Oracle® Enterprise Repository

Integration with Development Environments

10*g* Release 3 (10.3)

July 2009

ORACLE

Oracle Enterprise Repository Enterprise Manager Integration User Guide, 10*g* Release 3 (10.3)

CHAPTER 1

Introduction

This chapter contains information on the following topics:

Overview

The Oracle Enterprise Repository provides integration within development environments so developers can easily search for and use assets from the repository without leaving their developent environment. Assets and any associated artifacts are downloaded directly to the developer's workspace. Integration also provides a convenient way to submit or harvest assets from the development environment into the Oracle Enterprise Repository for use throughout the enterprise.

Repository Access within the developer's workspace also provides a view into Oracle Enterprise Repository that enables developers to, download artifacts and assets from the repository, query the repository, and view the contents of the repository.

The goal of this integration is to ensure that SOA Governance becomes part of the fabric of every day development.

Best Practices

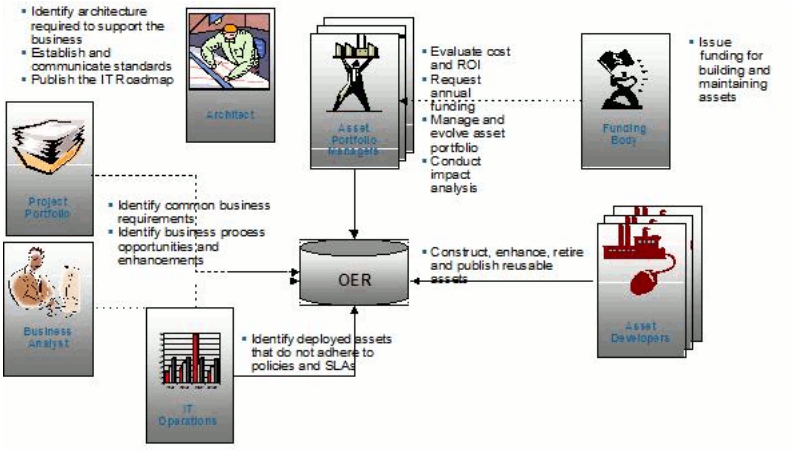
This section describes the following best practice processes:

Asset Production Process

Oracle Enterprise Repository is used to track new assets and asset enhancements from the time they are proposed through the retirement of the assets, which also includes the time when the assets are being developed and when the assets are completed. Through Oracle Enterprise Repository, assets are used to produce and consume projects, to provide traceability and support impact analysis, and to point out project-level impacts of changes to the asset release plans.

Initially, a business analyst provides a description of the functionality that needs to be produced. If determined that the proposed functionality does not already exist, then an architect provides the functional designs and non-functional requirements. The development team then produces, harvests, or enhances an asset that meets the functional and non-functional requirements.

Figure 1-1 The Asset Production Process



This means that the development team needs visibility into the assets that they are to produce such as the requirements, use cases and so on. Oracle Enterprise Repository provides that visibility by allowing developers to view an asset details of an asset directly from the development environment. The development team builds the specified asset and then harvests the asset into the Oracle Enterprise Repository, where it goes through a review and approval process. Figure 1-1 describes a sample asset production process.

Policies

The production and enhancement of assets can be governed through design-time policies. 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% .

Policy information in Oracle Enterprise Repository can be accessed through the development environment. Once the development team harvests an asset into Oracle Enterprise Repository, then the policies can be automatically validated through tooling or manually validated by subject matter experts.

Propose/Submit Assets

Assets may be proposed or submitted to the Oracle Enterprise Repository in multiple ways, depending on the role and situation, some of the methods are as follows:

- The general Oracle Enterprise Repository user community can submit asset requests or completed assets from the console.
- Business analysts and architects can create assets to be built using the Asset Editor.
- Development teams can submit or harvest assets from their development environments.
- Existing assets stored in files or directories can be harvested by Competency Centers or Portfolio Managers.
- QA or IT Operations can harvest assets from the build environment or from the run time environment.

Asset Consumption Process

There are two primary design-time consumption models within a standard Software Development Lifecycle (SDLC) process:

Model 1: Assets are identified early in the lifecycle (also known as Prescriptive Reuse)

- Business analysts and/or project architects identify assets that fulfill the functional and non-functional requirements of the project.
- Development teams then receive a “kit” of relevant assets.

Model 2: Development-driven discovery

- Developers identify assets that might fulfill the functional and non-functional requirements of the project.

Model 1 is the most-preferred design-time consumption models because it results in a higher level of reuse, but requires the organization to have a mature SDLC process and well-defined roles.

Prescriptive Reuse

Within the Oracle Enterprise Repository Web console, business analysts and/or project architects identify assets that fulfill the functional and non-functional requirements of the project. They package these assets into a “kit” called a Compliance Template. A Compliance Template communicates asset requirements or asset solution sets to internal or outsourced project teams. Two types of Compliance Templates included in Oracle Enterprise Repository are:

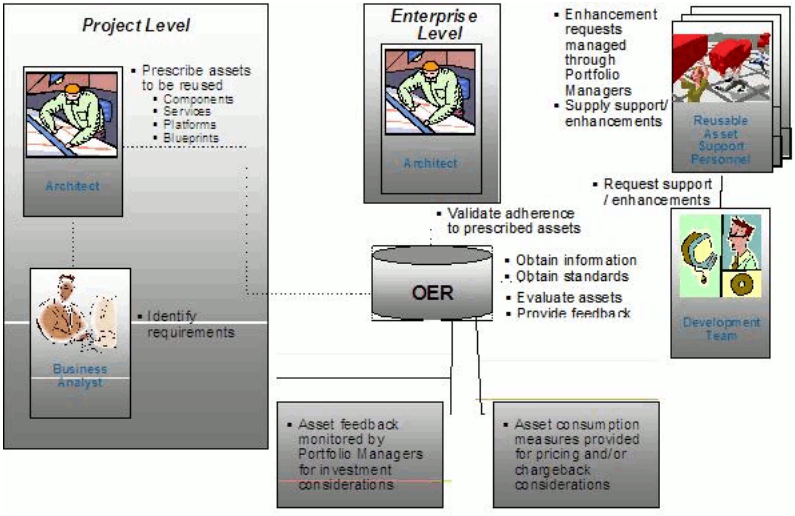
- Project Profiles
- Architecture Blueprints

Project profiles are usually created for individual projects, whereas architecture blueprints are reusable solution sets that can be leveraged by multiple projects. Some of the common use cases include:

- Project planners generate a project profile for each project in the portfolio, identifying the reusable assets that factored into the project’s planning and estimating assumptions.
- Business analysts generate a project profile to identify assets that fulfill a project's business requirements.
- Project architects generate a project profile to identify assets that fulfill a project's technical requirements.
- Enterprise architects generate an architecture blueprint that specifies the standard frameworks and assets that are to be used to fulfill project-level security requirements.
- Those responsible for Service-Oriented Architecture(s) (SOA) generate an architecture blueprint to identify the services that orchestrate a particular business function.
- Product line architects generate an architecture blueprint that specifies the assets that are to be used to build a specific product line (similar to a Bill of Material).

Figure 1-2 describes a sample prescriptive reuse process.

Figure 1-2 Prescriptive Reuse



Compliance Templates are applied to projects through the Oracle Enterprise Repository Web console and the assets identified in the compliance template are automatically displayed in the project's development environment. In this way, development teams get a jump-start on their development efforts. For more information on Compliance Templates, see *Oracle Enterprise Repository Compliance Templates Guide*.

Developer-Driven Discovery

There are two ways that developers can get assets from the Oracle Enterprise Repository:

- Through the Oracle Enterprise Repository Web Console
- Through their IDE (Note that this functionality is currently available for Eclipse and VS .NET. Developers will be able to download assets from JDeveloper in a future release.)

Developers can come to the Oracle Enterprise Repository to obtain the assets that they would like to use on their projects. The Use-Download function in the Oracle Enterprise Repository provides access to the asset artifacts. In addition, developers can select and download all assets related to the primary asset, ensuring that they have all necessary dependencies. The repository tracks usage and generates usage-based reports.

Developers can also access Oracle Enterprise Repository from their development environment. This means that developers never have to leave their IDE's to get access to the assets that they need.

Automated Usage Detection

Oracle Enterprise Repository tracks and reports on the design-time use of assets. The automated usage detection is tracked through two methods:

- Through the manual asset Use - Download process within Oracle Enterprise Repository or through the development environment
- Automatic usage detection leveraging Software File Identification (SFID)

Software File Identification (SFID) provides the ability to determine asset usage independent of the manual asset Use - Download process. The SFID process tags selected files and asset artifacts 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 process. An instance of usage is recorded by Oracle Enterprise Repository when tagged files within the asset are opened in a developer's IDE. For more information about SFID, see the *Oracle Enterprise Repository Software File Identification (SFID) Guide*.

High Level Use Cases

The "Best Practices" on page 1-1 section of this guide provides an overview of the production and consumption processes, and encompasses the development environment use cases. The individual use cases are described below. Each development environment includes a subset of these use cases.

Submit Files

Through the development environment, developers can select files to submit to the Enterprise Repository. The files are bundled into a .zip format for submission. The developer can submit single and/or compound-payload assets to Oracle Enterprise Repository.

Harvest (BPEL, WSDL, XSD, XSLT)

Oracle Enterprise Repository can harvest BPEL, WSDL, XSD and XSLT files and file directories. When harvested, OER automatically creates assets, populates asset metadata, and generates relationship links based on the information in the artifact files. Note that the Harvester is not restricted to Oracle products - it can be used to harvest standards-based artifacts generated

from any tooling. The harvesting function is available from the command line, and can be integrated into the IDE, or into the build process.

Search Oracle Enterprise Repository

Access to assets and artifacts in the Enterprise Repository is available through the development environment. Through the IDE, developers can search for assets matching various criteria or view assets that may be of interest to a development project project.

View Asset Details

For selected assets, developer can view asset details, such as description, usage history, expected savings, relationships, etc. Within the asset metadata, links to supporting documentation, user guides, test cases, etc., are provided to better enable developers to reuse existing functionality.

