Oracle® Fusion Middleware

Upgrade Guide for Oracle Enterprise Repository

11*g* Release 1 (11.1.1.6.3)

E15746-12

August 2012

Provides information and instructions for upgrading and troubleshooting the upgrade process for Oracle Enterprise Repository.

ORACLE®

Oracle Fusion Middleware Upgrade Guide for Oracle Enterprise Repository, 11*g* Release 1 (11.1.1.6.3)

E15746-12

Contents

1   Upgrading Oracle Enterprise Repository

2 Post Installation Tasks

3 Data Migration Tool

Index

Preface

Welcome to *Oracle Fusion Middleware Upgrade Guide for Oracle Enterprise Repository*. This document provides information and instructions for upgrading and troubleshooting the upgrade process for Oracle Enterprise Repository.

Audience

This document is intended for Oracle Enterprise Repository system administrators who are responsible for installing, maintaining, and upgrading Oracle Enterprise Repository instances.

Documentation Accessibility

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

Access to Oracle Support

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

Related Documents

For more information, see the following documents in the Oracle Enterprise Repository 11*g* Release 1 (11.1.1.6.3) documentation set:

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

  `http://www.oracle.com/technetwork/middleware/repository/overv`
  `iew/index-100687.html`

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

  `http://blogs.oracle.com/governance`

Conventions

The following text conventions are used in this document:

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

What's New in This Guide for Release 11.1.1.6.x

This guide was updated for 11*g* Release 1 (11.1.1.6.3). The following table lists sections that are new or changed.

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

Sections	Changes Made	11.1.1.6.0	11.1.1.6.3
All Chapters			
Chapter summaries	Enhanced document and chapter summaries for better web search results.	X	
Chapter 3			
Chapter 3.1.1, "Best Practices"	Updated this section to reflect changes to the Data Migration Tool.	X	
Chapter 3.1.4, "Migration Tool Functionality"	Section revised to include updates to the Data Migraton Tool.	X	

x

1

Upgrading Oracle Enterprise Repository

This chapter describes the upgrade process tasks and requirements when upgrading from Oracle Enterprise Repository 10g Release 3 (10.3) and ALER 3.x versions. Use the following links for additional instructions specific to your application server.

This chapter contains the following sections:

- Section 1.1, "Upgrade from Oracle Enterprise Repository 10g R3 to 11g R1"
- Section 1.2, "Upgrade Oracle Enterprise Repository Workflows"

1.1 Upgrade from Oracle Enterprise Repository 10*g* R3 to 11*g* R1

You can upgrade from Oracle Enterprise Repository 10.3 to Oracle Enterprise Repository 11*g* R1.

This section contains the following topics:

- Section 1.1.1, "Upgrade Process"
- Section 1.1.2, "Tomcat Application Upgrade"
- Section 1.1.3, "WebLogic Server Application Upgrade"
- Section 1.1.4, "WebSphere Application Upgrade"
- Section 1.1.5, "Migrating Your Custom Assets"

1.1.1 Upgrade Process

To upgrade from Oracle Enterprise Repository 10.3 to Oracle Enterprise Repository 11*g*, perform the following steps:

1. Back up the existing Oracle Enterprise Repository database.

2. Encrypt clear text passwords within the existing `database.properties` and `ldap.properties` files by using the Oracle Enterprise Repository diagnostic tool.

    - Also, encrypt the `activemq.xml` and `EndPointEventSubscription.xml` files that contain the password properties and are found at the `oer-app/WEB-INF/classes` directory.

    - If you are upgrading from ALER 3.x, then change the `<sub:uri>` element in the `EndPointEventSubscription.xml` file, which represents the service URI for the supported server, to the following by uncommenting the following line:

```
<sub:uri>albpmServices/aler_
engine/ws/StatusChangeEndpointServiceListener</sub:uri>
```

and by providing a comment the following line:

```
<sub:uri>fuegoServices/ws/StatusChangeEndpointServiceListe
ner</sub:uri>
```

- Encrypt the passwords in the `workflow.xml`, which can be found at *<Oracle_ Home>\obpm\enterprise\server\aler_engine*, and then back up this file.

3. Back up the existing Oracle Enterprise Repository application deployment:

- It is recommended that you back up the entire `<BEA_HOME>`. This backup is necessary for some post-installation tasks, in addition to the application server upgrade.

- For clustered deployments, each node of the cluster should be separately backed up as all nodes must have Oracle Enterprise Repository upgraded. For more information, see *Oracle Fusion Middleware Installation Guide for Oracle Enterprise Repository*.

4. Verify required platforms and version information (including JDBC Driver version support), using Table 1–1, which describes the supported JDBC Driver file names for Oracle Enterprise Repository 11*g* and, also its mapping to the appropriate application server.

