

# Oracle® Enterprise Repository

Harvester User Guide

10g Release 3 (10.3)

July 2009

ORACLE®

Copyright © 2008, 2009, Oracle and/or its affiliates. All rights reserved.

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

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

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

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

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

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

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

# Contents

1.	What is the Harvester? .....	1-2
	Prerequisites .....	1-2
	High Level Use Cases .....	1-2
	Related Documentation .....	1-3
2.	Obtaining the Harvester .....	2-2
	Configuring the Harvester for the Command Line .....	2-2
	Setting Repository Connection Information for the Command Line .....	2-2
	Selecting the Artifacts to Harvest for the Command Line .....	2-3
	Configuring Harvester for Oracle JDeveloper .....	2-4
	Configure Oracle JDeveloper to Integrate the Harvester .....	2-5
	Setting Repository Connection Information for Oracle JDeveloper .....	2-6
	Selecting the Artifacts to Harvest for Oracle JDeveloper .....	2-7
	Invoking the Harvester Using the Repository.Submit Ant Task .....	2-8
	Specifying Parameters for the repository.submit Ant Task .....	2-8
	Specifying Parameters as Nested Elements .....	2-10
	Running the Harvester from Ant .....	2-11
	Performing Optional Harvester Configuration .....	2-11
	Configuring Logging for the Harvester .....	2-12
3.	Harvesting Artifacts .....	3-2
	BPEL .....	3-3
	WSDL .....	3-4

XSD.....	3-4
XSL.....	3-4
Deployment Information.....	3-5
Detecting Duplicate Artifacts.....	3-6
Downloading Artifacts.....	3-6
Tutorial for Oracle JDeveloper.....	3-8
Searching Harvested Metadata.....	3-9
Best Practices.....	3-11
Recommended Privileges for Harvesting.....	3-11
Use a Unique Namespace for Each Unique Interface, Service, and Endpoint...	3-11
Harvest Completed Work.....	3-12
Harvesting and Maintenance Releases of XSD.....	3-12
Known Issues.....	3-12
Asset Types Must be Present in the System.....	3-12
Two Versions of an Asset Type.....	3-12
Do Not Delete the Harvester-Specific Metadata Entries in a in a Harvested Asset	3-13

# Getting Started with the Harvester

This chapter contains information on the following subjects:

- [“What is the Harvester?”](#) on page 1-2
- [“Prerequisites”](#) on page 1-2
- [“High Level Use Cases”](#) on page 1-2
- [“Related Documentation”](#) on page 1-3

## What is the Harvester?

The Harvester harvests Oracle SOA Suite artifacts, including BPEL, WSDL, XSD and XSLT files and file directories, and 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 Harvester can be used from the command line, it can be integrated with Oracle JDeveloper and then run within Oracle JDeveloper, or it can be invoked using an Ant task.

## Prerequisites

Before using the Harvester, users must import the SOA/BPM Suite Solution Pack, which includes the required Asset Types and Relationships, into Oracle Enterprise Repository. Import the `BEA_HOME/Tools/Solutions/OER103-SOA-BPM-Suite-Solution-Pack.zip` file from the Oracle Enterprise Repository installation home. See the [Oracle Enterprise Repository Import/Export Guide](#) for information about importing.

## High Level Use Cases

You can use the Harvester to:

- Publish standard SOA artifacts such as BPEL (from Oracle and other vendors), WSDL, XSD, and XSLT to Oracle Enterprise Repository.
- Capture dependencies between the artifacts for impact analysis.
- Handle nested WSDLs and XSDs.
- Create abstract Interface assets and concrete deployment / Endpoints.
- Capture BPEL partnerlink dependencies and relating them to Interfaces.
- Capture BPEL PM dependencies to Transformations.
- Store artifact content in OER for reusability.
- Calculate Software File IDs (SFIDs) for artifacts for duplicate detection.
- Publish artifacts from nested directories or zip files, jar files, or a BPEL suitcase.
- Publish artifacts in a transactional fashion.

- Publish artifacts from command line, and from automated Ant build process using the Harvester Ant task.

## 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>

## Getting Started with the Harvester



# Configuring the Harvester

This chapter contains information on the following subjects:

- [“Obtaining the Harvester” on page 2-2](#)
- [“Configuring the Harvester for the Command Line” on page 2-2](#)
- [“Configuring Harvester for Oracle JDeveloper” on page 2-4](#)
- [“Invoking the Harvester Using the Repository.Submit Ant Task” on page 2-8](#)
- [“Performing Optional Harvester Configuration” on page 2-11](#)
- [“Configuring Logging for the Harvester” on page 2-12](#)

## Obtaining the Harvester

The Harvester is available in the Harvester Solution Pack, which is bundled with the Oracle Enterprise Repository 10.3 installation in the following .zip file:

```
<BEA_HOME>/repository103/core/tools/solutions/OER103-SOA-BPM-Harvester.zip
```

This manual refers to the directory that you extract the OER103-SOA-BPM-Harvester.zip file to as the `<Harvester Home>` directory.