Download Artifacts

Developers can download an asset's artifacts (i.e., payload) into their project. Typically an asset payload is the functionality that a developer needs to use a service (such as a WSDL file) or incorporate into their code base (usually a binary).

Prescriptive Reuse

Through the Oracle Enterprise Repository, analysts, architects, technical leads, and others who are involved in the design stages of a project can create a list of assets that fulfills a project's requirements. The list of assets are captured in compliance templates in the repository and the compliance templates are associated with an Oracle Enterprise Repository project.

From within the IDE, you can view a list of assets appearing in all of the Compliance Templates assigned to your project. You can see which of the assets have been used and/or other project members. For more information on compliance templates, see *Oracle Enterprise Repository Compliance Templates Guide*.

Automatic Usage Detection

Oracle Enterprise Repository can automatically detect asset reuse within the development environment. This allows development teams to ensure that they get asset reuse credit, regardless of whether the assets have been downloaded through Oracle Enterprise Repository or pulled from another source, such as the developer's desktop.

For more information about each of these use cases for various IDEs, see Chapter 4, "Using the IDE to Interact with Oracle Enterprise Repository."

Table 1-1 describes the specific use cases supported for each development environment.

Table 1-1 Use Cases and Supported Development Environments

Use Case	JDeveloper	Eclipse	VS .NET
Submit Files		Yes	
Harvest (BPEL, WSDL, XSD, XSLT)	Yes	Yes	Yes
Harvest (SCA)	11g		
Search Oracle Enterprise Repository	11g	Yes	Yes
View Asset Details	11g	Yes	Yes
Download Artifacts	11g	Yes	Yes
Prescriptive Reuse		Yes	Yes
Automatic Usage Detection		Yes	Yes

Table 1-2 describes the supported versions and target audience for each IDE.

Table 1-2 Supported Versions for the IDEs

IDE	Target Audience	Supported Versions
JDeveloper	Oracle Customers	10gR3, 11gR1
Eclipse	Java Developers	3.4 (get details from John Yoder) 3.3 with Sun JDK 5.0 2.0.3 with Sun JDK 5.0
VS .NET	.Net Developers	2008 2005

Related Documentation

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

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

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

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

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

  http://blogs.oracle.com/governance

CHAPTER 2

Configuring Oracle Enterprise Repository to Support Integration with Your IDE

You need to perform some steps in the Oracle Enterprise Repository before configuring the IDE. The steps are as mentioned below:

Install the Harvester

The Harvester can harvest standards-based artifacts generated from any IDE such as Oracle JDeveloper, Eclipse, and VS .NET. The Harvester can be integrated with any of these IDEs and then run with the respective IDE client.

For more information about the Harvester, see *Oracle Enterprise Repository Harvester User Guide*.

Assign IDE Users to Oracle Enterprise Repository Projects

Oracle Enterprise Repository tracks assets produced by projects in any IDE such as Oracle JDeveloper, Eclipse, and VS .NET, as well as assets consumed by the projects created in these IDEs.

For more information about adding a new project and assigning users to the project, see *Oracle Enterprise Repository User Guide.*

Establish Compliance Templates

Compliance Templates describe a particular family of Oracle Enterprise Repository artifacts and are available only for some product configurations such as Eclipse and VS .NET. Compliance templates are required to support the Prescriptive Reuse use cases.

For more information about how to establish compliance templates, see *Oracle Enterprise Repository Compliance Templates Guide.*

Set up Automatic Usage Detection

Software File Identification (SFID) is a process of determining asset usage in Oracle Enterprise Repository. You can use SFID to work with your development environment such as Eclipse and VS .NET. Depending on your IDE, SFID requires the installation of the Oracle Enterprise Repository Plug-in for Workspace Studio, which is an Eclipse-based IDE, or the Oracle Enterprise Repository Plug-in for Visual Studio .NET.

For more information about setting up automatic usage detection using SFID, see *Oracle Enterprise Repository Software File Identification (SFID) Guide.*

CHAPTER 3

Configuring Your IDE to Support Integration with Oracle Enterprise Repository

This chapter contains information on the following topics:

Configuring Oracle JDeveloper

Oracle JDeveloper includes bundled integration with the Apache Ant build system. These integrations provide convenient mechanisms for the invocation of custom Ant tasks such as those provided by the Harvester.

The following sections describe how to configure Oracle JDeveloper to invoke the Harvester from within the developement environment with the corresponding examples:

Enable Harvesting in JDeveloper

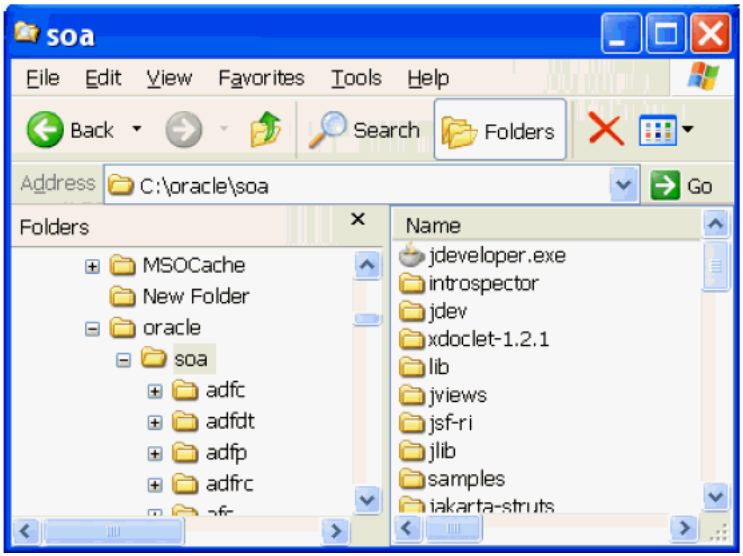
This section contains the following topics:

Enable Harvesting in JDeveloper 10*g*

To configure Oracle JDeveloper to support the integration with Oracle Enterprise Repository, perform the following steps:

1. Navigate to the BEA_HOME\repository103\core\tools\solutions directory and unzip the OER103-SOA-BPM-Harvester.zip file to the Oracle JDeveloper directory. For example, if the jdeveloper.exe file is located in C:\oracle\soa, ensure that the introspector directory is unzipped into that directory, as shown in Figure 3-1.

Figure 3-1 Sample Directory Structure



2. Navigate to the <jdeveloper_home>\harvester directory and right-click the tools.xml file to open in a text editor.

3. Copy all the elements between the <tools> and </tools> elements and paste the copied elements into the tools.xml file in the <jdeveloper_home>\jdev\system\oracle.jdeveloper.10.1.xxxxx directory.

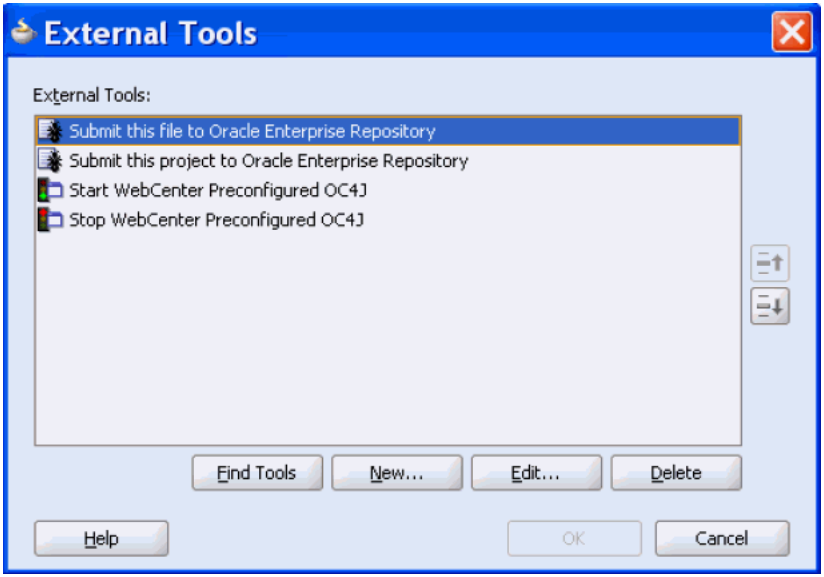
4. Save the tools.xml file in the <jdeveloper_home>\jdev\system\oracle.jdeveloper.10.1.xxxxx directory.

Set Repository Connection Information for JDeveloper

To set repository connection information for Oracle JDeveloper, perform the following steps:

1. Launch Oracle JDeveloper and choose **Tools**, **External Tools**. The External Tools dialog is displayed, as shown in Figure 3-2.

Figure 3-2 External Tools Dialog



2. Select **Submit this file to Oracle Enterprise Repository** and click **Edit**. The Edit External Tool dialog is displayed.
3. In the Properties tab, edit registry.url, registry.username, and registry.password properties to set the Oracle Enterprise Repository URL, username, and password.
4. To point to an external HarvesterSettings.xml file, add a property called "settings.file" and set the value to the URL of the settings file, for example, c:\temp\MyHarvesterSettings.xml.
5. Select an item in the External Tools dialog and click **OK** to submit.
6. Repeat step 2 to step 5 for the Submit this project to Oracle Enterprise Repository external tool.

Note: The steps mentioned in the section are common to both Oracle JDeveloper 10*g*.