Table 1–1   Oracle Enterprise Repository 11g Supported JDBC Driver Information

App Server JDK Version	Oracle JDBC	Oracle Driver Version	UDB JDBC	UDB Driver Version	SQL Server JDBC	SQL Driver Version
1.5.x	ojdbc5.jar	11.1.0.7.0-Production	db2jcc.jar db2jcc_license_cu.jar db2jcc4.jar	3.50.152	sqljdbc4.jar	2.0.1803.100
1.6.x	ojdbc6.jar	11.1.0.7.0-Production	db2jcc.jar db2jcc_license_cu.jar db2jcc4.jar	3.50.152	sqljdbc4.jar	2.0.1803.100

> **Note:**   A patch is required for AIX, which would be available in `tools/solutions` as a zip file that you can drop on the server.

5. Uninstall Oracle Enterprise Repository 10.3.

6. Run the Oracle Enterprise Repository 11*g* product installer.

> **Note:**   Run the database configuration assistant at the end of the Oracle Enterprise Repository installer process.

For more information about running the product installer, see *Oracle Fusion Middleware Installation Guide for Oracle Enterprise Repository*.

7. Restore the Oracle Enterprise Repository backup files from the previous Oracle Enterprise Repository deployment to the newly installed Oracle Enterprise Repository 11*g* version, including:

- The Oracle Enterprise Repository property files

If you are using LDAP, then confirm that the Enable LDAP Login option is set to True and verify the LDAP Configuration settings. If you are using Oracle Application Server Single Sign-On (OracleAS Single Sign-On), then confirm that the Enable Advanced Container Login option is set to True and verify the Advanced Container Configuration settings. If your LDAP or OracleAS Single Sign-On configurations are not set correctly, then you will not be able to authenticate with the Oracle Enterprise Repository application.

The property value for the LDAP authentication feature is:

```
enterprise.authentication.ldap.enabled=true
```

For Container Managed Authentication:

```
enterprise.authentication.container.enabled=true
enterprise.authentication.container.synchroles.enabled=true/false
(depending on customer need)
```

For OracleAS Single Sign-On:

```
enterprise.authentication.advancedcontainer.enabled=true
```

These properties should be enabled (as appropriate) and saved within the *oer/WEB-INF/classes/enterprise.properties* file.

- Any custom files, such as custom home pages, images, XSLT templates, upload files, and any other customer-related files.

> **Note:**
>
> - For more information about performing encryption for passwords, see Section 2.6.4, "Creating Encrypted Passwords".
>
> - If you have moved settings to database, then see Section 2.3.2, "Move Settings to Database".

8. Start your application server.

9. Verify the application deployment using Oracle Enterprise Repository diagnostics. For more information about enabling the Oracle Enterprise Repository diagnostics tools, see Section 2.6.3, "Enable the Oracle Enterprise Repository Diagnostics Tool".

> **Note:** No database operations are required for this upgrade.

1.1.2 Tomcat Application Upgrade

> **Note:** Tomcat is no longer supported in Oracle Enterprise Repository 11*g* Release 1 (11.1.1.6.3). You may upgrade on Tomcat at your own risk. It is suggested that you move to a supported Application Server as soon as reasonable. For a complete list of supported servers, see *http://www.oracle.com/technology/software/products/ias/files/fusion_certification.html*

These instructions enable you to upgrade from Oracle Enterprise Repository 10.3 to Oracle Enterprise Repository 11*g* on the Tomcat server using the Oracle Products installation program.

This section contains the following topics:

1.1.2.1 Prerequisites

To upgrade from ALER 3.x or Oracle Enterprise Repository 10.3 to Oracle Enterprise Repository 11*g* on the Tomcat server, you must ensure you have the following prerequisites:

- Refer to the password encryption requirements section before performing upgrade steps. For more information, see Section 2.6.4, "Creating Encrypted Passwords".
- Refer to the Supported Configurations list for the most recent information regarding your application server version.
- Ensure that the application server is stopped.
- If the installed application server version is no longer supported by Oracle Enterprise Repository, or the application server has been upgraded to a newer version, then perform the application upgrade instructions for the application server side.
- The directory where Tomcat is installed on your system is referenced in these instructions as CATALINA_HOME.
- The web application context that is being upgraded is referenced in these instructions as CATALINA_HOME/webapps/oer.
- The web content context that is being upgraded is referenced in these instructions as CATALINA_HOME/webapps/oer-web.
- If BPM is in use, then ensure that you back up the BPM Database.
- Back up the existing Oracle Enterprise Repository database

1.1.2.2 Create an Oracle Enterprise Repository Temp Directory

On the Tomcat server, create a directory named oer_temp. This directory contains the backup copies of the current Oracle Enterprise Repository application. This also serves as a temporary location for properties files, JDBC drivers, and any customized Oracle Enterprise Repository items (home.jsp for example) that are required for the new application.

1.1.2.3 Back Up the Current Oracle Enterprise Repository Webapps

Navigate to the Tomcat installation location CATALINA_HOME/webapps and copy the existing oer and oer-web application directories into the oer_temp directory.

1.1.2.4 Uninstall AquaLogic Enterprise Repository 3.x or Oracle Enterprise Repository 10.3

Use the Oracle Enterprise Repository 10.3 Installer to remove all components of the previous release. For more information, see "Uninstalling the Software" in *Oracle Fusion Middleware Installation Guide for Oracle Enterprise Repository*.

