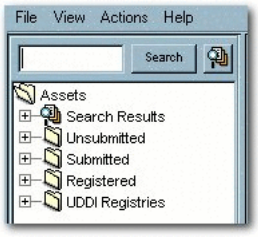
  - Password: `flash`
  - URL: `ftp://user_oer:flash@meta.oer.com/upload/`
- ClearCase
  - Hostname: `clearcase.oer.com`
  - Path: `ccaseweb/bin/ccweb`
  - Username: `user_oer`
  - Password: `flash`
  - URL: `http://clearcase.oer.com/ccaseweb/bin/ccweb/?dir=/&elem=&cmd=view&user=user_oer&password=flash`
- PVCS
  - Hostname: `pvcs.oer.com`
  - Path: `dav`
  - Username: `user_oer`
  - Password: `flash`
  - URL: `http://user_oer:flash@pvcs.oer.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`

B.2.2 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.

1. Click **Edit/Manage Assets**. The Asset Editor is displayed, as shown in Figure B–4.

Figure B–4 Asset Editor

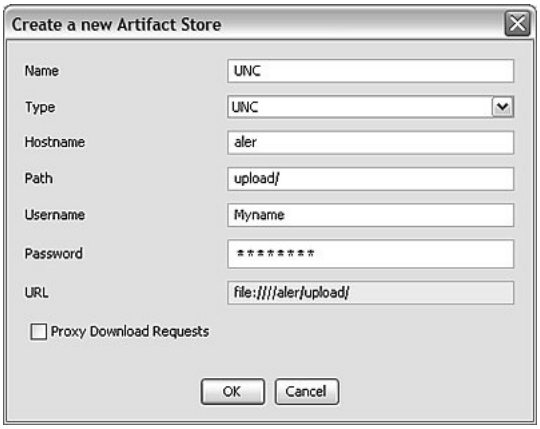


2. Select **Configure Artifact Stores** in the Actions menu. The Configure Artifact Stores dialog is displayed, as shown in Figure B–5.

Figure B–5 Configure Artifact Stores Dialog

3. Click **Add**. The Create a New Artifact Store dialog is displayed, as shown in Figure B–6.

Figure B–6 Create a New Artifact Store Dialog



4. Enter a unique name for the artifact store in the Name text box.
5. Select the type of artifact store from the Type list (this example uses UNC).
6. Enter the host name of the application server in the Hostname box.
7. Enter the rest of the path in the Path text box.

   A file link, appended with the host name and path, appears in the URL text box. This link can be cut/pasted into a file explorer/browser window in order to view the file.
8. If necessary, enter the appropriate information in the Username and Password text boxes.
9. Click **OK**.
10. Enable the artifact store using Submission Upload Artifact Store system setting, as described in Section B.2.3, "Selecting a New Artifact Store".

B.2.3 Selecting a New Artifact Store

After configuring an artifact store as described in Creating and 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, as shown in Figure B–3.
4. Use the Submission Upload Artifact Store list to select the newly created artifact store.
5. Click **Save**.

B.3 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**.

B.4 Plug-ins and Adapters

The following plug-ins and adapters are available for Oracle Enterprise Repository. Installation and configuration guides are available for each. Contact Oracle support for more information.

- LDAP Adapter
- SSO Adapter
- Repository Integration with Eclipse using Oracle Enterprise Platform for Eclipse (OEPE)
- Visual Studio .NET Plug-in
- WSAD Adapter
- PVCS Integration
- Connector for IBM Rational ClearQuest
- Connector for IBM Rational ClearCase
- Connector for Serena ChangeMan Dimensions

- Producing Projects
- Assets-In-Progress
- Apache Ant Integration
- Harvest-HTTP Repository Integration

B.5 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:

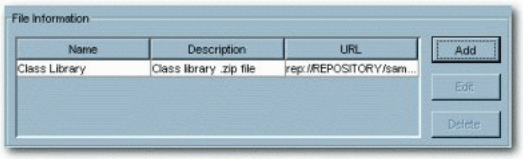
```
http://server.host.com:port/additionalPathStructure/fileName
```

Adding Files to an Asset

This procedure is performed in the Oracle Enterprise Repository Asset Editor.

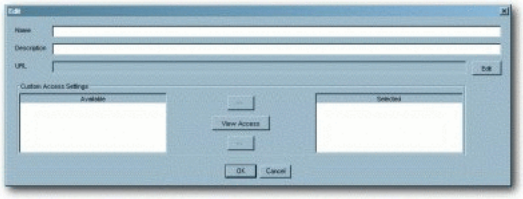
1. Use Search or other means to locate the asset to which a file is added.
2. Locate the File Information element on the Overview tab, as shown in Figure B–7.