Select the Artifacts to Harvest for Oracle JDeveloper

After you configure Oracle JDeveloper to integrate with the Harvester and specify the correct repository connection information, you can harvest the artifacts in both Oracle JDeveloper 10*g*. Perform the following steps to harvest artifacts from Oracle JDeveloper:

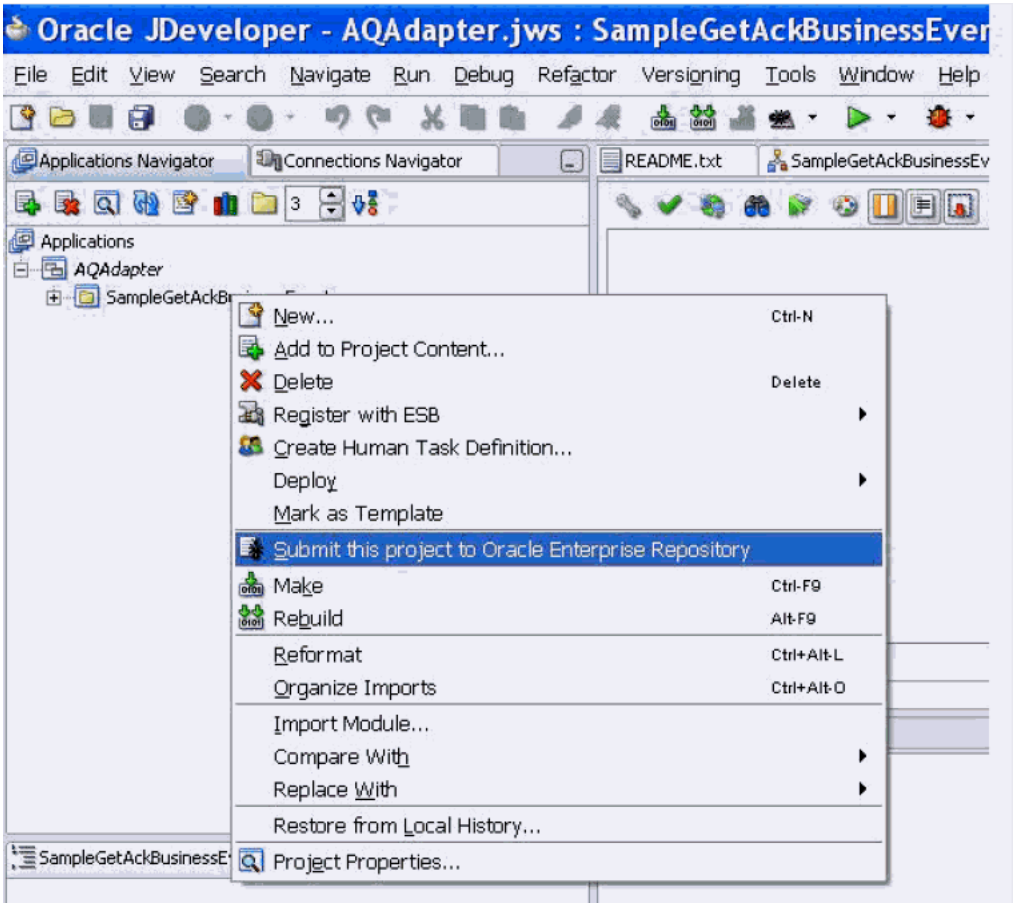
1. Select a file in the Application Navigator.

2. Right-click the file and choose either of the following options:

- **Submit this file to Oracle Enterprise Repository** - This option will harvest just the selected file. This works for zips and jars too.
- **Submit this project to Oracle Enterprise Repository** - This option, as displayed in Figure 3-3, will harvest the entire project containing the selected file.

Note: Using this option, you can only submit one project at a time from JDeveloper, but you can not submit the complete application.

Figure 3-3 Submitting a JDeveloper Project to Oracle Enterprise Repository



In Oracle JDeveloper, you can also right-click an XSL, WSDL, XSD, or .zip file and choose Submit this file to Oracle Enterprise Repository from the context menu.

Note: The right-click option is not available for BPEL files in the Applications Navigator, but if you open the BPEL file and click the Source tab, you can submit the file to the Oracle Enterprise Repository.

Configuring Eclipse

The Harvester integrates the Oracle SOA Suite artifacts to Oracle Enterprise Repository to support the visibility, impact analysis, and reusability use cases. This section describes the various steps involved in configuring Eclipse to support integration with Oracle Enterprise Repository:

Enable Harvesting in Eclipse

This section describes how to introspect sample artifacts into Oracle Enterprise Repository using Eclipse:

Setting up Eclipse Environment to use Harvester as an "External Program"

1. In Eclipse, click **Run**, **External Tools**. The External Tools dialog is displayed.
2. Right-click **Program**, and then select **New**.
3. Enter the following details in the External Tools dialog, as shown in Figure 3-4.

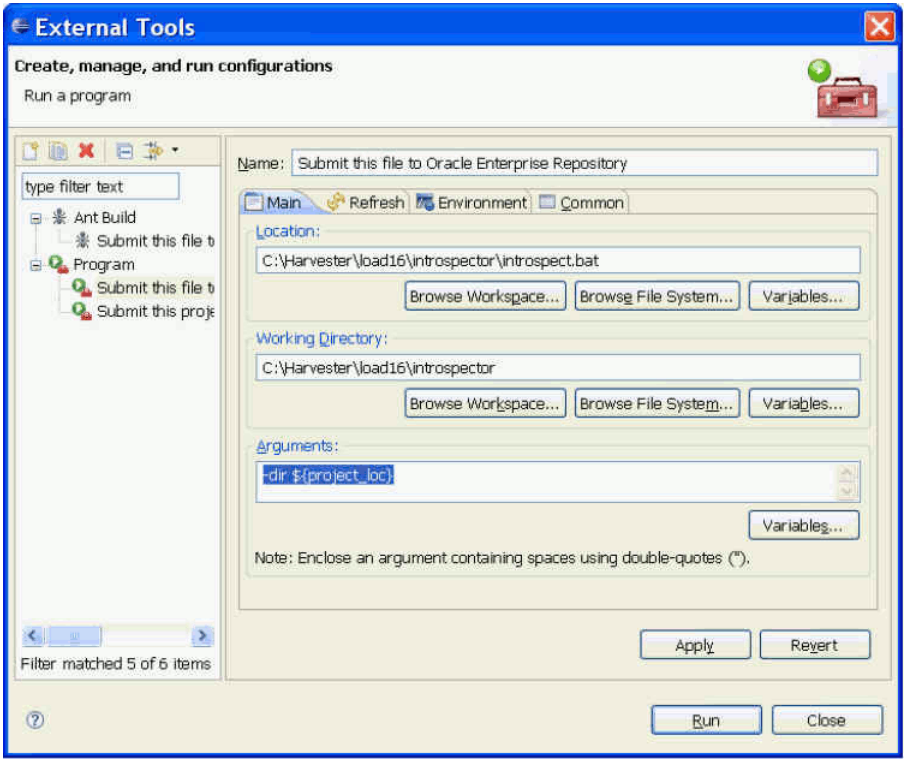
– In the Name field, type `Submit this project to Oracle Enterprise Repository.`

– In the Location field, type `<Harvester Home>\introspector\introspect.bat`. You can also browse the introspect.bat file using the Browse button.

– In the Working Directory field, type `<Harvester Home>\introspector`. You can also browse the introspect.bat file using the Browse button.

– In the Arguments field, type `-dir ${project_loc}`.

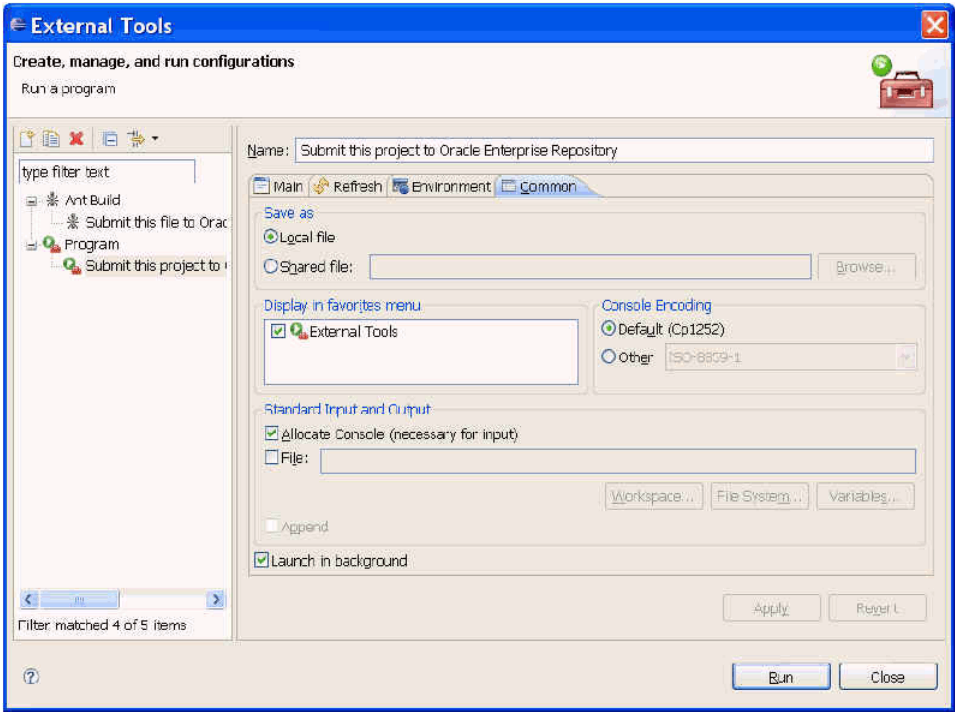
Figure 3-4 External Tool Dialog



4. Click the **Common** tab.

5. In the Display in favorites menu pane, enable External Tools, as shown in Figure 3-5.

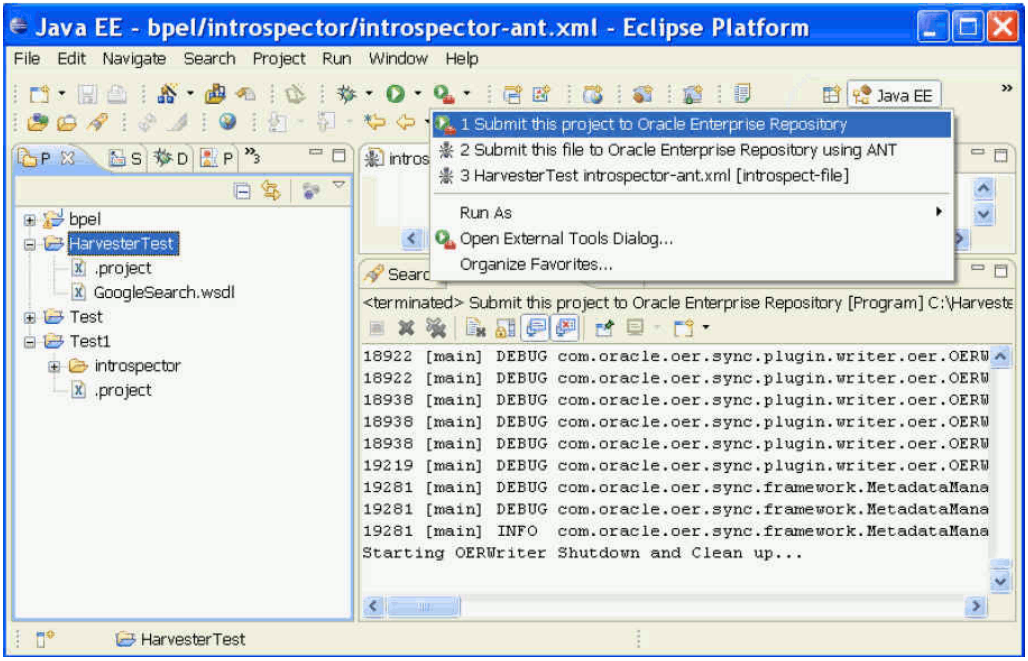
Figure 3-5 External Tools Dialog - Common Tab