1.1.2.5 Run the Oracle Products Installation Program

To run the Oracle Products Installation program, perform the following steps:

1. When prompted for the Tomcat (Catalina) Home directory, supply the directory where Tomcat has been installed.

2. When prompted for the Repository application name, you can use the same application name that the previous deployment had used.

3. When prompted for the Port number, use the same port number that the previous deployment used.

4. If you encounter a "One or more installation directories are not empty. Proceed with installation?" message, select **YES**. You will receive this message only when you re-install the same version.

The installer overwrites the existing deployment with the new files, thus preserving any additional custom files (such as an upload repository directory and the custom home pages.). However, if you select No, the existing deployment is saved to a new context path and the application content is stored in a new directory.

1.1.2.6 Post-Installation Instructions

After the Oracle Enterprise Repository installation is completed, and you have selected to use the Generic Application Server option, then you are left with a *$MIDDLEWARE_HOME/repository111/generic* directory.

You will find the EAR file containing the Oracle Enterprise Repository application within this directory. This file and the contained WAR files have to be unzipped into the *CATALINA_HOME/webapps* directory then renamed to match the previous ALER/OER context name.

1.1.2.7 Restore the Oracle Enterprise Repository Back Up Files

To restore the Oracle Enterprise Repository, back up files from Tomcat to your Oracle Enterprise Repository 11*g* deployment, and perform the following steps:

1. Navigate to the `oer_temp` directory.

2. Edit the *oer_temp/oer/WEB-INF/classes/enterprise.properties* file by adding the following properties to the end of the file:

    ```
    enterprise.authentication.ldap.enabled=true
    enterprise.authentication.container.enabled=true
    enterprise.authentication.advancedcontainer.enabled=true
    ```

3. Copy the `oer_temp/oer/WEB-INF/classes/*.xml` files to the *OER_DEPLOYMENT_DIRECTORY/oer/WEB-INF/classes* directory. This action overwrites the XML files that are a part of the new install.

4. Copy the `oer_temp/oer/WEB-INF/classes/*.properties` files to the *OER_DEPLOYMENT_DIRECTORY/oer/WEB-INF/classes* directory. This action overwrites the properties files that are a part of the new install.

5. Copy the supported JDBC Driver jar files to the *OER_DEPLOYMENT_ DIRECTORY/oer/WEB-INF/lib* directory.

6. Copy the `oer_temp/oer/custom/home.jsp` (and/or all of the custom home pages) to the *OER_DEPLOYMENT_DIRECTORY/oer/custom* directory. This overwrites the `existing home.jsp` file that is part of the new install. In addition, copy any custom images that were included in the custom home page(s) to the appropriate directory.

7. If you are using any custom Type icons, then copy the files from the `oer_ temp/oer-web/images/tree/enterprise/asset` directory to the *OER_ DEPLOYMENT_DIRECTORY/oer-web/images/tree/enterprise/assets* directory.

8. If custom reports were created, then see Section 2.1, "Migrating Custom Reports".

9. Copy/Move the Upload Repository directory (normally found in `oer_ temp/oer/custom/upload`) into the new *OER_DEPLOYMENT_DIR/oer/custom* installation directory.

1.1.2.8 Start Application Server

If you are using LDAP, confirm that the Enable LDAP Login option is set to `True` and verify the LDAP Configuration settings. If you are using OracleAS Single Sign-On, confirm that the Enable Advanced Container Login option is set to `True` and verify the Advanced Container Configuration settings. If your LDAP or OracleAS Single Sign-On configurations are not set correctly, then you will not be able to authenticate with the Oracle Enterprise Repository application.

1. Navigate to the application server's JSP cache directory and delete the cache for the ALER/Oracle Enterprise Repository application.

2. Start the generic application server.

> **Note:** For more information about performing encryption for passwords, see Section 2.6.4, "Creating Encrypted Passwords".

1.1.2.9 Test the Application

You have now deployed the Oracle Enterprise Repository. You should test the installation using the Oracle Enterprise Repository Diagnostics page. You must perform the following tests:

1. On the Oracle Enterprise Repository Diagnostics page

   - Product Version Information

   - Test Required Libraries

   - Test Database

2. If you are unable to access the Oracle Enterprise Repository login page or application, or if you see an error message of some kind, it may be possible that the `db.password` property in the `database.properties` file was not encrypted. Check the **cmee.log** file for any messages that explains this behavior.

For more information about enabling the Oracle Enterprise Repository diagnostics tools, see Section 2.6.3, "Enable the Oracle Enterprise Repository Diagnostics Tool".

1.1.3 WebLogic Server Application Upgrade

These instructions enable you to upgrade from ALER 3.x to Oracle Enterprise Repository 10.3 on WebLogic Server using the Oracle Enterprise Repository 11*g* Release 1 (11.1.1.6.3) installation program.