Figure B–7 File Information Section



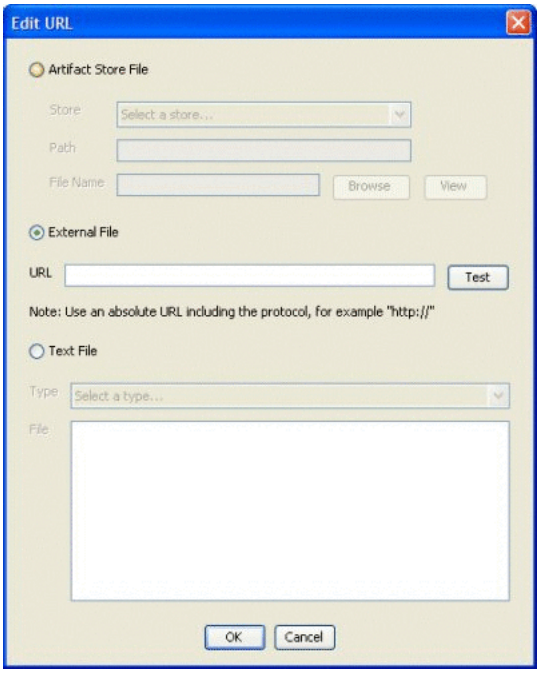
3. Click the **Add** button. The Edit dialog is displayed, as shown in Figure B–8

Figure B–8 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 B–9.

Figure B–9 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.

B.6 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.

Note: The proper directory location of the application deployment varies according to the application server in use. Please consult your systems administrator for the proper path information to supply.

B.6.1 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 <<HOW DID IT GET THERE?>> 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.

B.7 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.xsl
- Business Process.xsl
- Communication Adapter.xsl
- Component.xsl
- Environment.xsl
- Framework.xsl

- Pattern.xsl
- Process.xsl
- Project Profile.xsl
- Service.xsl
- XML Schema.xsl

Sample XSLTs which work only in Single Byte Character Set

- Artifact_ DTD.xsl
- Artifact_ MFL.xsl
- Artifact_ WS-Policy.xsl
- Artifact_ WSDL.xsl
- Artifact_ XQuery.xsl
- Artifact_ XSD.xsl
- Artifact_ XSLT.xsl
- Assembly Model.xsl
- Binding.xsl
- Binding_ File Source.xsl
- Binding_ JMS.xsl
- Binding_ Relational Database.xsl
- Binding_ SB-Custom.xsl
- Binding_ SB-DSP.xsl
- Binding_ SB-EJB.xsl
- Binding_ SB-Email.xsl
- Binding_ SB-File.xsl
- Binding_ SB-FTP.xsl
- Binding_ SB-HTTP.xsl
- Binding_ SB-JMS.xsl
- Binding_ SB-Local.xsl
- Binding_ SB-MQNative.xsl
- Binding_ SB-Oracle.xsl
- Binding_ SB-PeopleSoft.xsl
- Binding_ SB-SAP.xsl
- Binding_ SB-SB.xsl
- Binding_ SB-SFTP.xsl
- Binding_ SB-Siebel.xsl
- Binding_ SB-Tuxedo.xsl
- Binding_ SB-WLI.xsl
- Binding_ Web Service.xsl

- Component_ Data Service.xsl
- Component_ DSP Web Service Mapper.xsl
- Component_ Java.xsl
- Component_ SB Proxy Service.xsl
- Component_ SCA Composite.xsl
- EA - Sample Asset Type.xsl
- EA - Waiver.xsl
- Endpoint.xsl
- Interface.xsl
- Interface_ Data Service.xsl
- Interface_ Java.xsl
- Interface_ Messaging.xsl
- Interface_ Web Service.xsl
- Procedure.xsl
- SCA Composite.xsl
- SCA Reference.xsl
- SCA Service.xsl
- SOA - Business Entity.xsl
- Training.xsl

B.8 Asset Type Inventory

This section describes the asset type inventory for base data pack and the harvester solution pack.

B.8.1 Base Data Pack

The Base Data Pack is the set Oracle Enterprise Repository data, which is installed with a new installation of Oracle Enterprise Repository. This Data Pack is initially required by every new instance of Oracle Enterprise Repository.