Harvesting in Eclipse Environment using “External Program”

1. In Eclipse, click **New**, **Project**, **General**, **Project** to create a new eclipse project.
2. Browse for any WSDL file in the file system, copy it and paste into the just created project.
3. Select the project and then click **Submit this project to Oracle Enterprise Repository**. This invokes the Harvester and submits all the artifacts in the project, as shown in .

Figure 3-6 Java EE - Eclipse Platform



4. Repeat the above steps to create another program called Submit this file to Oracle Enterprise Repository by using –dir ${resource_loc} instead of the ${project_loc} variable and to submit the individual files.

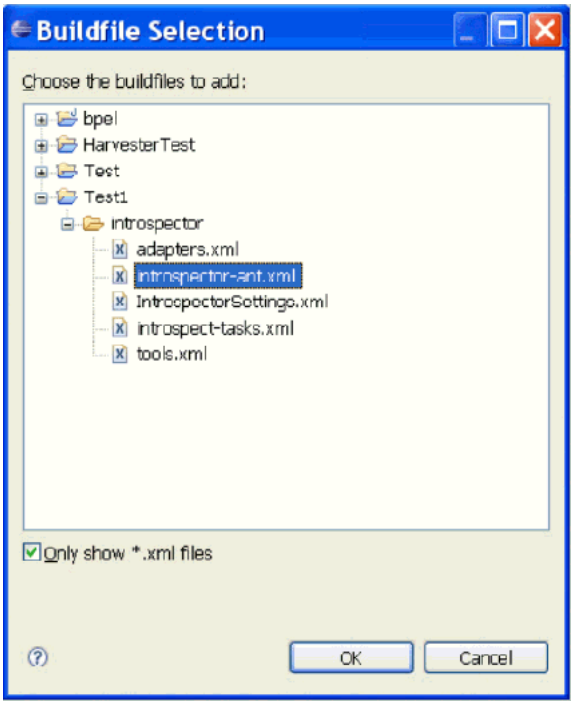
Setting up Eclipse Environment to use Harvester via ANT

1. Copy the Harvester directory to any project in the Eclipse workspace.
2. In the Eclipse workspace, click **Window**, **Preferences**. The Preferences dialog is displayed.
3. Select **Ant**, **Runtime**. The Runtime page is displayed.
4. Select **Global Entries** and then click **Add Folder**. The Browse for Folder dialog is displayed.
5. Select the introspector and click **OK**.
6. In the Eclipse workspace, click **Window**, **Show View**, **Other**, and then **Ant**. The Ant view is displayed.

7. Right-click the ANT view window and select **Add Build Files** option. The Buildfile Selection dialog is displayed.

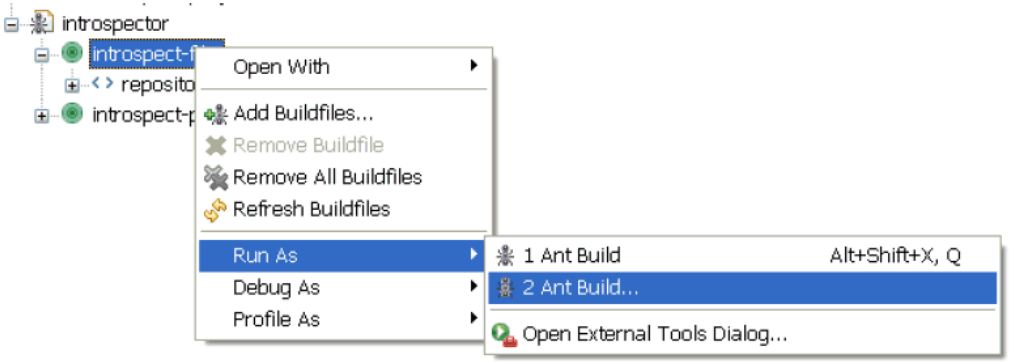
8. Select **introspector-ant.xml** from the Harvester directory, as shown in Figure 3-7.

Figure 3-7 BuildFile Selection Dialog



9. In the ANT window, expand the introspector node and right-click **introspect-file** and select **Run As**, **Ant Build**, as shown in Figure 3-8. The External Tools dialog is displayed.

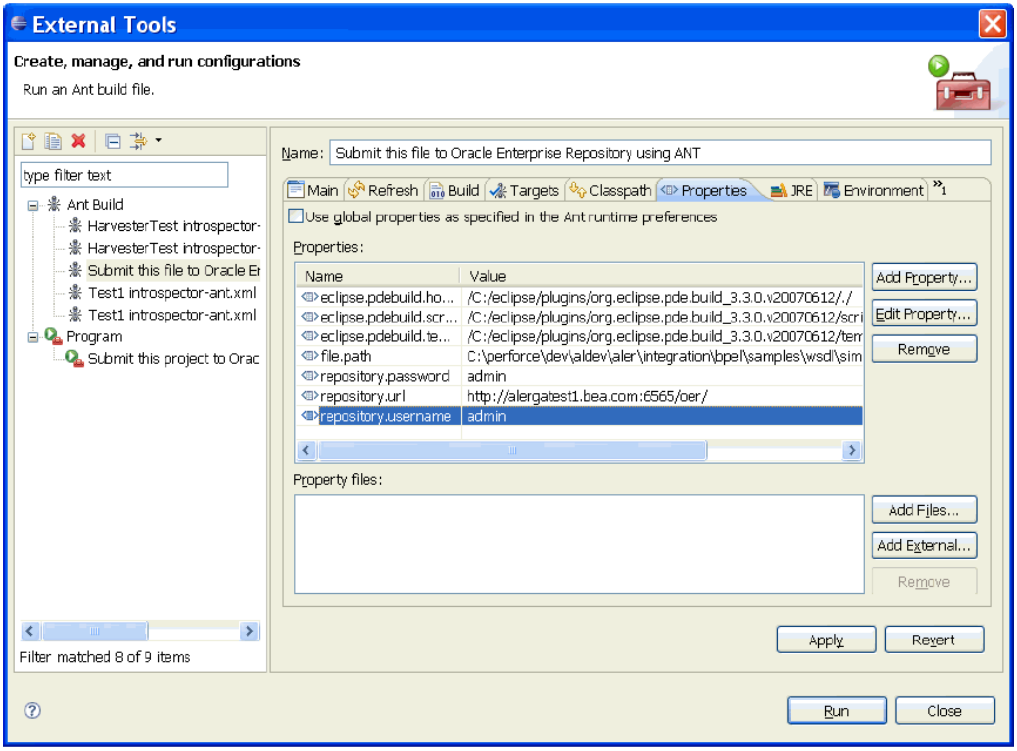
Figure 3-8 The ANT Window



10. Enter the following details in the External Tools dialog:
    - Name: Submit this file to Oracle Enterprise Repository using ANT.
    - Click the **Classpath** tab and select **User Entries**.
    - Click **Add External JARs** and select all the jars under <Harvester Home>\introspector and <Harvester Home>\introspector\lib directories.
    - Click the **Common** tab. In the Display in favorites menu pane, enable **External Tools**.
    - Click the **Properties** tab and uncheck the **Use global properties as specified in the Ant runtime preferences** option.
11. Click the **Add Property** button and add the following properties as shown below:
    - file.path = ${resource_loc}
    - repository.url = http://localhost:7101/oer/
    - repository.username = admin
    - repository.password = admin

After the configuration of the Properties tab, the External Tools dialog is displayed, as shown in Figure 3-9.

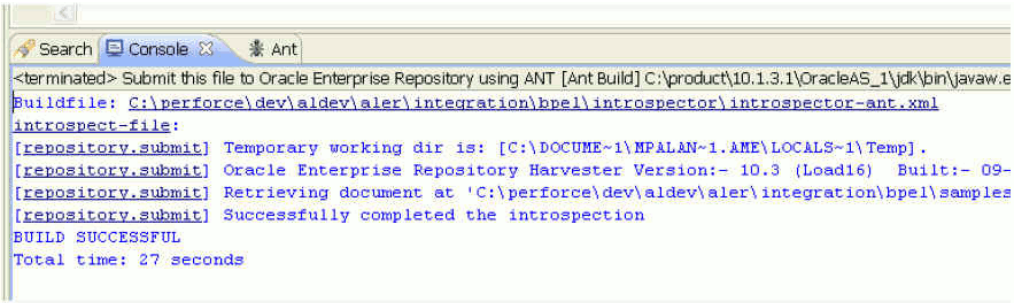
Figure 3-9 External Tools Dialog - Properties Tab



Harvesting in Eclipse Environment using ANT

1. In Eclipse, click **New**, **Project**, **General**, **Project** to create a new eclipse project.
2. Browse for any WSDL file in the file system, copy and paste it into the project that you just created.
3. Select the project and then click **Submit this file to Oracle Enterprise Repository using ANT**. This invokes Harvester and submits all the artifacts in the project, as shown in Figure 3-10.