If you plan to configure Oracle JDeveloper to integrate the Harvester, read [“Configuring Harvester for Oracle JDeveloper” on page 2-4](#) before extracting the .zip file to a directory.

## Configuring the Harvester for the Command Line

To configure the Harvester for the command line, you must:

1. Set the repository connection information. See [“Setting Repository Connection Information for the Command Line” on page 2-2](#) for instructions.
2. Select the artifacts to be harvested. See [“Selecting the Artifacts to Harvest for the Command Line” on page 2-3](#) for instructions.

### Setting Repository Connection Information for the Command Line

Open the XML file `IntrospectorSettings.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/aler</uri>
  <credentials>
    <user>smith</user>
    <password>oeruser</password>
  </credentials>
  <timeout>30000</timeout>
</repository>
```

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

```
C:\test\harvester> introspect -url http://localhost:7101/aler -user admin
-password admin -dir c:\test\samples
```

## Selecting the Artifacts to Harvest for the Command Line

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

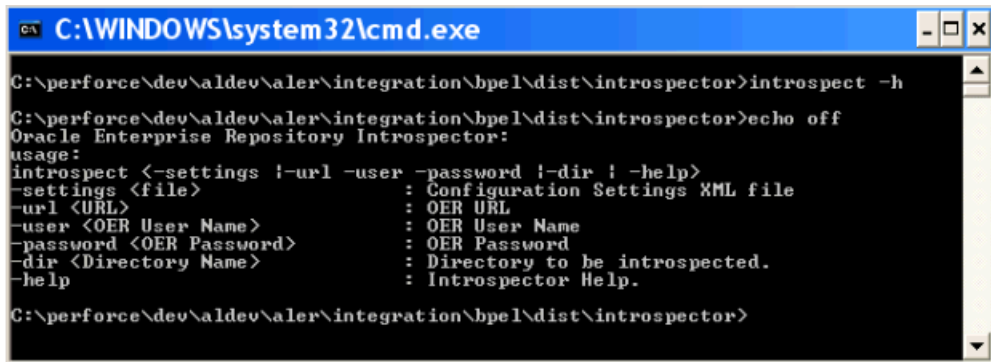
[Table 2-1](#) shows the options that can be specified using the Harvester command line utility:

**Table 2-1 Command Line Options for the Harvester**

Harvester Options	Description
-settings <file>	Advises the Harvester to use the configuration settings specified in the XML file, IntrospectorSettings.xml
-URL <URL>	Specifies the URL for the Oracle Enterprise Repository instance
-user <OER User Name>	The name of the Oracle Enterprise Repository user.
-password <OER Password>	The password of the Oracle Enterprise Repository user.
-dir <directory name>	Specifies the directory to be harvested.
-help	Displays the online help for the Harvester command line utility.

[Figure 2-1](#) shows the command line utility options and online help displayed by the introspect -h command.

Figure 2-1 Introspect Command Line Utility Options



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

If the options are omitted, the Harvester uses the information in the IntrospectorSettings.xml file in the <Harvester Home> directory, where introspect.bat resides. If options are specified on the command line, they override the settings in IntrospectorSettings.xml.

To point to the artifacts to be harvested using the IntrospectorSettings.xml file in the <Harvester Home>, modify the following XML:

```
<query>
  <fileQuery>
    <rootDir>C:\samples</rootDir>
    <files>BPEL</files>
  </fileQuery>
</query>
```

## Configuring Harvester for Oracle JDeveloper

To configure the Harvester for Oracle JDeveloper, you must:

1. Configure Oracle JDeveloper to integrate the Harvester. See [“Configure Oracle JDeveloper to Integrate the Harvester”](#) on page 2-5 for instructions.
2. Set the repository connection information. See [“Setting Repository Connection Information for Oracle JDeveloper”](#) on page 2-6 for instructions.

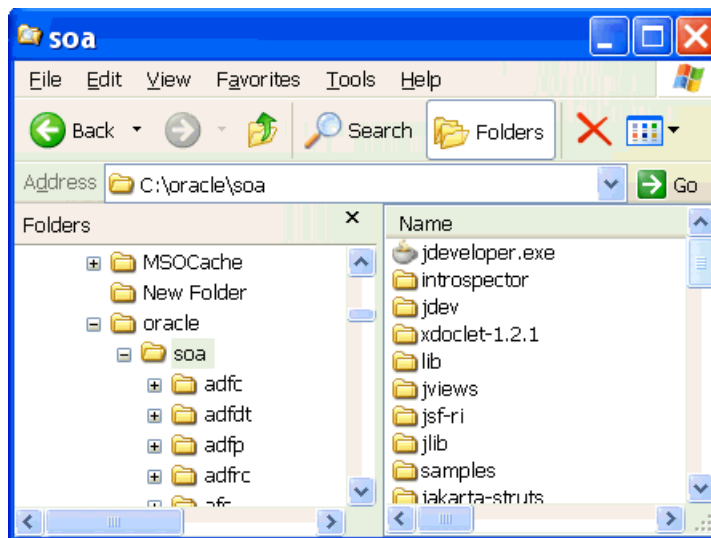
3. Select the artifacts to be harvested. See [“Selecting the Artifacts to Harvest for Oracle JDeveloper”](#) on page 2-7 for instructions.

## Configure Oracle JDeveloper to Integrate the Harvester

Follow these steps to configure Oracle JDeveloper to integrate the Harvester:

1. Unzip the Harvester (which can be found in the OER103-SOA-BPM-Harvester.zip file in the `BEA_HOME/repository103/core/tools/solutions` directory) to the Oracle JDeveloper directory. For example, if the `jdeveloper.exe` file is located in `C:\oracle\soa`, make sure that the introspector directory is unzipped into that directory, as shown in [Figure 2-2](#):

**Figure 2-2 Unzip the Introspector Directory to the Directory Where `jdeveloper.exe` is Located**

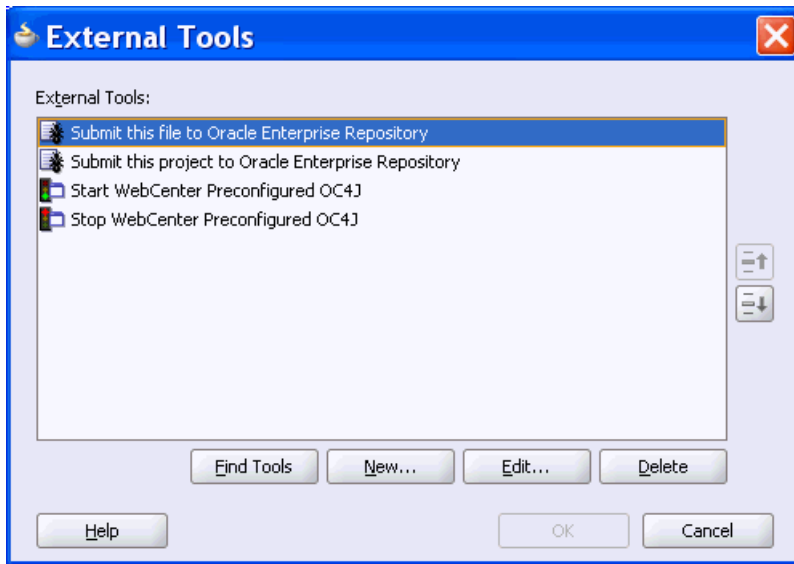


2. Open the `tools.xml` file in `<jdeveloper_home>\jdev\system\oracle.jdeveloper.10.1.xxxxx` in a text editor.
3. Open the `tools.xml` file in `<jdeveloper_home>\introspector` in a text editor. Copy all the elements between the `<tools>` and `</tools>` elements.
4. Paste the copied elements from the previous step into the `tools.xml` file in the `<jdeveloper_home>\jdev\system\oracle.jdeveloper.10.1.xxxxx` directory.

## Configuring the Harvester

5. Save the `tools.xml` file in the `<jdeveloper_home>\jdev\system\oracle.jdeveloper.10.1.xxxxx` directory.
6. Launch Oracle JDeveloper and click **Tools > External Tools**. The Harvester menu items shown in [Figure 2-3](#) display:

**Figure 2-3** Viewing the Harvester Menu Items in the JDeveloper External Tools Dialog Box



7. To edit a Harvester item, select it and click **Edit**. In the Edit External Tool dialog box, edit the `registry.url`, `registry.username`, and `registry.password` properties to set the Oracle Enterprise Repository URL, username, and password.
8. To submit an item in the External Tools dialog box to Oracle Enterprise Repository, select it and click **OK**.

## Setting Repository Connection Information for Oracle JDeveloper

Note that you do not need to perform the following steps if you performed them in the previous section.

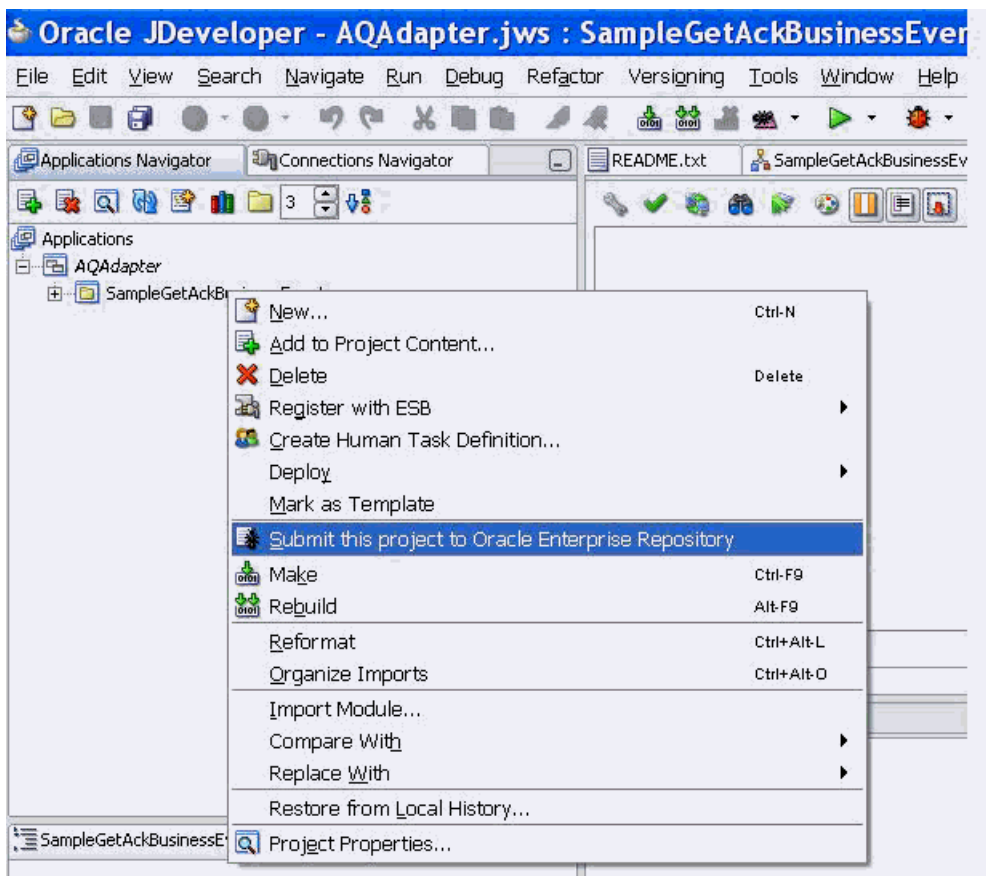
1. Launch Oracle JDeveloper and choose **Tools > External Tools**.

2. Select **Submit this file to Oracle Enterprise Repository** and click **Edit**.
3. On the Properties tab, edit `registry.url`, `registry.username`, and `registry.password`.