Name	UUID	Description
Application	af788182-4c10-11dc-ac31-7de68f5ca4a1	
Artifact: DTD	af693f35-4c10-11dc-ac31-7de68f5ca4a1	
Artifact: WSDL	dcb567bc-2428-11dc-8a61-b94a9aa7eb2f	Represents the entity that contains the WSDL artifact.
Artifact: WS-Policy	dccb87e1-2428-11dc-8a61-b94a9aa7eb2f	Represents the entity that contains the WS-Policy artifact.
Artifact: XSD	dcca7670-2428-11dc-8a61-b94a9aa7eb2f	
Binding	af6ff5fb-4c10-11dc-ac31-7de68f5ca4a1	

Name	UUID	Description
Business Process	af7218dd-4c10-11dc-ac31-7de68f5ca4a1	
Communication Adapter	af6dd319-4c10-11dc-ac31-7de68f5ca4a1	
Component	af7c5215-4c10-11dc-ac31-7de68f5ca4a1	
Endpoint	af7b40a4-4c10-11dc-ac31-7de68f5ca4a1	
Environment	af763790-4c10-11dc-ac31-7de68f5ca4a1	
Framework	af6806b4-4c10-11dc-ac31-7de68f5ca4a1	
Interface	af74d7ff-4c10-11dc-ac31-7de68f5ca4a1	
Pattern	af774901-4c10-11dc-ac31-7de68f5ca4a1	
Procedure	af71076c-4c10-11dc-ac31-7de68f5ca4a1	
Service	af6ee48a-4c10-11dc-ac31-7de68f5ca4a1	Represents the concrete Service that is an implementation of an Interface
SOA - Business Entity	af73c68e-4c10-11dc-ac31-7de68f5ca4a1	
Training	af7a0823-4c10-11dc-ac31-7de68f5ca4a1	

B.8.2 Harvester Solution Pack

The Harvester is available in **11.1.1.x.x-OER-Harvester-Solution-Pack.zip**, which is bundled with the Oracle Enterprise Repository 11*g* installation.

Name	UUID	Description
Adapter	a22d8c30-d767-11dd-bee0-79d657a0a2b0	Represents the concrete Adapter that provides access to an Enterprise Application.
Adapter Configuration	a30bde41-d767-11dd-bee0-79d657a0a2b0	Represents the configuration information which connects a Service to an Adapter.
Artifact : XQuery	908940d3-23e1-11dc-a7fd-f9d708c0d3f4	Represents the entity that contains the XQuery artifact.
Artifact : XSD	dcca7670-2428-11dc-8a61-b94a9aa7eb2f	Represents the entity that contains the XSD artifact.
Artifact : XSLT	907347be-23e1-11dc-a7fd-f9d708c0d3f4	Represents the entity that contains the XSLT artifact.
Artifact: Biz	ed1ed534-39b8-11de-bee0-79d657a0a2b0	Represents the entity that contains the BIZ artifact.
Artifact: BPEL	4000545f-73ae-11dd-833d-572100858305	Represents the entity that contains the BPEL artifact.
Artifact: ComponentType	b68aed1b-4705-11de-bee0-79d657a0a2b0	
Artifact: Composite	01299694-3680-11de-bee0-79d657a0a2b0	A composite.xml file which defines a composite

Name	UUID	Description
Artifact: Decs	1225bb95-3680-11de-bee0-79d657a0a2b0	
Artifact: EDL	b20c5569-3680-11de-bee0-79d657a0a2b0	An .edl file which defines an event
Artifact: FaultPolicies	c531c4c9-88fd-11de-bc41-3f5f390f1c4f	
Artifact: Flow	6cb09bd6-39b9-11de-bee0-79d657a0a2b0	
Artifact: Jar	87e299ac-9988-11de-bc41-3f5f390f1c4f	
Artifact: Java	95e94c6d-9988-11de-bc41-3f5f390f1c4f	
Artifact: JCA	d3a652ca-3680-11de-bee0-79d657a0a2b0	
Artifact: MFL	90767c11-23e1-11dc-a7fd-f9d708c0d3f4	Represents the entity that contains the MFL artifact.
Artifact: MPlan	955e06e3-367e-11de-bee0-79d657a0a2b0	An .mplan file which defines a Mediator
Artifact: Proxy	132bbd15-39b9-11de-bee0-79d657a0a2b0	
Artifact: Rules	4eda8116-3680-11de-bee0-79d657a0a2b0	A .rules file which defines a rules dictionary
Artifact: SpringBeans	b443f7a8-88fd-11de-bc41-3f5f390f1c4f	
Artifact: Task	898dd147-3680-11de-bee0-79d657a0a2b0	A .task file which defines a human task (workflow)
Artifact: Toplink	6c09ffcb-9988-11de-bc41-3f5f390f1c4f	
Artifact: WSDL	dcb567bc-2428-11dc-8a61-b94a9aa7eb2f	Represents the entity that contains the WSDL artifact.
Artifact: WS-Policy	dccb87e1-2428-11dc-8a61-b94a9aa7eb2f	Represents the entity that contains the WS-Policy artifact.
Business Process: BPEL	d9f3ac95-73ab-11dd-833d-572100858305	Represents the high level Business Process.
Business Rule	80f3cb02-3683-11de-bee0-79d657a0a2b0	A business rule (decision service) definition, from Oracle Rules
Composite	feb0203b-3681-11de-bee0-79d657a0a2b0	An SCA composite
Composite Deployment	daad43be-3682-11de-bee0-79d657a0a2b0	A deployed SCA composite
Composite Deployment Reference	ee9eb3ef-3682-11de-bee0-79d657a0a2b0	A reference required by a deployed composite
Composite Deployment Service	14b97e80-3683-11de-bee0-79d657a0a2b0	A service offered by a deployed composite
Composite Reference	cad63dcd-3682-11de-bee0-79d657a0a2b0	A reference required by a composite
Composite Service	bdb3d08c-3682-11de-bee0-79d657a0a2b0	A service offered by a composite
Deployment: BPEL	11683a9e-73ae-11dd-833d-572100858305	Represents the deployed instance of a Business Process