This section contains the following topics:

- Section 1.1.3.1, "Prerequisites"

- Section 1.1.3.2, "Create Oracle Enterprise Repository Temp Directory"

- Section 1.1.3.3, "Back Up the Current ALER Webapps"

- Section 1.1.3.4, "Remove the Current Oracle Enterprise Repository Installation Files"

- Section 1.1.3.5, "Uninstall ALER 3.x/Oracle Enterprise Repository 10.3"

- Section 1.1.3.6, "Run the Oracle Enterprise Repository Installation Program"

- Section 1.1.3.7, "Post-Installation Instructions"

- Section 1.1.3.8, "Restore the Oracle Enterprise Repository Back Up Files"

- Section 1.1.3.9, "Start WebLogic Server"

- Section 1.1.3.10, "Test the Application"

1.1.3.1 Prerequisites

To upgrade from ALER 3.x or Oracle Enterprise Repository 10.3 to Oracle Enterprise Repository 11*g* on WebLogic Server, you must ensure you have the following prerequisites:

- Oracle Enterprise Repository 11*g* requires WebLogic Server 11*g*. If the currently installed Oracle Enterprise Repository application is not running on the required version, upgrade the application server or install the supported version of WebLogic Server into a separate Oracle Home directory.

- The Oracle Enterprise Repository domain should be installed within the default location of *BEA_HOME/user_projects/domains/applications/<domain_ name>/applications/oer_XXX/oer_app*. If the Oracle Enterprise Repository domain is installed in a different location, you must make the appropriate adjustments to the location.

- Refer to the password encryption requirements section before performing upgrade steps. For more information, see Section 2.6.4, "Creating Encrypted Passwords".

- Stop the WebLogic server before performing these steps.

- If BPM is in use, ensure that you back up the BPM Database.

- Back up the existing Oracle Enterprise Repository database.

1.1.3.2 Create Oracle Enterprise Repository Temp Directory

To create an Oracle Enterprise Repository temp directory, on the Oracle WebLogic Server, create a directory named `oer_temp`. This directory is used to back up the current Oracle Enterprise Repository application. This location is used as a temporary location for properties files, JDBC drivers, and any customized Oracle Enterprise Repository items (`home.jsp`, for example) that are necessary for the new application.

1.1.3.3 Back Up the Current ALER Webapps

Back-up the current Oracle Enterprise Repository domain by navigating to the `BEA_HOME/user_projects/domains/applications/<domain_name>/applications/oer_XXX/oer_app` directory and copy the `oer-app` and `oer-web` directories to the `oer_temp` directory.

> **Note:** The Oracle Product installer provides an application template to create a or extend an existing domain with the Oracle Enterprise Repository 11*g* application. Therefore, the existing domain is no longer used.

1.1.3.4 Remove the Current Oracle Enterprise Repository Installation Files

> **Caution:** Perform this step only after completing a backup of the directory as described in Section 1.1.3.3, "Back Up the Current ALER Webapps".

If you wish to use the same domain name, then rename /remove the current Oracle Enterprise Repository domain. For example, `BEA_HOME/user_projects/domains/<domain_name>` to `BEA_HOME/user_projects/domains/old_<domain_name>`.

1.1.3.5 Uninstall ALER 3.x/Oracle Enterprise Repository 10.3

Use the ALER 3.x or Oracle Enterprise Repository 10.3 removal tool to remove all components of the previous release. For more information, see "Uninstalling the Software" in *Oracle Fusion Middleware Installation Guide for Oracle Enterprise Repository*.

1.1.3.6 Run the Oracle Enterprise Repository Installation Program

To run the Oracle Enterprise Repository Installation program, perform the following steps:

1. When prompted for the Middleware Home directory, supply the directory where the WebLogic Server 11*g* is installed.

2. When prompted for the Fully qualified server name, supply the fully qualified host name of the server that this domain is being installed on.

3. When prompted for the Repository application name, you can use the same application name that the previous deployment had used.

4. When prompted for the Port number, you can use the same port number that the previous deployment had used.

1.1.3.6.1 Oracle Enterprise Repository Database Installation

When the installer prompts for database installation, choose the option to NOT configure the database.

1.1.3.7 Post-Installation Instructions

The Oracle Enterprise Repository application is now distributed as a bundled WebLogic Server application domain template. To deploy the Oracle Enterprise Repository application template, perform the following tasks:

1. Start the WLS Domain Configuration Wizard using *BEA_HOME/wlserver_ 10.3/common/bin/config.sh* or .cmd

2. Create a new WebLogic Domain, and then click **Next**.

3. Choose Generate a domain configured automatically to support the following products. Check Oracle Enterprise Repository 11.1.1.x.x, and then click **Next**.

4. Enter an appropriate domain name. Use the default locations for the domain and application locations unless you are experienced with these types of definitions. Click **Next**.

5. Supply the Appropriate user credentials required to start/stop the domain processes, and then click **Next**.

6. Select the appropriate JDK and production mode (defaults to JRockit JDK in production mode) for this domain, and then click **Next**.

7. There is no need to modify the optional configuration elements if Oracle Enterprise Repository is a standalone domain (not extending an existing domain - recommended). Click **Next**.

8. Review the configuration information for this new Oracle Enterprise Repository domain, and then click the **Create** button.

The Oracle Enterprise Repository application is deployed within the newly created domain in the directories specified in step 4. This directory path is referenced later to restore existing property and JDBC Driver files.