## Selecting the Artifacts to Harvest for Oracle JDeveloper

After Oracle JDeveloper has been configured to integrate the Harvester and the correct repository connection information has been specified, a user can right-click a project and choose **Submit this project to Oracle Enterprise Repository** from the context menu, as shown in [Figure 2-4](#).

**Figure 2-4** 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.

## Invoking the Harvester Using the Repository.Submit Ant Task

You can use the repository.submit Ant task provided with the Harvester to harvest and import BPEL files into Oracle Enterprise Repository. This task can be defined in the introspect-tasks.xml file, which is located in the <Harvester Home> directory (the directory into which you unzipped the Harvester Solution Pack).

### Specifying Parameters for the repository.submit Ant Task

Table 2-2 shows parameters that can be specified for the repository.submit Ant task in the introspect-tasks.xml file:

**Table 2-2 Parameters for the repository.submit Ant Task**

Attribute	Description	Required
repositoryURL	Registry 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	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."



**Table 2-2 Parameters for the repository.submit Ant Task**

<b>Attribute</b>	<b>Description</b>	<b>Required</b>
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 will be set to "true". Otherwise, the specified property will remain unset.	No.
debug	Display debug information from the Oracle Enterprise Repository task regardless of the Ant "-debug" setting. Debug information will also be displayed if you pass the -debug parameter to the Ant runtime.	No. Defaults to "false."
description	Introspection Description to associate with each asset created in Oracle Enterprise Repository. This is visible in the "Introspection Properties."	No.
namespace	Namespace with which to prefix each asset created in Oracle Enterprise Repository.	No.
version	Introspection Version to associate with each asset created in Oracle Enterprise Repository. This is visible in the "Introspection Properties."	No.
jdeveloper	Whether or not the specified fileset is inside an Oracle JDeveloper workspace. If so, the Harvester will automatically search the "output" directories for deployment information.	No. Defaults to "false."

**Table 2-2 Parameters for the repository.submit Ant Task**

Attribute	Description	Required
settingsFile	Location of settings XML file. Must conform to BPEL_Introspector_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 will be bundled with the Ant task.
settingsURL	Location of settings XML file. Must conform to BPEL_Introspector_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 will be bundled with the Ant task.

## Specifying Parameters as Nested Elements

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 [Listing 2-1](#), the Harvester examines all the files and directories under the fileSet directory (the /tmp/components directory) and imports them into Oracle Enterprise Repository.

### Listing 2-1 Specifying the Files to Harvest with the Ant repository.submit Task