Name	UUID	Description
Endpoint	af7b40a4-4c10-11dc-ac31-7de68f5ca4a1	
Enterprise Application	a381aa82-d767-11dd-bee0-79d657a0a2b0	Represents the high-level Enterprise Application.
Enterprise Application Deployment	a4007773-d767-11dd-bee0-79d657a0a2b0	Represents the deployment of an Enterprise Application.
Fault Policy	d240f82a-88fd-11de-bc41-3f5f390f1c4f	
Human Task	5750eb21-3683-11de-bee0-79d657a0a2b0	A human task definition, from Oracle Workflow
Interface	af74d7ff-4c10-11dc-ac31-7de68f5ca4a1	
Service	af6ee48a-4c10-11dc-ac31-7de68f5ca4a1	Represents the concrete Service that is an implementation of an Interface
Spring Context	a2698097-88fd-11de-bc41-3f5f390f1c4f	

B.9 Data Rules

This section contains the following topics:

- Section B.9.1, "Flags on Asset Type Elements"
- Section B.9.2, "Grandfathered Data"
- Section B.9.3, "Element Storage"
- Section B.9.4, "Elements Required to Exist on Asset Types"
- Section B.9.5, "Elements Tied to Functionality"

B.9.1 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.

B.9.2 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".

B.9.3 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.

B.9.4 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.

B.9.5 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 `cmee.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" (`cmee.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 B.10, "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						

B.9.6 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.

B.10 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

Property Name	Applies to Asset Type(s)	Description
(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.
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,Composite 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,Composite 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.

Property Name	Applies to Asset Type(s)	Description
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
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

Source Product	Source interface / Component / File Type	Oracle Enterprise Repository Service Type
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
SOA Suite	interface.wsdl	Java Service
raw WSDL	(all)	Web Service

Index

A

Acceptance, A-71
Accepting a Submission, A-71
Access, 4-5
Access Administrator, 4-14
Access Control, 17-18
Access Definitions, 1-25, 4-4
Access Options, 4-15
Access Setting and the Asset Editor, 1-24
Access Settings, 1-21, 4-4
Access Settings and the Asset Editor, 4-2
Access System Settings, 17-1
Accessing Oracle Enterprise Repository Diagnostics Page, 5-1
Activating the Policy Management Feature, 14-2
adapterAsset, 6-22
Adapters and Applications, 6-21
Adding a Relationship to a Project, 1-13
Adding an SFID to an Asset, 13-5
Adding Files to an Asset, B-8
Adding Metrics, 7-2
Adding the Applied Asset Policies Element to a Type, 14-6
Adding the Policy Assertion Element to the Policy Type, 14-4
Adding the SFID Element to a Type, 13-2
Administering Oracle Business Process Management Web Applications, A-22
Administrating Oracle Business Process Management Processes, A-22
Administration Tab, A-77
Advanced Configuration, 6-21
Advanced Options, 3-14
Advanced Registration Flow Components, A-81
Advanced Role-based Access Control, 4-1
Advanced Submitter, 4-14
Affinity Settings, 17-25
applicationName, 6-23
Applications via Oracle BSE (WSDL/SOAP), 6-32
Apply a Compliance Template to a Project, 11-6
Applying a Policy to an Asset, 14-7
Approval, A-71
Architecture Blueprints, 11-1
Artifacts/Products Version Matrix, 6-3
Asset Expiration Warning Notification, A-49
Asset Type Inventory, B-14
Asset Types Created by the Harvester, 6-24
Asset Upload System Settings, B-3
Assets, 4-4, A-1
Assets in Progress, 12-1, 17-15
Assets Pane - Consumption Method, 6-42
Audit Log, Reviews, and Notes, A-79
Automated Asset Registration Using Automated Workflows, A-71
Automated Workflow Components, A-14
Automated Workflows, A-3, A-80
Automated Workflows Engine, A-24
Automatic Asset Acceptance Flowchart, A-32
Automatic Asset Registration Flows, A-31