1.1.3.8 Restore the Oracle Enterprise Repository Back Up Files

To restore the Oracle Enterprise Repository backup files from WebLogic to your Oracle Enterprise Repository 11*g* deployment, perform the following steps:

1. Navigate to the `oer_temp` directory.

2. Edit the *oer_temp/oer/WEB-INF/classes/enterprise.properties* file by adding the following properties to the end of the file:

   ```
   enterprise.authentication.ldap.enabled=true
   enterprise.authentication.container.enabled=true
   enterprise.authentication.advancedcontainer.enabled=true
   ```

3. Copy the `oer_temp/oer-app/WEB-INF/classes/*.xml` files to the *BEA_ HOME/user_projects/applications/<domain>/oer_ 11.1.1.x.x/applications/oer-app/WEB-INF/classes* directory. This action overwrites the XML files that are a part of the new install.

4. Copy the `oer_temp/oer-app/WEB-INF/classes/*.properties` files to the *BEA_HOME/user_projects/applications/<domain>/oer_ 11.1.1.x.x/applications/oer-app/WEB-INF/classes* directory. This action overwrites the .properties files that are a part of the new install. If a **.properties** file exists in the new deployment, and does not exist in the backup, then this may indicate a "Move Setting to Database" operation in the past. These .properties files should be deleted from the upgraded instance. For more information about moving settings to database, see Section 2.3.2, "Move Settings to Database".

5. Copy the `oer_temp/oer/custom/home.jsp` (and/or all of the custom home pages) to the *BEA_HOME/user_projects/applications/<domain>/oer_ 11.1.1.x.x/applications/oer-app/custom* directory. This overwrites the existing `home.jsp` file that is part of the new install. In addition, copy any custom images that were included in the custom home page(s) to the appropriate directory.

6. If using any custom Type icons, copy the files from the `oer_`
`temp/oer-web/images/tree/enterprise/asset` directory to the *BEA_*
*HOME/user_projects/applications/<domain>/oer_*
11.1.1.x.x/applications/oer-web/images/tree/enterprise/asset directory.

7. If custom reports were created, see the Section 2.1, "Migrating Custom Reports"
section.

8. Copy/Move the Upload Repository directory (normally found in `oer_`
`temp/oer/custom/upload`) into the new *BEA_HOME/user_*
projects/applications/<domain>/oer_11.1.1.x.x/applications/oer-app/custom/upload
installation directory.

1.1.3.9 Start WebLogic Server

If you are using LDAP, then confirm that the Enable LDAP Login option is set to `True`
and verify the LDAP Configuration settings. If you are using OracleAS Single Sign-On,
then confirm that the Enable Advanced Container Login option is set to `True` and
verify the Advanced Container Configuration settings. If your LDAP or OracleAS
Single Sign-On configurations are not set correctly, you will not be able to authenticate
with the Oracle Enterprise Repository application.

1. Start the WebLogic Administration Server (required to be running before the
Oracle Enterprise Repository managed server can be started):

*BEA_HOME/user_projects/domains/<domain>/startWebLogic.sh or *.cmd*

2. Start the Oracle Enterprise Repository Managed Server:

*BEA_HOME/user_projects/domains/<domain>/bin/startManagedWebLogic.sh oer_*
server1 http://localhost:7001

1.1.3.10 Test the Application

Oracle Enterprise Repository 11*g* is now deployed. It is recommended that you test the
installation using the Oracle Enterprise Repository Diagnostics page and the
recommended tests are:

1. On the Oracle Enterprise Repository Diagnostics page

- Product Version Information
- Test Required Libraries
- Test Database

2. Update the following Oracle Enterprise Repository properties that reference the
application server file system paths:

`cmee.asset.xml.paths.export-destination`

`cmee.asset.xml.paths.xsl-source`

`cmee.asset.xml.paths.out-destination`

`cmee.server.paths.upload`

`cmee.extframework.export.tempfile`

Additional properties that are affected by the deployment of Oracle Enterprise
Repository are:

`cmee.server.paths.upload-registrar`

The application server based FTP, HTTP, HTTPS and SMB/CIFS artifact store definitions are also affected by the change in paths based on the WLS application template configuration process.

Ensure you take a special note of these types of artifact stores and update the configurations appropriately.

3. If you are unable to access the Oracle Enterprise Repository login page or application, or if you see an error message of some kind, it may be possible that the db.password property in the database.properties file was not encrypted. Check the **cmee.log** file for any messages that explains this behavior.

For more information about enabling the Oracle Enterprise Repository diagnostics tools, see Section 2.6.3, "Enable the Oracle Enterprise Repository Diagnostics Tool".

1.1.4 WebSphere Application Upgrade

These instructions enable you to upgrade from Oracle Enterprise Repository 10.3 to Oracle Enterprise Repository 11*g* on WebSphere using the Oracle Products installation program. These instructions assume use of the Application Server profile type. For other profile types, modify the configuration according to your environment's requirements.