```
<repository.submit repositoryurl="http://server.example.com:8080/oer"
  repositoryusername="myuser"
  registrypassword="mypassword"
  settingsFile=" ../MyCustomSettings.xml">

  <fileset dir="/tmp/components/">
    <include name="**/*" />
  </fileset>
```

```
</repository.submit>
```

---

## Running the Harvester from Ant

To import the Harvester Ant task, include a line like this in your Ant XML:

```
<taskdef file="{harvester.dir}/introspect-tasks.xml" />
```

where `harvester.dir` is the `<harvester home>` directory to which you unzipped the Harvester Solution Pack.

When running from the command line, make sure that the Harvester libraries are available to Ant by using Ant's `-lib` command line switch. For example:

```
ant -lib <harvester home> -lib <harvester home>/lib -f mybuild.xml
```

where `<harvester home>` is the directory where you unzipped the Harvester Solution Pack.

## Performing Optional Harvester Configuration

You can optionally modify these additional configuration settings in the XML file `IntrospectorSettings.xml` in the `<Harvester Home>` directory:

- `<introspectionDescription>`: A description about the harvesting performed. This information will be stored in the "Introspection Properties" of the assets created in Oracle Enterprise Repository.
- `<introspectionVersion>`: Version of the harvesting performed. This information will be stored in the "Introspection Properties" of the assets created in Oracle Enterprise Repository.
- `<namespace>`: Users can supply a namespace in the configuration when the harvesting is started. The scope of the namespace is that it applies to *only* the non artifact asset and not to artifact based assets.
- `<workDir>`: This is the temporary directory where the zip and jar files will be unzipped. By default, the system temp directory is used.

## 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 `log4j.properties` file located in the <Harvester Home> directory.

# Harvesting of Artifacts

This chapter contains information on the following subjects:

- [“Harvesting Artifacts”](#) on page 3-2
- [“Detecting Duplicate Artifacts”](#) on page 3-6
- [“Downloading Artifacts”](#) on page 3-6
- [“Tutorial for Oracle JDeveloper”](#) on page 3-8
- [“Searching Harvested Metadata”](#) on page 3-9
- [“Best Practices”](#) on page 3-11
- [“Known Issues”](#) on page 3-12

## Harvesting Artifacts

An Oracle BPEL PM project can be submitted to Oracle Enterprise Repository either from the command line, from Oracle JDeveloper, or using an Ant task.

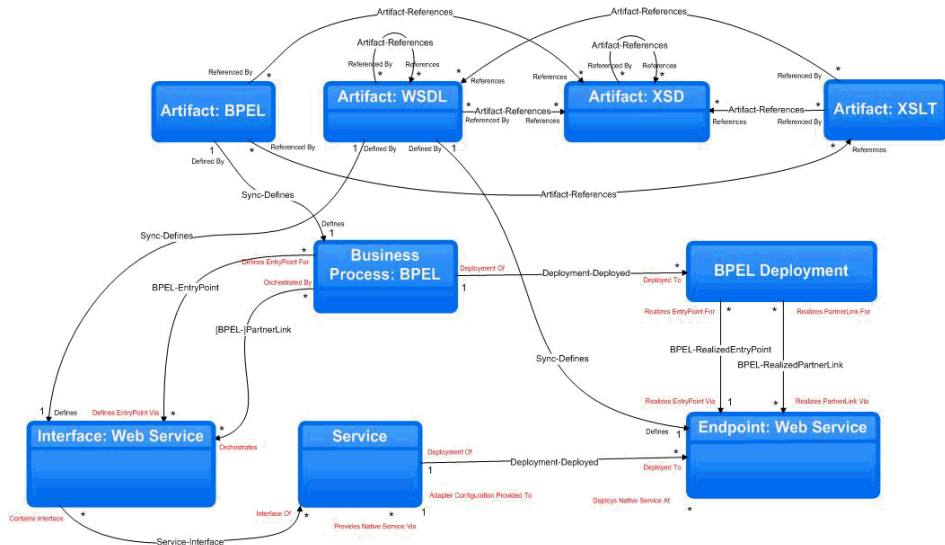
The Harvester scans for artifacts such as the following and harvests those artifacts to detect the dependencies that exist between them:

- BPEL: See [“BPEL” on page 3-3](#) for more information about how Oracle Enterprise Repository deals with submitted BPEL artifacts.
- WSDL: See [“WSDL” on page 3-4](#) for more information about how Oracle Enterprise Repository deals with submitted WSDL artifacts.
- XSD: See [“XSD” on page 3-4](#) for more information about how Oracle Enterprise Repository deals with submitted XSD artifacts.