B

Base Data Pack, 2-7, B-14
Basic Concepts, 4-3
Basic Configuration, 1-1
see Departments
see Overview
Best Practices, 6-44
BI Publisher, 16-1
BPEL, 6-25
build.properties File Property Values, A-8
Business Entities, 10-23

C

Calculating the SFID Programmatically, 13-11
Categorizations, 10-25
Checking the Oracle Registry Repository Exchange Utility Log File, 10-30
Classpath URLs, 6-21
Cleaning Up Stored Events, A-61
Cloning a User, 1-6
Closing a Project, 1-12
Command Line Options for the Harvester, 6-7
Completing the Tab Approval Process, A-78
Compliance Template, 11-1
Compliance Templates, 1-7
Configure and Run the Oracle Business Process Management Process Engine, A-81
Configure Oracle Enterprise Repository for Container Managed Authentication, 3-19

Configure Oracle Enterprise Repository For Use With Single Sign-On Authentication, 3-8
Configure the Custom Help URL Property, B-1
Configure the Oracle Enterprise Repository Event Manager, A-80
Configure the setenv File, A-10
Configure Using the Asset Type Name, 10-4
Configuring a JMS Provider In WebSphere 6.1.0.5, A-57
Configuring a Standalone Process Engine for Failover, A-26
Configuring Access Settings for Existing Roles, 4-13
Configuring Advanced Role-based Access Control, 4-1
Configuring and Using Automated Harvesting, 6-1
Configuring Assets in Progress, 12-1
Configuring Automated Usage Detection, 13-1
Configuring Automated Workflows, A-27
Configuring BI Publisher Enterprise Server, 16-1
Configuring Community Flows, A-31
Configuring Connectivity Properties for External JMS Servers, A-53
Configuring Email Notifications and Distribution Lists, 8-1
Configuring Embedded JMS Servers for Clustering, A-56
Configuring External JMS Jar Files, A-55
Configuring External JMS Servers for Clustering, A-57
Configuring JMS Durable Subscribers for Web Service Endpoints, A-56
Configuring JMS Message Header Properties, A-54
Configuring JMS Servers, A-52
Configuring JMS Servers in an Oracle Enterprise Repository Cluster, A-56
Configuring Logging for the Harvester, 6-21
Configuring Logging of Event Manager Events, A-21
Configuring Metadata Change Flows, A-40
Configuring Metrics, 7-1
Configuring Optional Event Manager Settings, A-17
Configuring Oracle Enterprise Repository Categorizations in the UDDI Mappings File, 10-11
Configuring Oracle Enterprise Repository to use External Authentication Tooling, 3-1
Configuring Oracle Enterprise Repository Workflow, 9-1, A-1
Configuring Policy Management, 14-1, 15-1
Configuring Prescriptive Reuse, 11-1
Configuring the Automated Workflows to Process a Submission Event, A-82
Configuring the BI Publisher Server - Cache Expiration, 16-4
Configuring the Embedded ActiveMQ JMS Server to Use a Database, A-55
Configuring the Event Manager, A-15
Configuring the Event Manager's System Settings, A-17
Configuring the Harvester, 6-3
Configuring the Harvester for the Command Line, 6-5
Configuring the Oracle Enterprise Repository Event Manager, A-14
Configuring the Oracle Enterprise Repository JDBC Data Source, 16-2
Configuring the Oracle Enterprise Repository System Settings to Enable Reporting, 16-4
Configuring the Oracle Registry Repository Exchange Utility, 10-3
Configuring the Oracle Registry Repository Exchange Utility Configuration File, 10-5
Configuring the Subscription Manager, A-18
Configuring the tModelKey UDDI Setting, 10-12
Configuring Web Service Endpoints, A-19
Configuring Workflow, A-63
Consumed Assets, 1-7
Container Managed Setup, 3-18
Create a Submission Upload Directory, B-2
Create New Users/Allow Unapproved Users, 3-16
Create the Asset CAS, 4-8
Create the File CAS, 4-13
Creating a Compliance Template, 11-5
Creating a Compliance Template Type, 11-2
Creating a Distribution List, 8-2
Creating a New Department, 1-2
Creating a New Role, 1-18
Creating a New User, 1-3
Creating a Policy, 14-5
Creating a Policy Type, 14-3, 15-2
Creating a Project, 1-8
Creating an Asset Audit Log, B-7
Creating and Configuring an Artifact Store, B-5
Creating and Configuring an Upload Submission Directory, B-2
Creating and Customizing a Workflow Configuration File, A-28
Creating/Assigning Default Roles for New Users, 3-15
Custom Links, 17-20
Customize the Oracle Enterprise Repository Help System, B-1
Customizing Email Notification Templates, A-49
Customizing Oracle Enterprise Repository, 2-1