Figure 3-10 The Console Window



Configure the Oracle Enterprise Repository Plug-ins

This section describes the steps to configure the Oracle Enterprise Repository Plug-ins for repository access and the prerequisities to enable this configuration. This section contains the following topics:

Configuring the Oracle Enterprise Repository Plug-ins for Repository Access in Workshop for WebLogic

For instructions on installing the Oracle Enterprise Repository plug-ins for repository access within the Eclipse-based Oracle Workshop for WebLogic IDE, see *Oracle Enterprise Repository Installation Guide*.

Uninstalling the Oracle Enterprise Repository Plug-ins

When uninstalling the Oracle Enterprise Repository plug-ins, Oracle Workshop for WebLogic will also be removed if it was installed by the Oracle Enterprise Repository installer. This may not be desirable if other plug-ins were since added to Oracle Workshop for WebLogic. To avoid uninstalling additional Oracle Workshop for WebLogic plug-ins, deselect the Oracle Enterprise Repository plug-ins for Eclipse option from the list of components to uninstall.

Installing Products After Installing Oracle Enterprise Repository

If Oracle Service Bus applications are installed after the Oracle Enterprise Repository plug-in is installed, then Eclipse must be launched using the `-clean` flag.

Prerequisites for Using the Oracle Enterprise Repository Plug-ins with Workshop for WebLogic

You should complete the prerequisites described in this section before using the Oracle Enterprise Repository plug-ins for Oracle Workshop for WebLogic.

- "Installing the Eclipse Solution Pack" on page 3-14
- "Assign Users to an Oracle Enterprise Repository Project" on page 3-15
- "Enabling the Assets-in-Progress Properties" on page 3-15
- "Enabling Assembly Model Submission Properties" on page 3-16
- "SiteMinder" on page 3-16
- "Java JDK" on page 3-16
- "XML Parsing" on page 3-16

Installing the Eclipse Solution Pack

In order to be able to submit Oracle Service Bus projects as assembly models to the repository, you must first import the Eclipse Solution Pack that is bundled with your installation into Oracle Enterprise Repository.

1. Start the Oracle Enterprise Repository Import/Export tool, as described in the *Oracle Enterprise Repository Import/Export Guide*.
2. Select the **Import** tab.
3. Navigate to the *BEA_HOME*`\repository103\core\tools\solutions` folder.
4. Select the `OER103-Eclipse-Solution-Pack.zip` as the target file to import into Oracle Enterprise Repository.
5. Click **Next**, and then click **Next** again to start the import process.
6. Click **Finish** to complete the process.

7. After importing the Eclipse Solution Pack, you must reestablish connectivity to the Oracle Enterprise Repository plug-ins by using the Eclipse Preferences page, as described in "Configuring the Oracle Enterprise Repository Connection" on page 5-2.

Once connectivity is established, then the Oracle Enterprise Repository plug-in imports all the necessary asset types, taxonomy, relationships, and other entities for application integration to Oracle Enterprise Repository. Once these entities are imported, they will be available whenever you connect the Oracle Enterprise Repository plug-in to the enterprise repository.

Assign Users to an Oracle Enterprise Repository Project

In order to download assets from the repository, users must be assigned to at least one Oracle Enterprise Repository project. An Oracle Enterprise Repository project administrator can assign users to projects using the Oracle Enterprise Repository Projects page.

Obtain the Eclipse integration path from the Oracle Enterprise Repository administrator. (For example, `http://appserver.example.com/aler-web/eclipse`).

Enabling the Assets-in-Progress Properties

Two system settings must be enabled in order to activate Assets-in-Progress when using Oracle Workshop for WebLogic with Oracle Enterprise Repository.

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

1. Click **System Settings**.
2. Click **General Settings** in the System Settings section.
3. Enter the property **cmee.asset.in-progress** in the Enable New System Setting box and click **Enable** to reveal this hidden property.
4. Make sure the **Asset in Progress** property is set to **True**.
5. Click **Save**.
6. Enter the property **cmee.asset.in-progress.visible** in the Enable New System Setting box and click **Enable** to reveal this hidden property.
7. Make sure the **Asset in Progress** property is set to **True**.
8. Click **Save**.

The Registration Status drop-down menu will now appear in the Search section on the AquaLogic Enterprise Repository Assets screen. For more information about Assets-in-Progress, see *Oracle Enterprise Repository Registrar Guide*.

Enabling Assembly Model Submission Properties

In order to be able to submit Oracle Service Bus projects as assembly models to the repository, this capability must be enabled in the Oracle Enterprise Repository System Settings. You can also enable the logging of asset submissions from external endpoints.

1. Click **System Settings** in the sidebar on the Oracle Enterprise Repository **Admin** page.
2. Enter the property **cmee.tooling.submission.enabled** in the Enable New System Setting box and click **Enable** to reveal this hidden property.
3. Set the **Asset Submission from Integrated Endpoint** property to **True** to enable asset submissions generated through integrations with external endpoints.
4. Click **Save**.
5. Enter the property **cmee.tooling.submission.logging** in the Enable New System Setting box and click **Enable** to reveal this hidden property.
6. Set the **Submission Logging of Integrated Endpoint** property to **True** to enable the logging of import/export jobs controlling asset submissions through integrations with external endpoints.
7. Click **Save**.

For more information about System Settings, see *Oracle Enterprise Repository Administration Guide*.

SiteMinder

If Oracle Enterprise Repository is or will be configured to be secured by Siteminder, the policy server must be configured to ignore (or unprotect) the following URL:

```
http://appserver.example.com/aler-web/eclipse/
```

Java JDK

The Java Cryptography Extension (JCE) is required. It is provided in JDK v1.4, and is available as an optional package in JDK 1.2.x through 1.5.x. Note that Oracle Enterprise Repository plug-ins for use with Eclipse 3.x require JDK v 1.5.x or later.

XML Parsing

Since Editor and Viewer metadata is represented as CDATA-escaped XML, some XML parsers may exceed their entity expansion limit when communicating with Oracle Enterprise Repository.

For example, if you have defined a large number of Asset Types in Oracle Enterprise Repository, then you may need to increase the Entity Expansion Limit of your XML parser.

On some popular parsers, the default entity expansion limit is set to 64,000. This limit can be increased on JAXP-compliant processors by passing a command-line parameter called `entityExpansionLimit`. The `entityExpansionLimit` can be increased by passing a VM argument on the Eclipse command-line (modify the Eclipse desktop shortcut). For example, set the target of the shortcut to the following:

```
  c:\eclipse\eclipse.exe -debug -consolelog -vmargs
-DentityExpansionLimit=1048576
```

Configure the Oracle Enterprise Repository Preferences

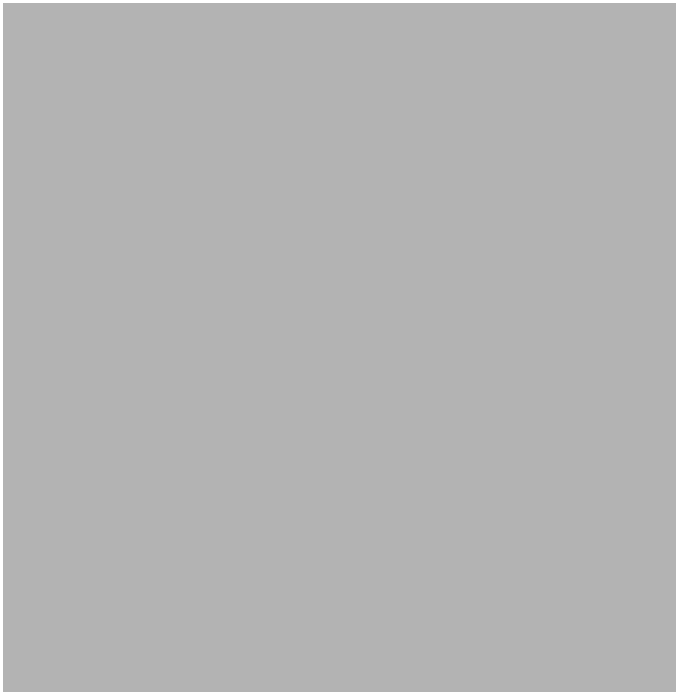
This section describes the steps to configure the Oracle Enterprise Repository connection.

When you invoke an action on a repository, such as querying or publishing assets, before repository connectivity has been established, then the Connect to Enterprise Repository wizard will either be automatically displayed (in the case of querying the repository), or will be launched by an explicit user gesture.

Note: If credential information had been specified in a previous session, the wizard will display this persisted information when it is launched.

1. In the Credentials area, enter the server location and login credentials, as follows:
   - **Repository URL**: the URL of the repository server. The URL must include the host, port, and Oracle Enterprise Repository server name. For example, `http://localhost:7001/oer`.
   - **User Name**: user name to gain access to the repository.
   - **Password**: password to gain access to the repository.

Figure 3-11 Establish Enterprise Repository Connection

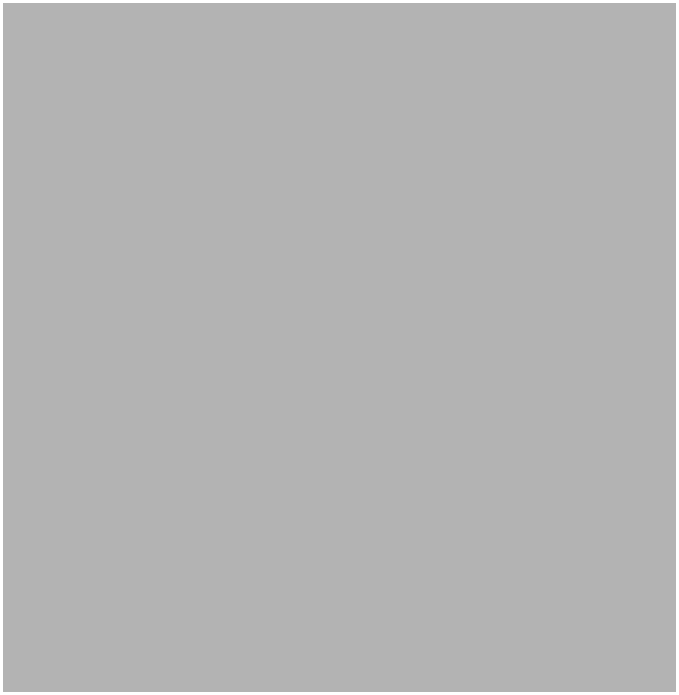


2. Click the **Establish Connection** button to ensure enterprise repository connectivity.

   If a connection cannot be established, an appropriate error message will be displayed.