- XSL: See [“XSL” on page 3-4](#) for more information about how Oracle Enterprise Repository deals with submitted XSL artifacts.
- Deployment information: See [“Deployment Information” on page 3-5](#) for more information about the concrete information the Harvester looks for when an Oracle BPEL PM project is submitted to Oracle Enterprise Repository.

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

[Figure 3-1](#) shows the asset types created by the Harvester and the relationships between them.

Figure 3-1 Asset Types Created by the Harvester



## 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”) will be created that will have metadata about the operations invoked by this BPEL definition.
- A BPEL artifact asset (of type "Artifact: BPEL") will be created that will contain the BPEL artifact contents.
- The Business Process will be 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: Webservice").
- The BPEL artifact asset will be related to WSDL and XSLT artifact (if a transformation is performed in the flow) assets. These are the WSDL that will contain the definitions of the partner links and the entry points for the Business Process.

## 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”) will be created.
- An interface asset (of type: “interface: Webservices”) will be created for the port type.
- The Service will be related to the Interface asset using the “Contains interface” relationship.
- An Endpoint asset will be created for the port.
- The Service asset will be related to the Endpoint asset using the “Deployed at” relationship.
- WSDL artifact asset (of type "Artifact: WSDL") will be created that will contain the WSDL artifact contents.
- If the WSDL artifact imports WSDLs and imports / includes XSDs, it will be related to those WSDL and XSD artifact assets using the “References” relationship.

## 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") will be created that will contain the XSD artifact contents.
- If the XSD artifact imports / includes XSDs, it will be related to those XSD artifact assets using the “References” relationship.

## 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") will be created that will contain the XSL artifact contents.



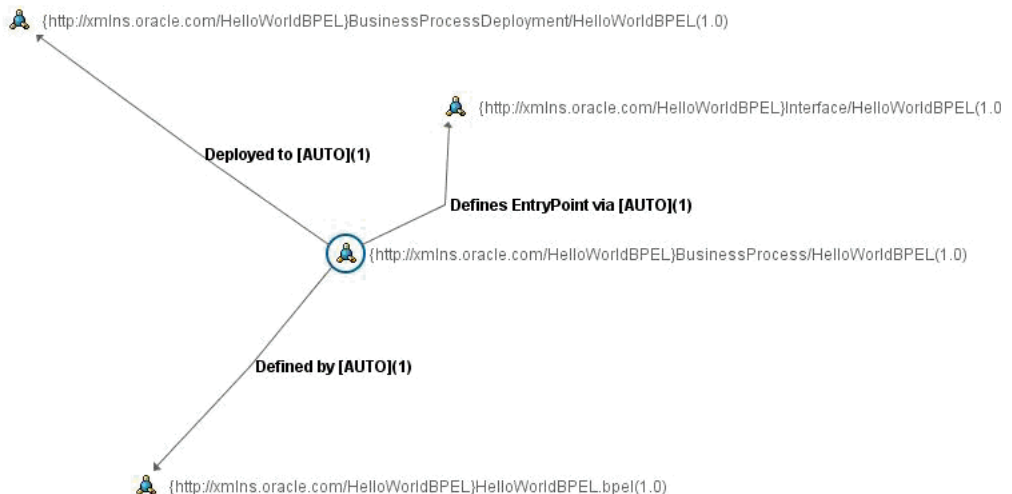
- If the XSL artifact references XSDs and WSDLs as source and target for the transformation, it will be related to those XSD and WSDL artifact assets using the “References” relationship.

## Deployment Information

When an Oracle BPEL PM project is submitted to Oracle Enterprise Repository, the Harvester will look for the concrete information for the following:

- A BPEL process is exposed as a Service. When a BPEL PM Project is deployed within the JDeveloper or from the command line using ANT framework, a property file containing the host, port, and revision of the BPEL definition is used by the ANT deployment. This file is harvested by the Harvester so that the concrete information (such as the information shown in [Figure 3-2](#)) will be updated in Oracle Enterprise Repository.

**Figure 3-2 Oracle Enterprise Repository Concrete Information for a BPEL Process**



Oracle BPEL PM uses the following format to construct the concrete WSDL URI:

```
<host> : <port> /orabpel /<BPELDomain> /<processName> /<version> /<ServiceName>
>?wsdl
```

The Harvester constructs the URI using the values from the property file. An example of a constructed concrete WSDL URI is:

```
http://localhost:8888/orabpel/default/OrderBooking/1.7/OrderBooking?wsd  
1
```

- Partner links are bound to concrete binding that are found in `bpel.xml`. This file is harvested so that the concrete information about where the dependent services are running is updated in Oracle Enterprise Repository.

## Detecting Duplicate Artifacts

The Harvester store files such as WSDLs, BPELs, and XSDs as artifacts in Oracle Enterprise Repository. To avoid storing the same artifact file twice, the Harvester will calculate a Software File ID ("SFID") for each artifact when it is stored. Before submitting a new artifact, the SFID can be compared against existing SFIDs in the repository to check for duplicates.

The SFID calculated is an MD5 hash. Some level of canonicalization is performed before calculating the SFID. In particular, if the artifact file is XML, it is canonicalized using the Canonicalizer class in the Apache XML Security library. This canonicalizes according to the W3C "Canonical XML" standard (see [www.w3.org/TR/xml-c14n](http://www.w3.org/TR/xml-c14n)), which includes canonicalizing the text encoding, line breaks, whitespace, comments, and attribute ordering. Some extra canonicalization not specified in the W3C standard is performed, including normalizing of namespace prefixes, normalizing the order of the elements in WSDLs, removing documentation elements, and inlining any included/imported files.

## Downloading Artifacts

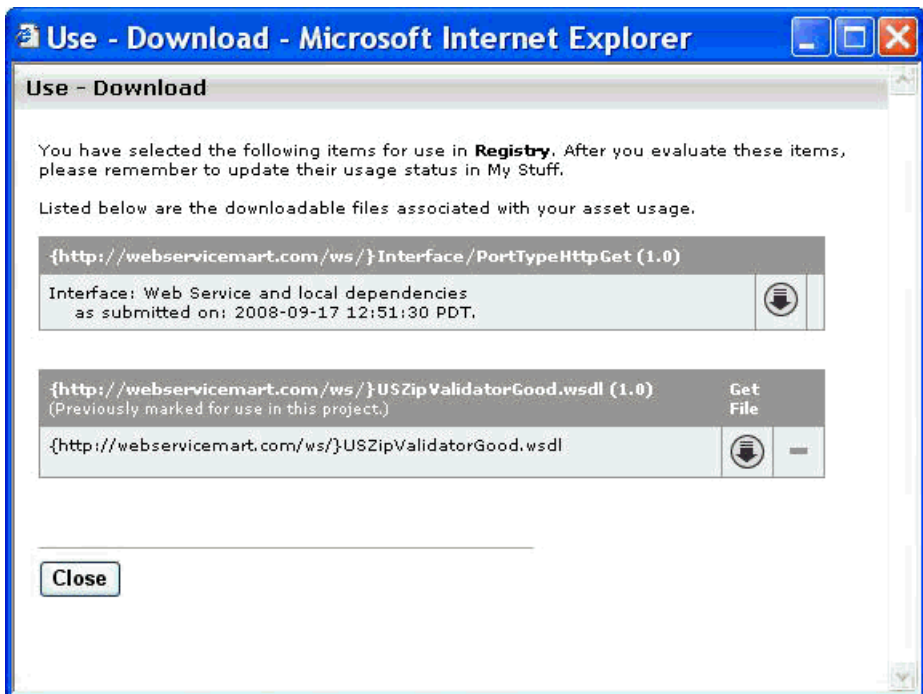
The Harvester creates artifact bundles that may be downloaded from Business Processes, Interfaces, and Endpoints. The artifact bundles for these assets are stored in zip files. For example, for an Endpoint, a WSDL file and its associated XSD files are stored in relative locations within the zip payload.

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.

After the Harvester runs, you can download the asset bundle by following these steps:

1. In the Oracle Enterprise Repository web console, search for any of the assets created by the harvesting. Choose **Interface: Webservice** in the **Type** field, optionally enter a search string, and click **Search**.
2. In the Search Results pane, select the asset you are interested in. Then in the asset detail pane, click the **Use - Download** button. This displays the Use - Download page in a separate window.
3. On the Use - Download page, choose the project to which you want to extract (download) the asset, then click **Next**.
4. On the Use - Download page, the artifact bundle(s) are shown, if any. See [Figure 3-3](#) for an example of the Download page. Click the **Download** link to save the artifact bundle in a zip file.

**Figure 3-3** Downloading Artifact Bundles from the Use - Download Page



**Note:** If the artifact bundle has only a single artifact as payload, then you will be provided the payload directly, instead of the payload being delivered within a zip file.

If you harvest a series of files, change some of the files, and then harvest the bundles again, multiple payloads (ordered by harvesting date, with the most recent first) will be available for download on the Download page.

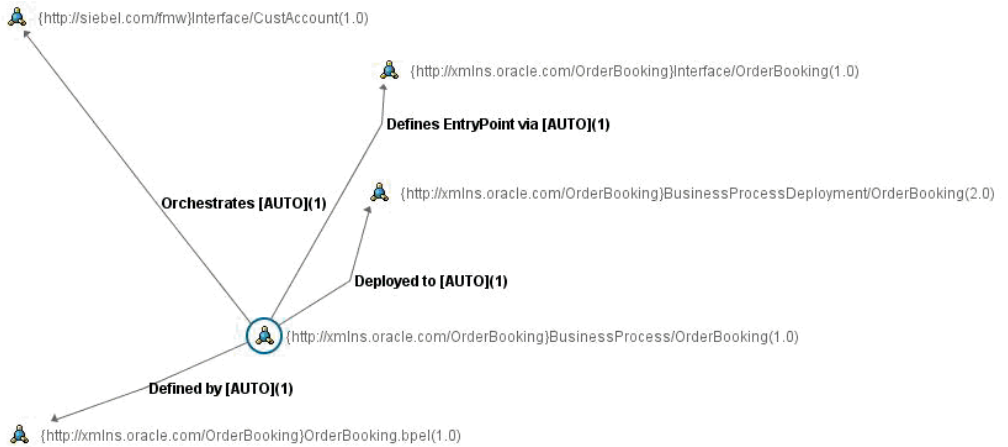
## Tutorial for Oracle JDeveloper

The following steps show how to create a sample BPEL project in Oracle JDeveloper, deploy the project to the BPEL Engine, and submit the project to Oracle Enterprise Repository along with the artifact dependencies and deployment information.

1. In Oracle JDeveloper, right-click an application and select **New Project** to create a new BPEL Process Project.
2. Select **BPEL Process Project** and click **OK**.
3. Enter “TestBPELProject” and click **Finish**.
4. Right-click **TestBPELProject** and choose **Rebuild**.
5. If you do not need concrete deployment information to be harvested, skip steps 6 through 8 and continue with step 9.
6. Under **TestBPELProject/Resources**, double-click **build.properties** to configure the BPEL container.
7. Uncomment the domain, rev, admin.user, admin.password, http.hostname, http.port properties and provide the correct values for them.
8. Right-click build.xml and choose **Run Ant Target > deploy**. This deploys the BPEL project just created.
9. Right-click **TestBPELProject** and choose **Submit this project to Oracle Enterprise Repository**. This submits the project to Oracle Enterprise Repository, along with deployment information and information about dependencies between the artifacts.
10. Browse to `http://localhost:7101/aler/` in your web browser. On the left side of the console, choose **Business Process** in the **Type** field and **All Assets** in the **Registration Status** field and then click **Search**.
11. In the returned results, you will see that the following asset has been updated with the deployment information, as shown in [Figure 3-4](#):