D

Data Rules, B-17
Database Runtime Properties, A-25
dbqueries, 7-5
Define Security Model, 16-2
Defining the Oracle Enterprise Repository Connection and Registrar, A-29
Delete Assets After Expiration, A-49
Deleting a Role, 1-20
Deleting a Single Session, 1-27
Deleting Multiple Session, 1-27
Departments, 1-2
Deploy Oracle Enterprise Repository Pre-configured BI Publisher Content, 16-1

Description of Sample Directory Crawler, 13-7
Description of the Asset Type Catalog, B-22
Detecting Duplicate Artifacts, 6-41
Developing a Notifier Plug-in, A-68
Developing a Web Service Endpoint, A-65
Developing an Endpoint with an Incompatible Contract, A-69
Diagnostics Page, 5-1
Disabling Assets in Progress, 12-4
Documentation Tab, A-75
Download the Help File, B-1
Downloading Harvested Artifacts, 6-41

E

Editing a Project, 1-11
Editing a Role, 1-20
Editing Department Information, 1-2
Editing Permissions, 1-23
Editing User Information, 1-6
Element Storage, B-18
Elements Required to Exist on Asset Types, B-19
Elements Tied to Functionality, B-19
Email Notification Templates, A-5
Email Notification Use Cases, A-50
Email Templates, 8-4
Enable Debug Logging, 3-18
Enable LDAP Integration System Properties, 3-2
Enable Single Sign-On Integration System Properties, 3-9
Enable System Settings Properties, 17-3
Enable Unapproved/New User Login, 3-16
Enabling and Configuring an External JMS Server, A-53
Enabling Assets in Progress, 12-3
Enabling Custom Access Settings, 4-2
Enabling Harvesting of Unrecognized SFIDs, 13-6
Enabling JMS Clustering Mode, A-56
Enabling SFID, 13-2
Enabling the Event Manager, A-17
Enabling the SFID Manager, 13-8
Enabling Viewing of Assets in Progress Through an IDE, 12-4
Encrypt Password Tool, 5-4
Encrypted Password Used in Oracle Enterprise Repository Deployment, 5-2
Encrypting Config File Passwords, A-5
Encrypting the Registrar User Password, A-29
Encrypting Your Passwords, A-64
Endpoint, 10-24
Enterprise Authentication Settings, 17-32
eTrust Single Sign-On, 3-8
Event Log Messages, A-26
Event Management Tools, A-4
Event Monitor, A-3
Event Monitoring Console, A-17
Example "Community Flow" Use Case, A-85
Example Use Cases, 10-2
Exchange Utility Configuration File, 5-3
Exchange Utility Tool
see Known Issues
Extending the Event Manager for Web Service Endpoints, A-65
External Integration Settings, 17-48

F

FileSet, 6-11
Flags on Asset Type Elements, B-17
Functional Settings, 17-4

G

General User Interface, 17-4
Generating a New Config File, A-5
Generating Encrypted Passwords, 5-2
Getting Started with Harvester, 6-1
see Best Practices
see Prerequisites
*see*Known Issues
Grandfathered Data, B-17
Grant/Deny Access to Specific Assets, 4-7
Grant/Deny Access to Specific Download Files Within an Asset, 4-12
Granting and Denying Permissions, 4-7
Granting Permissions to a Role, 1-21