3. Once connectivity is established, you can either:
   - Click **Finish** to exit.
   - Click **Next** to select your workspace preferences (skip to Step 4).

Figure 3-12 Specify Workspace Preferences



4. Once connectivity is established, you can specify your workspace preferences:
   - Enter a **Model Namespace** to use as a default for your all of your projects. The Namespace provides a means to organize your models, with the Namespace pre-pended to the names of all the assets in the model in the repository. However, you can change the Namespace on a project-by-project basis (such as when submitting assets), and the new Namespace will only be saved for that project, but will not affect the Workspace Preference name.
   - Select a **Repository project** in Oracle Enterprise Repository that the submitted model will be associated with. Asset usage is tracked in the repository and attributed to repository projects, which typically represent software development programs, business initiatives, etc.
   - **Enable usage detection**: If you selected an Oracle Enterprise Repository project as the workspace default, usage detection will be enabled for the default Oracle Enterprise Repository project. For more information about workspace preferences, see Chapter 5, "Setting Eclipse Preferences for Oracle Enterprise Repository."

5. The **Artifact Store** area displays the name of a preconfigured Artifact Store that the submitted assets will be associated with. Artifact Stores are Oracle Enterprise Repository's representation of Software Configuration Management (SCM) systems. SCMs contain the master artifacts, which are referenced by URLs in Artifact Assets and in asset FileInfos. Only those Artifact Stores that have been defined by an Oracle Enterprise Repository administrator will appear in the **Name** list. If an SCM does not appear in the **Name** list, an administrator must add it to the Oracle Enterprise Repository instance. The **Details** box may also display some additional information about the Artifact Store.
6. Click **Finish** to exit.

Enable Automatic Usage Detection

Oracle Enterprise Repository can automatically detect asset reuse within the development environment. This allows development teams to ensure that they get asset reuse credit, regardless of whether the assets have been downloaded through Oracle Enterprise Repository or pulled from another source, such as the developer's desktop. Automated Usage Detection relies on a fingerprinting process, called *Software File Identification* (SFID), which tags selected files within an asset with a unique ID. This SFID is then used to detect when and where an asset is used, even if the asset was acquired through means other than the Oracle Enterprise Repository Use - Download process. An instance of usage is recorded by Oracle Enterprise Repository when tagged files within the asset are brought into the developer's IDE, and a new build or build clean occurs.

For more information, see *Oracle Enterprise Repository Software File Identification Guide.*

Note: Automated Usage Detection requires the installation of the Oracle Enterprise Repository Plug-in for Oracle Workshop for WebLogic, and is currently compatible only with Eclipse and Eclipse-based IDEs.

1. On the Window menu, click **Preferences**.
2. Select **Oracle Enterprise Repository**.
3. Select **Workspace Automatic Usage Detection**.

Figure 3-13 Preferences - Workspace Automatic Usage Detection

4. Click the **Detect usage in workspace projects** check box, and then activate the desired usage detection features, as appropriate:
   - Enable usage detection in new workspace projects by default – monitors new projects
   - Detect usage of files on classpath – monitors files on classpath.
   - Detect usage of Java Runtime JARs – monitors Java Runtime JARs
   - Cache calculated SFIDs (recommended) – caches calculated SFIDs (enhances performance)
   - Detect usage of files matching pattern – monitors files matching specified patterns
5. Enter the appropriate information in the **File Pattern** text boxes:
   - Include File Pattern – Includes indicated file pattern

  - Exclude File Pattern – Excludes the indicated file pattern

6. Specify which project directories will be targets for automatic usage detection by using the individual check boxes or by using the **Select All** and/or **Unselect All** buttons.
7. Click **OK** when finished.

Configuring VS .NET

Oracle Enterprise Repository integration with Visual Studio .NET provides users with the ability to easily search for and use assets from the repository without leaving the VS .NET IDE environment. Assets and any associated artifacts are downloaded directly to your VS .NET solution. Repository Access within the VS .NET solution also provides a view into Oracle Enterprise Repository that enables you to download artifacts and assets from the repository, query the repository, and view the contents of the repository.

This section contains the following topics:

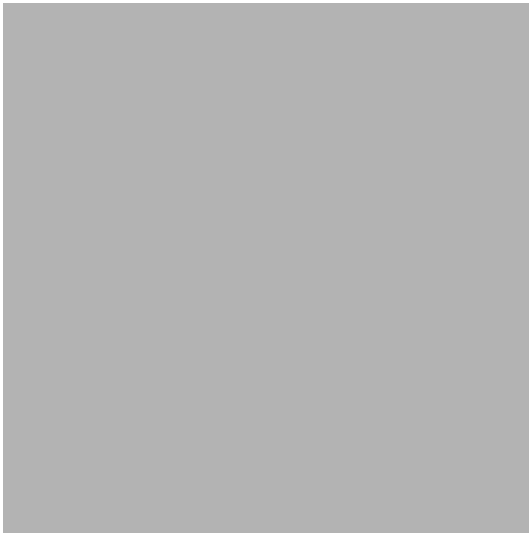
Enable Harvesting in VS .NET

1. In Microsoft Visual Studio, click **Tools**, **External Tools**. The External Tools dialog is displayed.
2. Click Add. A entry is added to the Menu Contents pane.
3. Enter the following details in the External Tools dialog, as shown in Figure 3-4.
   - In the Title field, type `OER - Harvest`.
   - In the Comman field, click the Browse button at the end of the field and select the `introspect.bat` file in the introspector directory.
   - In the Arguments field, type the `-dir` parameter. Click the right-arrow at the end of this field and select **ItemPath** from the menu.

– In the Initial Directory field, type the location of the introspector directory.

– Select the **Use Output Window** option. This option enables you to monitor progress.

Figure 3-14 External Tool Dialog



4. Click **OK**.
5. Select the WSDL file in the Microsoft Visual Studio and click **Tools**, **OER - Harvest**. The Output window is displayed with the Shutdown and Clean up messages indicating that the introspection is complete.
6. Open the Oracle Enterprise Repository home page with your username/password credentials.
7. In Assets pane, enter the name of the WSDL as the search criteria in the Enter Search String field, and then click **Search**. The search results are displayed in the right pane.
8. Select the service in the search results section, the details of the service are displayed in the bottom pane.
9. Click the **Navigator** button to view the relationships.
10. In the Oracle Enterprise Repository main page, click **Admin**, and then **System Settings**. The System Settings page is displayed.
11. Enter Show in the Search field, and set the Show System-Supplied Relationships option to True.

12. Click **Save** at the bottom of the page.
13. In the Oracle Enterprise Repository main page, click **Assets** and repeat the same search that you performed in step 7. The automatic relationships that were not imported earlier are now imported.

Configure the Oracle Enterprise Repository Plug-ins

Oracle Enterprise Repository can automatically detect asset reuse within the development environment. This allows development teams to ensure that they get asset reuse credit, regardless of whether the assets have been downloaded through Oracle Enterprise Repository. For more information, see "Enable Automatic Usage Detection" on page 3-28

Prerequisities

- Microsoft Visual Studio 2008.
- Microsoft Visual J# 2005 runtime. (If J# is not installed on your machine, the installer will prompt you download the correct version from Microsoft.)
- The VS .NET **Always show solution** option should be selected (Tools -> Options -> Projects and Solutions -> General).
- Users must be assigned to at least one Oracle Enterprise Repository project. A Project Administrator can assign users to projects using the Oracle Enterprise Repository Projects page.
- If your Oracle Enterprise Repository is or will be secured by Siteminder, you will need to configure the policy server to ignore (or unprotect) the following URL to allow the OpenAPI integration to function properly:
  - http://appserver.example.com/OER/services/

Installation

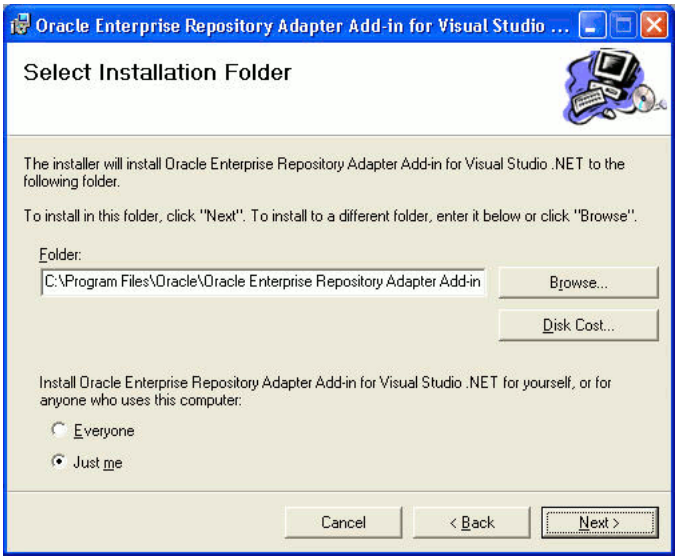
1. Download the VS .NET plug-in Zip file from your Oracle Enterprise Repository instance at the following URL:

http://appserver.example.com/oer-web/integration/dotnet/OER103-VisualStudioAddin.zip

2. Unzip the `OER103-VisualStudioAddin.zip` file.
3. Locate and run the `setup.exe` program.

4. Follow the prompts to select installation parameters.

Figure 3-15 Oracle Enterprise Repository Adapter Add-in for Visual Studio



5. Click **Finish** to complete the installation.

6. Follow the instructions in "Configure the Connection to Oracle Enterprise Repository" on page 3-25 to configure and establish a connection to an Oracle Enterprise Repository instance from VS. NET.