```
{http://xmlns.oracle.com/TestBPELProject}BusinessProcess/TestBPELProject
```

**Figure 3-4 Viewing Updated Deployment Information for an Asset**

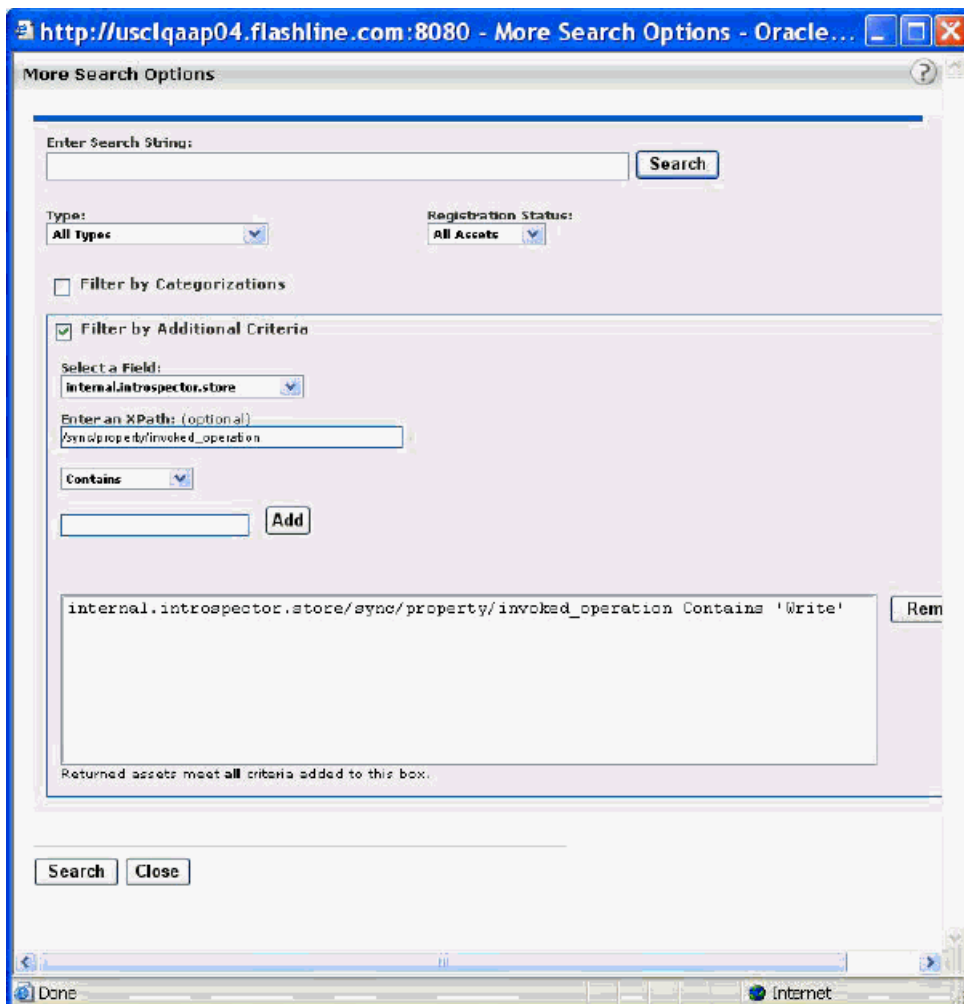


## Searching Harvested Metadata

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

Figure 3-5 shows how to query for Business Processes that invoke the operation “Write.” To get the search screen, click **More Search Options** on the main page of the Oracle Enterprise Repository Web console.

Figure 3-5 Viewing the More Search Options Dialog Box in Oracle Enterprise Repository



The following search criteria are available:

- Introspection Description
- Introspection Version
- Introspection Namespace

- Introspected by
- Invoked Operations of Business Processes

## Best Practices

This section describes best practices for the Harvester.

### 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.

### Use a Unique Namespace for Each Unique Interface, Service, and Endpoint

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 3-1](#) shows the correlation of Oracle Enterprise Repository assets with WSDL structure:

**Table 3-1 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

## Harvest 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.

## 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, etc. 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.

## Known Issues

This section describes some known Harvester issues:

### 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 the Harvester Solution Pack.

### 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 the Harvester Solution Pack, 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.



## **Do Not Delete the Harvester-Specific Metadata Entries 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.introspector.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.

## Harvesting of Artifacts