H

Harvester, 6-1
Harvester Configuration File, 5-3
Harvester Functionality, 6-2
Harvester Properties Matrix, B-22
Harvester Solution Pack, 2-7, B-15
Harvesting from Oracle SOA Suite Server, 6-17
Harvesting of Artifacts, 6-23
Harvesting Oracle Business Process Analysis Projects, 6-36
Harvesting Oracle SOA Suite, 6-26
Harvesting Web Services from Oracle Service Bus, 6-19
Harvesting Web Services from WebLogic Server, 6-19
How the Exchanged Metadata Is Synchronized, 10-23

I

Inactivate After Expiration, A-49
Install OBPM, A-6
Install Oracle Enterprise Repository Solution Packs, 2-7
Install the Oracle Registry Repository Exchange Utility, 10-4
Installing and Configuring the Oracle Registry Repository Exchange Utility, 10-4
Installing Oracle Enterprise Repository Workflow into Oracle BPM 10.3, A-5
Installing Oracle Service Registry, 10-4
Invoking Harvester from WLST, 6-15
Invoking the Harvester Using the Repository.Submit

Ant Task, 6-10
Invoking the Oracle Registry Repository Exchange Utility Using Workflows, 10-21

J

JCA Adapters, 6-31
jcaCSAAdapterName, 6-22
jcaSpecBinding, 6-22
JMS Server, A-3

K

Known Issues, 6-47, 10-30

L

LDAP Property Examples, 3-7
LDAP/Active Directory, 3-1
see Overview
see Security Considerations
Locating a Project, 1-12
Log Viewer, A-4
Log Viewer With OER Filter, A-27

M

Maintaining XSLT Templates for use with the Export-to-PDF Feature, B-10
Managing E-mail Distribution Lists, 8-2
Memory and Execution Thread Properties, A-25
Metadata Change Flows, A-39
Metric Elements
see Overview
Miscellaneous, 17-23
Miscellaneous JMS Properties, A-55
Modify LDAP/Active Directory Properties, 3-3
Modify the Web Application's Web.xml File, 3-20
Monitoring and Managing Events, A-60
Monitoring Events, A-60
Move Settings to Database, 17-3
Multi-tier Automatic Assignment Flowchart, A-37
Multi-tier Automatic Assignment Flows, A-36

N

namespace, 6-22
New User Notification, 3-16
Notes on Department Synchronization, 3-17
Notifications, 17-17
Notifier Plug-in, A-68

O

Object Model, 13-7
Obtaining the Harvester, 6-4
Open API, 17-48
Oracle Business Process Management, A-3
Oracle Business Process Management Log Viewer, A-26, A-84
Oracle Enterprise Repository Automated Workflows, A-2
Oracle Enterprise Repository Default Artifact Store Integration, B-8
Oracle Enterprise Repository Event Manager, A-3
Oracle Metadata Service URLs, 6-20
Oracle Registry Repository Exchange Utility, 10-1
Oracle Service Bus (OSB) Project, 6-34
Other Configurations, B-1
Other Passwords, 5-2
Overview, 1-1, 3-1, 7-1
Overview of Automated Workflows, A-28
Overview of JMS for the Event Manager, A-52
Overview Tab, A-72

P

parameter.properties, 6-4
Parameters for the repository.submit Ant Task, 6-10
Password Encryption, 5-1
Performing Optional Harvester Configuration, 6-20
Plug-ins and Adapters, B-7
Policies, 4-5, 14-1
Policy Management, 17-16
Prerequisites, 6-1
prescriptive reuse, 11-1
Print Asset Detail via PDF, 17-12
Process Overview, 4-6
Process to approve a tab, A-79
Produced Assets, 1-8
Project Administrator, 4-15
Project Profiles, 11-1
Projects, 1-6, 4-5
Projects and Navigator, 17-13
Proxy Service or Business Service, 6-33

Q

Query by Categorizations, 10-7
Query by Filters, 10-7
Query by Name, 10-6
Query by Registration Status, 10-7

R

Reassigning Users/Usage, 1-11
Rebranding Oracle Enterprise Repository, 2-2
Refreshing an Existing Config File, A-5
Refreshing the Workflow Config File, A-63
Registering an Asset, A-72
Registrar, 4-14
Registrar Administrator, 4-15
Registration and Active Status, 10-25
Rejection, A-70
Relationship Diagram, 1-2
Relationships Tab, A-76
RemoteProject, 6-13, 6-14
Remove an SFID File, 13-6
Reports, 4-5, 17-22
Review, A-70
Role Synchronization, 3-17
Role-based Access Control Use Cases, 4-25