This section contains the following topics:

- Section 1.1.4.1, "Prerequisites"

- Section 1.1.4.2, "Create Oracle Enterprise Repository Temp Directory"

- Section 1.1.4.3, "Back Up the Current Oracle Enterprise Repository Webapps"

- Section 1.1.4.4, "Remove Current ALER/OER Installation Files"

- Section 1.1.4.5, "Uninstall Oracle Enterprise Repository 10.3"

- Section 1.1.4.6, "Run the Oracle Enterprise Repository Installation Program"

- Section 1.1.4.7, "Install New Application Files"

- Section 1.1.4.8, "Restore the Oracle Enterprise Repository Back Up Files"

- Section 1.1.4.9, "Start WebSphere"

- Section 1.1.4.10, "Test the Application"

1.1.4.1 Prerequisites

To upgrade from Oracle Enterprise Repository 10.3 to Oracle Enterprise Repository 11*g* on WebSphere, you must ensure that you have the following prerequisites:

- When using WebSphere 6.1.0.5 with Oracle 11*g* database server, ensure that you use the **ojdbc5.jar** JDBC driver for Oracle Database servers to maintain compatibility with the IBM WebSphere 1.5.x JDK.

- Oracle Enterprise Repository requires WebSphere version 6.1.0.5. If the currently installed Oracle Enterprise Repository application is not running on the required version, then upgrade the application server as appropriate.

- The current deployment directory containing the Oracle Enterprise Repository application is referred to as *oer_current* (this directory normally is located in the following directory:

```
WAS_
HOME/AppServer/profiles/AppSrv01/installedApps/localhostNode0
1Cell/installedApps/oer.ear.
```

- Once the upgrade process is complete, the Oracle Enterprise Repository property files, XML files, and JDBC driver files are located within a new `BEA_HOME/repositoryXXX/websphere` directory.

- Back up the existing ALER database.

1.1.4.2 Create Oracle Enterprise Repository Temp Directory

On the WebSphere server, create a directory named `oer_temp`. This directory contains backups of the current Oracle Enterprise Repository application properties files, JDBC drivers, and any customized Oracle Enterprise Repository items (`home.jsp` for example) that are necessary for the new application.

1.1.4.3 Back Up the Current Oracle Enterprise Repository Webapps

To back up the current Oracle Enterprise Repository webapps, perform the following steps:

1. Copy the contents of the current `OER_HOME` shared library directory contents to the `oer_temp` directory. The `oer_temp` directory should then contain all of the original Oracle Enterprise Repository properties files, XML files, and your current JDBC driver JAR files.

2. Edit the *oer_temp/oer/WEB-INF/classes/enterprise.properties* file by adding the following properties to the end of the file:

    ```
    enterprise.authentication.ldap.enabled=true
    enterprise.authentication.container.enabled=true
    enterprise.authentication.advancedcontainer.enabled=true
    ```

3. Refer to the password encryption requirements section before performing upgrade steps. For more information, see Section 2.6.4, "Creating Encrypted Passwords".

4. If you have a customized Oracle Enterprise Repository home page, navigate to the `oer_current/oer-app.war/custom` directory and copy the `home.jsp` file to the `oer_temp` directory. If you have used the Role or Department based home pages, then copy those files from the directory that is indicated within the `oer_current/oer-app.war/WEB-INF/classes/homepages.xml` file.

5. Copy any custom images that are used in the custom home page file(s) to the `oer_temp` directory.

6. If you have any custom Asset Type icons, navigate to the `oer_current/oer-web.war/images/tree/enterprise/asset` directory and copy the images to the `oer_temp` directory.

7. Optionally, copy all XSLT templates from the PDF Printing web-application to the `oer_temp` directory. This web application is bundled directly into the core of the Oracle Enterprise Repository product.

8. Copy any additional files or directories that were included or modified from the base Oracle Enterprise Repository installation into the `oer_temp` directory.

1.1.4.4 Remove Current ALER/OER Installation Files

To remove the current Oracle Enterprise Repository installation files, perform the following steps:

1. Start the WebSphere 6.1.0.5 Administrative Console.

2. Expand the **Applications** -> **Enterprise Applications** folder and select the **ALER/OER** option.

3. Click **Stop**.

4. Select the **ALER/OER** option.

5. Click **Uninstall**.

6. Select the **ALER/OER** option.

7. Click **Remove**.

8. Repeat the above steps to remove the PDF Printing web-application (if present). This feature has been bundled within the Oracle Enterprise Repository base application.

9. Save the changes to the Master Configuration when prompted.

10. Stop WebSphere Server.

11. Associate Shared Libraries with Oracle Enterprise Repository 11*g*, as follows:

    a. Expand the **Applications**, **Enterprise Applications** folder and click the **Oracle Enterprise Repository 11g** installation.

    b. Choose the **Shared Library References** link.

    c. Select the **oer-app.war** application.

    d. Click the **Reference Shared Libraries** button.

    e. Select the appropriate shared library definitions.

    f. Click the Right Arrow button to move the definitions to the selected box on the right.

    g. Click **OK** to save the selected references.

    Edit the path of the existing Oracle Enterprise Repository shared library containing the property files. The new path for this library is *BEA_ HOME/repository111/websphere*.

    The existing property files are copied into this location once the Oracle Enterprise Repository 11*g* installation is completed.