Configure the Connection to Oracle Enterprise Repository

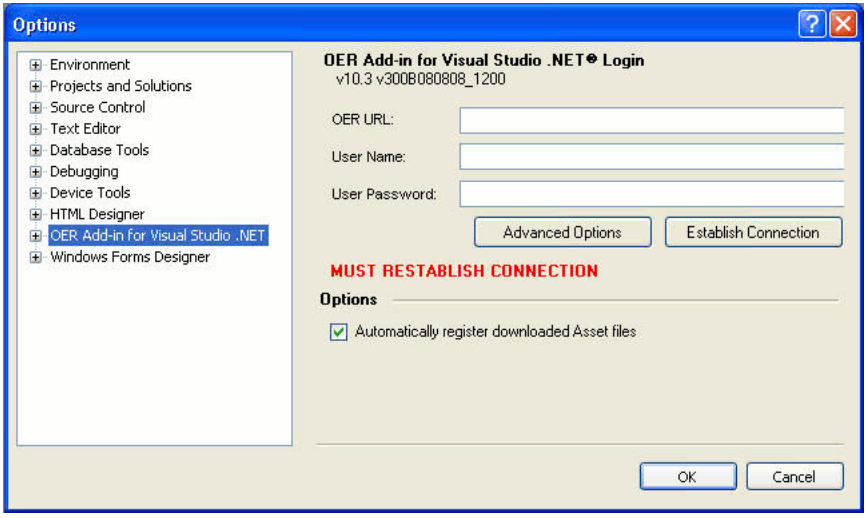
Follow these steps to configure and establish a connection to an Oracle Enterprise Repository instance from VS. NET.

1. Launch Visual Studio .NET.

2. Open the **Tools** menu and click **Options**.

3. On the list of options, click the **OER Add-in for Visual Studio .NET** option, as shown in Figure 3-16, and provide the required login information.

Figure 3-16 The Options dialog



- Oracle Enterprise Repository URL
- The URL of the Oracle Enterprise Repository instance. For example: http://appserver.example.com/OER

Note: Do not include the index.jsp used in the default home page as part of the URL.

- **User Name:** The user name to connect as.
- **User Password:** The password to connect with. Passwords are case-sensitive.
- **Establish Connection:** Click to verify a valid connection.
- **Automatically register downloaded Asset files:** If selected, downloaded asset files are registered with the Windows Registry, as appropriate. This may be overridden on an case-by-case basis for each asset download.

4. Click the **Establish Connection** button to connect to the Oracle Enterprise Repository instance you specified.
5. Optionally, click the **Advanced** button to enable additional Oracle Enterprise Repository options:
   – Usage detection for VS .NET Solution Projects

– Automated usage detection of referenced DLLs, WSDLs, and allow local caching of SFIDs (if SFID is enabled at your installation)

– File name patterns to include and exclude

6. Click **OK** when finished.

Assign an Oracle Enterprise Repository Project to a .NET Solution

In order to track the usage of downloaded assets, an Oracle Enterprise Repository project must be assigned to a .NET solution.

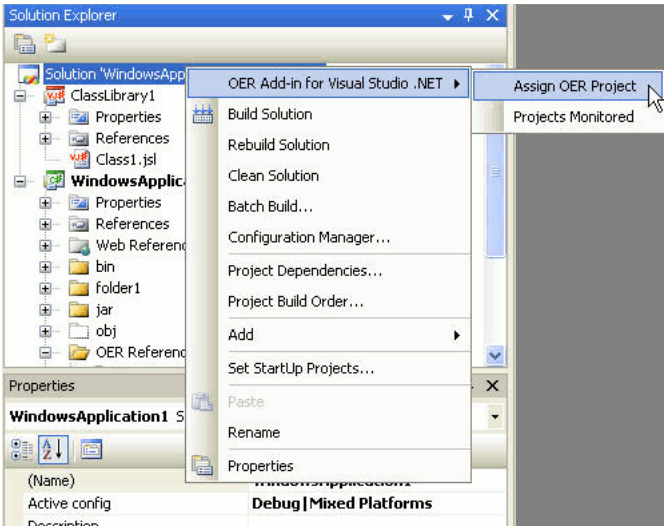
Note: Before using this feature, you must be assigned to at least one Oracle Enterprise Repository Project by a Project Administrator.

1. Open the .NET Solution Explorer.

2. Right-click a solution in the file tree and select the **Oracle Enterprise Repository Add-in for Visual Studio .NET** option from the context menu.

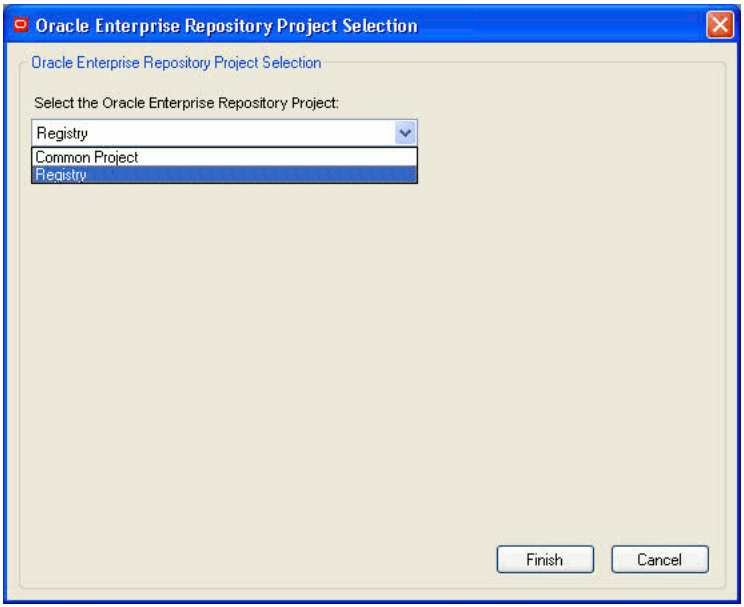
3. Click **Assign Oracle Enterprise Repository Project** from the submenu, as shown in Figure 3-17.

Figure 3-17 The Solution Explorer Window



4. In the Project Selection window, use the **Select the Oracle Enterprise Repository Project** drop-down list to view the Oracle Enterprise Repository projects that you are assigned to, as shown in Figure 3-18.

Figure 3-18 Oracle Enterprise Repository Project Selection Dialog



Note: If the list is empty, you have not been assigned to any projects and the procedure must be canceled.

5. Select an Oracle Enterprise Repository project from the list.
6. Click **Finish** to save your changes.

Enable Automatic Usage Detection

Follow these steps to enable advanced configuration options, such as enabling automatic usage detection of DLLs, WSDLs, local caching of SFIDs, and file pattern detection.

Overview of SFID

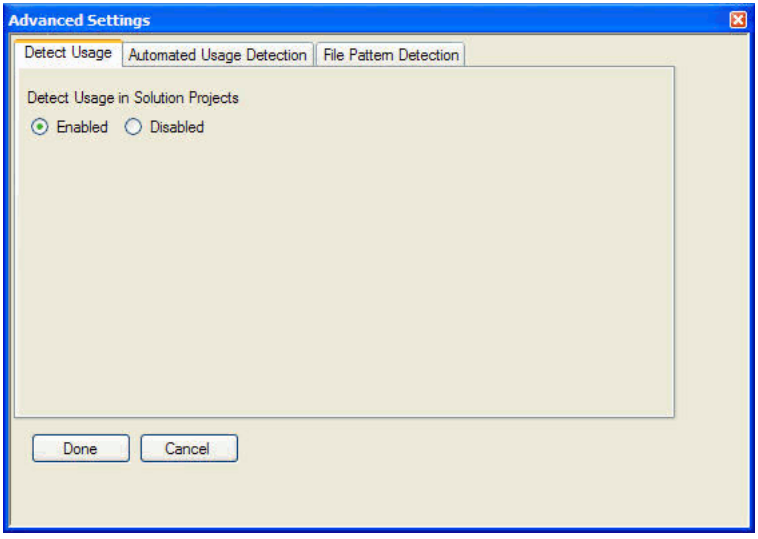
If SFID is enabled at your installation, Oracle Enterprise Repository can automatically detect asset reuse within the development environment. This allows development teams to ensure that they get asset reuse credit, regardless of whether the assets have been downloaded through Oracle

Enterprise Repository. Automated Usage Detection relies on a fingerprinting process, called Software File Identification (SFID), which tags selected files within an asset with a unique ID. This SFID is then used to detect when and where an asset is used, even if the asset was acquired through means other than the Oracle Enterprise Repository Use - Download process. An instance of usage is recorded by Oracle Enterprise Repository when tagged files within the asset are brought into the developer's IDE, and a new build or build clean occurs.

Configuring Automatic Usage Detection

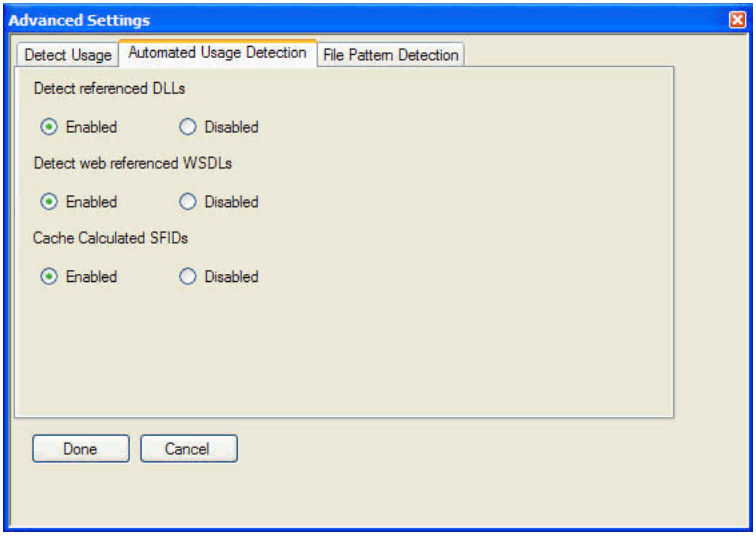
1. Launch Visual Studio .NET.
2. Open the **Tools** menu and click **Options**.
3. On the list of options, click **Oracle Enterprise Repository Add-in for Visual Studio .NET** to reopen the Login window.
4. Click the **Advanced Options** button to open the Advanced Settings window. Use the **Detect Usage** tab to enable usage detection for VS .NET Solution Projects, as shown in Figure 3-19.