Role/Department-based Homepage Display, 2-5
Roles, 1-16, 4-3
Running a Project Compliance Report, 11-9
Running Queries Against the Database, 7-3
Running the Harvester from Ant, 6-14
Running the Oracle Enterprise Repository Policy Reports, 14-12
Running the Oracle Registry Repository Exchange Utility, 10-20
Runtime Harvesting Details, 6-17

S

Sample Event Cleanup, A-62
Sample WSDL File, A-66
Search, 17-9
Search for System Settings, 17-2
Searching for Oracle Service Registry, 10-29
Searching Harvested Assets, 6-41
Security Considerations, 1-24, 3-5, 4-2
Security Encrypt Password tool, A-29
Selecting a New Artifact Store, B-7
Selecting a Web Service Operation, A-67
Selecting the Artifacts to Harvest for the Command Line, 6-6
Server Settings, 17-25
Service Keys, 10-24
Sessions, 1-26
see Deleting a Single Session
see Deleting Multiple Session
see Viewing Sessions
Set up a Shared Directory for a Submission Upload Directory, B-3
setenv File Property Values, A-10
Setting Repository Connection Information for the Command Line, 6-6
Setting the Destination Registries, 10-10
Setting the Expression to Filter Events, A-20
Setting the Registry Connection Information, 10-5
Setting the Registry Query, 10-10
Setting the Repository Connection Information, 10-5
Setting the Repository Query, 10-6
Setting the Source Registry, 10-11
Setting Up a Community for Multi-tier Tab Approval, A-38
Setting Up an Asset Type for Multi-tier Tab Approval, A-38
SFID Limitations, 13-11
SFID Manager, 13-8
SFID Tool Kit, 13-6
Software Components, A-3
Software File Identification (SFID), 13-1, 17-19
Specifying Parameters as Nested Elements, 6-11
Specifying Parameters for the repository.submit Ant Task, 6-10
Starting the Oracle Business Process Management Admin Center, A-22
Starting the Oracle Business Process Management Process Engine, A-23
Submission, A-70
Subscription Manager, A-3
Support for UTF-8/Unicode Characters, B-12
Synchronizing Using Timer Based Workflows, 10-21
Synchronizing When Events are Triggered, 10-22
Syncing, 3-17
Syncing Departments, 3-16
Syncing Roles, 3-17
System Administration, 4-5
System Administrator, 4-15
System Settings in Oracle Enterprise Repository, 17-3
System Settings Overview, 17-1

T

Taxonomy Tab, A-73
Time-based Escalation Flowchart, A-46
Time-based Escalation Flows, A-45
Tracking Unaccepted Assets, A-46
Tracking Unapproved Assets, A-47
Tracking Unregistered Assets, A-47
Tracking Unsubmitted Assets, A-46
Transaction Handling in Harvester, 6-21
Triggering an Asset Event, A-16
Triggering an Asset Submission Event, A-83

U

Understanding the Oracle Registry Repository Exchange Utility's Property File, 10-15
Understanding the Registration Process, A-69
Unregister Assets After Expiration, A-49
Update an SFID File, 13-5
Updating a Distribution List, 8-4
URI, 6-11, 6-12
Use - Download, 17-19
User, 4-14
User Role Descriptions, 1-16
User Roles and Default Privileges, 4-14
User Search, 1-5
Users, 1-3, 1-8
Using an alerid for Tab Approvals, A-37
Using Assets Associated with Compliance Templates, 11-7
Using Consumption Method, 6-42
Using Email Notifications/ Email Distribution Lists, 8-1
Using Harvester within an Integrated Development Environment, 6-9
Using Metadata, 6-42
Using the Oracle Enterprise Repository Exchange Utility, 10-19
Using the SFID Manager, 13-9
Using the Third-Party Tasks, 6-15

V

Valid Metadata Entities, 10-2
Validate the Elements on the Asset Type, 7-3
Validating Policy Assertions, 14-9
Validation Expiration Flowchart, A-48

Validation Expiration Flows, A-47
Verifying User Permissions, 1-23
Viewing Assets in Progress, 12-2
Viewing Policy Status, 14-9
Viewing Project Details, 1-7
Viewing Sessions, 1-26
Viewing User Information, 1-4

W

Web Service Operations, A-66
Web-based Process Administrator, A-4
What is the Oracle Enterprise Repository Event Manager?, A-14
What is the Oracle Registry Repository Exchange Utility?, 10-1
When List Members Are Copied on E-mails, 8-4
Wiring Asset Events to Flows, A-29
Workflow Configuration file, A-29
Workflow Configuration Tools, A-5
WSDL, 6-25

X

XSD, 6-26
XSL, 6-26