    > **Note:** For more information about configuring workflows, see *Oracle Fusion Middleware Configuration Guide for Oracle Enterprise Repository*.

1.1.4.5 Uninstall Oracle Enterprise Repository 10.3

Use the Oracle Enterprise Repository 10.3 Installer to remove all components of the previous release. For more information, see "Uninstalling the Software" in *Oracle Fusion Middleware Installation Guide for Oracle Enterprise Repository*.

1.1.4.6 Run the Oracle Enterprise Repository Installation Program

To run the Oracle Products Installation program, perform the following steps:

1. When prompted for the Repository application name, use the same application name that the previous deployment had used.

2. When prompted for the Port number, use the same port number that the previous deployment had used.

   The Oracle Products installer generates a deployable EAR file, thus allowing you to quickly and easily install the updated version of the application directly from

the application server's file system. It also creates and deploys the Oracle Enterprise Repository Workflow application components appropriately.

1.1.4.7 Install New Application Files

To install new application files in Oracle Enterprise Repository, perform the following steps:

1. Start WebSphere Server.

2. Start the WebSphere Administrative Console.

3. Expand the **Applications** -> **Enterprise Applications** folder and click **Install**.

4. Choose the option to select a Remote Filesystem (this option is relative to the application server), browse to the `BEA_HOME/repository111/websphere` directory, and select the `oer.ear` file.

5. Select the Prompt me only when additional information is required option and click **Next** to continue.

6. On the Select Installation Options page, click **Next** to continue after making the appropriate changes.

7. On the Map modules to servers page, click **Next** to continue after making the appropriate changes.

8. On the Map virtual hosts for Web modules page, click **Next** after making the appropriate changes.

9. On the Deployment summary page, click **Finish** to deploy the EAR file.

> **Note:** This process may take several minutes.

10. After the installation is finished, the Application installed successfully message is displayed.

11. Click **Save** to Master Configuration when prompted.

12. Stop WebSphere Server.

1.1.4.8 Restore the Oracle Enterprise Repository Back Up Files

These steps modify files deployed by the newly installed version of Oracle Enterprise Repository with files from the backup and the previously installed version.

1. If you have a customized Oracle Enterprise Repository home page(s), use the Update feature of WebSphere to replace the deployed `custom/home.jsp` file with the customized version of `home.jsp` in the backup.

2. Copy any custom images that may be part of the home.jsp file to the appropriate directory.

3. If you use custom asset icon images, then use the Update feature.

4. If custom reports were created, then see the Section 2.1, "Migrating Custom Reports" section.

5. If additional files were included or modified in the original Oracle Enterprise Repository install, consult Oracle Technical Support for assistance with inclusion of these additions into Oracle Enterprise Repository 10.3.

6. Copy all previous property files from the oer_temp to the `BEA_HOME/repository111/websphere` directory.

7. Copy all previous XML files from the oer_temp to the `BEA_HOME/repository111/websphere` directory.

8. Copy the currently supported JDBC Driver files into the `BEA_HOME/repository111/websphere` directory.

1.1.4.9 Start WebSphere

If you are using LDAP, then confirm that the Enable LDAP Login option is set to `True` and verify the LDAP Configuration settings. If you are using OracleAS Single Sign-On, confirm that the Enable Advanced Container Login option is set to `True` and verify the Advanced Container Configuration settings. If your LDAP or OracleAS Single Sign-On configurations are not set correctly, then you will not be able to authenticate with the Oracle Enterprise Repository application.

1. Navigate to the `WebSphere_HOME/AppServer/temp/` directory and delete the cache for the Oracle Enterprise Repository application.

2. Restart WebSphere.

1.1.4.10 Test the Application

Oracle Enterprise Repository is now deployed. It is recommended that you test the installation using the Oracle Enterprise Repository Diagnostics page and the recommended tests are as follows:

1. On the Oracle Enterprise Repository Diagnostics page

- Product Version Information

- Test Required Libraries

- Test Database

2. Update the following Oracle Enterprise Repository properties that reference the application server file system paths:

`cmee.asset.xml.paths.export-destination`

`cmee.asset.xml.paths.xsl-source`

`cmee.asset.xml.paths.out-destination`

`cmee.server.paths.upload`

`cmee.extframework.export.tempfile`

Additional properties that are effected by the deployment of Oracle Enterprise Repository are:

`cmee.server.paths.upload-registrar`

The application server based FTP, HTTP, HTTPS and SMB/CIFS artifact store definitions are also effected by the change in paths based on the WLS application template configuration process.

Ensure you take note of these artifact stores and update the configurations appropriately.

3. If you are unable to access the Oracle Enterprise Repository login page or application, or if you see an error message of some kind, it may be possible that the `db.password` property in the `database.properties` file was not encrypted. Check the **cmee.log** file for any messages that explains this behavior.

1.1.5 Migrating Your Custom Assets

> **Note:** The following instructions apply to generic services that are manually created in Oracle Enterprise Repository 10g or ALER 3.x, with standard artifacts specified by WSDL. These instructions do not pertain to services automatically harvested from other sources through Oracle Enterprise Repository integrations.