Figure 3-19 The Advanced Settings Dialog - Detect Usage Tab



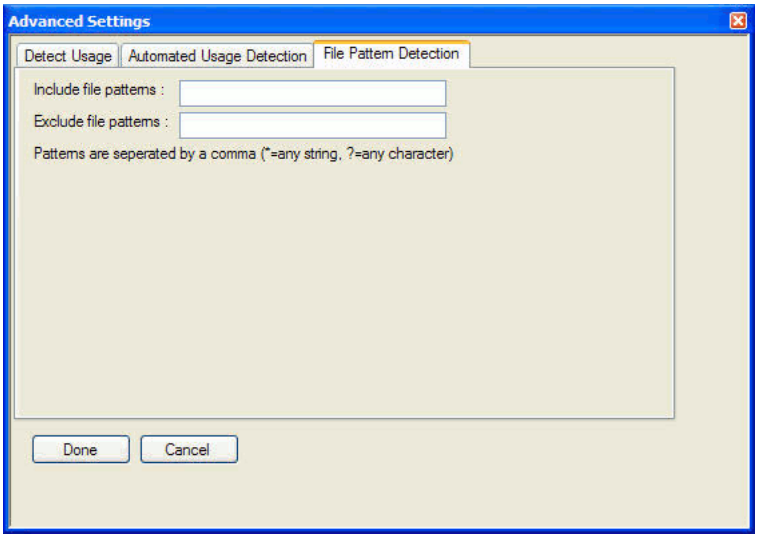
5. Click the **Automated Usage Detection** tab to enable usage detection of referenced DLLs, WSDLs, and allow local caching of SFIDs, as shown in Figure 3-20.

Figure 3-20 The Advanced Settings Dialog - Automated Usage Detection Tab



6. Click the **File Pattern Detection** tab to specify include and exclude file name patterns, as shown in Figure 3-21.

Figure 3-21 The Advanced Settings Dialog - File Pattern Detection Tab



7. Click **Done** to save your settings.

CHAPTER 4

Using the IDE to Interact with Oracle Enterprise Repository

This chapter describes using the various IDEs to interact with Oracle Enterprise Repository. This chapter provides an overview of the production and consumption processes, and encompasses the development environment use cases. It contains the following topics:

Using Oracle JDeveloper

The Oracle JDeveloper development environment use cases are as follows:

Submit Files

Through Oracle JDeveloper, you can select files to submit to Oracle Enterprise Repository. The files are bundled into a .zip format for submission. You can submit single and/or compound-payload assets to Oracle Enterprise Repository.

Harvest Artifacts

Oracle Enterprise Repository can harvest BPEL, WSDL, XSD, and XSLT files and file directories. After harvesting, Oracle Enterprise Repository automatically creates assets, populates asset metadata, and generates relationship links based on the information in the artifact files. The harvesting function is available from the command line, and can be integrated into Oracle JDeveloper or into the build process.

Note: The Harvester is not restricted to Oracle products, it can be used to harvest standards-based artifacts generated from any tooling.

Search Oracle Enterprise Repository

You can access the assets and artifacts available in the Oracle Enterprise Repository through Oracle JDeveloper. Through Oracle JDeveloper, you can search for assets matching various criteria or view assets that may be of interest to a development project.

View Asset Details

For selected assets, you can view asset details such as description, usage history, expected savings, and relationships. Within the asset metadata, links to the supporting documentation, user guides, test cases are provided to better enable you to reuse the existing functionality.

Download Artifacts

You can download an asset's artifacts (i.e., payload) into your project. Typically an asset payload is the functionality that you need to use a service (such as a WSDL file) or incorporate into your code base (usually a binary).

Prescriptive Reuse

Through the Oracle Enterprise Repository, analysts, architects, technical leads, and others who are involved in the design stages of a project can create a list of assets that fulfills a project's requirements. The list of assets are captured in compliance templates in the repository and the compliance templates are associated with an Oracle Enterprise Repository project.

From within the Oracle JDeveloper, you can view a list of assets appearing in all of the Compliance Templates assigned to your project. You can see which of the assets have been used and/or other project members. For more information on compliance templates, see *Oracle Enterprise Repository Compliance Templates Guide*.

Automatic Usage Detection

Oracle Enterprise Repository can automatically detect asset reuse within Oracle JDeveloper. This allows development teams to ensure that they get asset reuse credit, regardless of whether the assets have been downloaded through Oracle Enterprise Repository or pulled from another source such as the developer's desktop.

Using Eclipse

The Oracle Enterprise Repository plug-in for Eclipse development environment use cases are as follows:

Submit Files

The Oracle Enterprise Repository plug-in for Eclipse allows you to select files to submit to the Oracle Enterprise Repository. It packages the files into a .zip format for archive submission. The Archive Submission Wizard allows you to submit single and/or compound-payload assets to Oracle Enterprise Repository via an archive ZIP file.

Harvest Artifacts

Oracle Enterprise Repository can harvest BPEL, WSDL, XSD, and XSLT files and file directories. After harvesting, Oracle Enterprise Repository automatically creates assets, populates asset metadata, and generates relationship links based on the information in the artifact files. The harvesting function is available from the command line, and can be integrated into Eclipse or into the build process.

Search Oracle Enterprise Repository

You can access the assets and artifacts available in the Oracle Enterprise Repository through the Oracle Enterprise Repository plug-in for Eclipse. Through Eclipse, you can search for assets matching various criteria or view assets that may be of interest to a development project.

View Asset Details

For selected assets, you can view asset details such as description, usage history, expected savings, and relationships. Within the asset metadata, links to the supporting documentation, user guides, test cases are provided to better enable you to reuse the existing functionality.

Download Artifacts

You can download an asset's artifacts (i.e., payload) into your project. Typically an asset payload is the functionality that you need to use a service (such as a WSDL file) or incorporate into your code base (usually a binary).

Prescriptive Reuse

Through the Oracle Enterprise Repository, analysts, architects, technical leads, and others who are involved in the design stages of a project can create a list of assets that fulfills a project's requirements. The list of assets are captured in compliance templates in the repository and the compliance templates are associated with an Oracle Enterprise Repository project.

From within the Oracle Enterprise Repository plug-in for Eclipse, you can view a list of assets appearing in all of the Compliance Templates assigned to your project. You can see which of the assets have been used and/or other project members. For more information on compliance templates, see *Oracle Enterprise Repository Compliance Templates Guide*.

Automatic Usage Detection

Oracle Enterprise Repository can automatically detect asset reuse within the development environment. This allows development teams to ensure that they get asset reuse credit, regardless of whether the assets have been downloaded through Oracle Enterprise Repository or pulled from another source, such as the developer's desktop. Automated Usage Detection relies on a fingerprinting process, called Software File Identification (SFID), which tags selected files within an asset with a unique ID. This SFID is then used to detect when and where an asset is used, even if the asset was acquired through means other than the Oracle Enterprise Repository Use - Download process. An instance of usage is recorded by Oracle Enterprise Repository when tagged files within the asset are brought into the Oracle Enterprise Repository plug-in for Eclipse, and a new build or build clean occurs. For more information, see *Oracle Enterprise Repository Software File Identification Guide*.

Using VS .NET

The Oracle Enterprise Repository plug-in for VS .NET development environment use cases are as follows:

Submit Files

The Oracle Enterprise Repository plug-in for VS .NET allows you to select files to submit to the Oracle Enterprise Repository. It packages the files into a .zip format for archive submission. The Archive Submission Wizard allows you to submit single and/or compound-payload assets to Oracle Enterprise Repository via an archive ZIP file.

Harvest Artifacts

Oracle Enterprise Repository can harvest BPEL, WSDL, XSD, and XSLT files and file directories. After harvesting, Oracle Enterprise Repository automatically creates assets, populates asset metadata, and generates relationship links based on the information in the artifact files. The harvesting function is available from the command line, and can be integrated into VS .NET or into the build process.

Search Oracle Enterprise Repository

You can access the assets and artifacts available in the Oracle Enterprise Repository through the Oracle Enterprise Repository plug-in for VS .NET. Through VS .NET, you can search for assets matching various criteria or view assets that may be of interest to a development project.

View Asset Details

For selected assets, you can view asset details such as description, usage history, expected savings, and relationships. Within the asset metadata, links to the supporting documentation, user guides, test cases are provided to better enable you to reuse the existing functionality.

Download Artifacts

You can download an asset's artifacts (i.e., payload) into your project. Typically an asset payload is the functionality that you need to use a service (such as a WSDL file) or incorporate into your code base (usually a binary).

Prescriptive Reuse

Through the Oracle Enterprise Repository, analysts, architects, technical leads, and others who are involved in the design stages of a project can create a list of assets that fulfills a project's requirements. The list of assets are captured in compliance templates in the repository and the compliance templates are associated with an Oracle Enterprise Repository project.

From within the Oracle Enterprise Repository plug-in for VS .NET, you can view a list of assets appearing in all of the Compliance Templates assigned to your project. You can see which of the assets have been used and/or other project members. For more information on compliance templates, see *Oracle Enterprise Repository Compliance Templates Guide*.

Automatic Usage Detection

Oracle Enterprise Repository can automatically detect asset reuse within the development environment. This allows development teams to ensure that they get asset reuse credit, regardless of whether the assets have been downloaded through Oracle Enterprise Repository or pulled from another source, such as the developer's desktop. Automated Usage Detection relies on a fingerprinting process, called Software File Identification (SFID), which tags selected files within an asset with a unique ID. This SFID is then used to detect when and where an asset is used, even if the asset was acquired through means other than the Oracle Enterprise Repository Use - Download process. An instance of usage is recorded by Oracle Enterprise Repository when tagged files within the asset are brought into the Oracle Enterprise Repository plug-in for VS .NET, and a new build or build clean occurs. For more information, see *Oracle Enterprise Repository Software File Identification Guide*.