You can migrate your custom assets using any of the following methods:

Migrating Customized Asset Types

Oracle Enterprise Repository comes with numerous out-of-the-box asset types that can be modified and extended, or used to create new custom asset types. This section describes how these asset types are migrated to Oracle Enterprise Repository 11*g*.

- New Asset Types - If you created your own custom asset types, then these are left intact when upgrading to 11*g*.

- Asset Type Extensions - If you extended standard out-of the box asset types, then these are left intact when upgrading to 11*g*.

- Oracle Enterprise Repository 11*g* Solution Pack - The Oracle Enterprise Repository 11*g* Harvester uses an updated Solution pack. The Solution pack modifies some of the asset types in Oracle Enterprise Repository. All asset type modifications are additive and all new elements are added to the bottom of the tabs in the Asset Editor.

Migrating Manually Created Services Instances

In Oracle Enterprise Repository 11*g*, assets are harvested throughout the lifecycle. Once a run-time implementation has been harvested into Oracle Enterprise Repository, the repository's workflow process the implementation according to the governance rules and practices established by the organization. As the implementation moves throughout the lifecycle, from testing, through staging, and into production, the enterprise repository promotes the services and endpoints to a service registry in each lifecycle environment.

Services harvested at design-time provide early visibility into service production, but these services cannot be reused through JDeveloper until they have been deployed. Similarly, services without a concrete endpoint and hosted WSDL cannot be promoted to Oracle Service Registry.

If you want to consume Oracle Enterprise Repository assets from JDeveloper and/or move assets from Oracle Enterprise Repository to Oracle Service Registry so run-time tooling can automatically detect changes, you must follow the following process:

- Asset Type Required

  For instances of services to be migrated properly to the new 11*g* model, the Oracle Enterprise Repository 10*g* R3 asset type that you must use is <Service>. In Oracle Enterprise Repository 10*g*, the <Service> type is available both in Base Data and the Harvester Solution Pack. In ALER 3.x, it is available in Base Data and the Introspection Solution Pack. To ensure that you have the right Type, the UUID is:

  <Service> af6ee48a-4c10-11dc-ac31-7de68f5ca4a1

  While it is possible (with more advanced migration tool configuration) to migrate assets of any asset type or other types called "Service", you should use the original

Service asset type from Oracle with the proper UUID. This ensures that you have the right asset Type in 11g, so that you can take advantage of Oracle Enterprise Repository functionality that is dependent on the Service asset Type.

- Remote Artifact Hosting Required

  The migration tool runs the Harvester on any valid WSDL that is referenced remotely on HTTP, HTTPS, or FTP sites by the Service assets being migrated. The references to XSDs found in WSDL must also be valid URIs, and hosted remotely from the system on which the migration tool is being run.

  Only HTTP/HTTPS and FTP are supported for remote hosting of these artifacts. Services with artifacts hosted locally or elsewhere does not migrate with good results.

  There are no specific instructions on setting up hosted artifacts, as long as that the remote file store can make the artifacts accessible through a valid URI. You should follow instructions for the product that you are using for the remote file store.

- Concrete WSDL Preferred

  If a concrete WSDL is attached to the Service, interface and endpoint assets are created on migration. If the abstract WSDL is attached, interface assets are created without endpoint assets. Remotely hosted artifacts must include concrete WSDL with endpoints for lifecycle use cases. For example, if endpoints are not present, services and artifacts are visible in the repository, but services are neither consumable in Oracle SOA Suite/JDeveloper, nor do the services publish to the Service Registry until they are re-harvested with endpoints.

- Service Reference to Artifacts

  The FileInfo element on the Service asset is the correct place to manually add the URI artifact reference to the remotely hosted service WSDL.

  There is no need to add multiple FileInfo entries for each service. Only the top-level WSDL is required to be referenced in FileInfo, because the Harvester used by the migration tool automatically parses WSDL and looks for import/Ref to get all the other referenced artifacts such as other WSDL/XSDs.

  > **Note:** If you have ALER and have not yet upgraded to Oracle Enterprise Repository, then the URI references to the remotely hosted WSDL artifacts (per instructions above) should be inserted in the same place that the artifacts would have been referenced after ALER WSDL introspection.

  ALER WSDL introspection is a legacy feature and no longer available in Oracle Enterprise Repository. For best results in an upgrade, any previously introspected artifact references should be deleted from the Service before the Oracle Enterprise Repository 11*g* upgrade, with references to remotely hosted artifacts manually inserted in their place. If this procedure is not followed, then the previously introspected WSDL is still available from Oracle Enterprise Repository, but certain important use cases are not supported, for example services are not consumable in Oracle SOA Suite/JDeveloper or published to the Service Registry until the concrete WSDL is re-harvested in Oracle Enterprise Repository 11*g*.

1.2 Upgrade Oracle Enterprise Repository Workflows

It is highly recommended that the currently delivered workflows be deployed instead of using the versions delivered in a previous release. The previous version of workflows are not supported with Oracle Enterprise Repository 11*g*.

The Oracle Enterprise Repository version 11*g* contains new and updated workflows that require the use of Oracle BPM Enterprise version 10.3.1. The earlier versions of the Oracle Enterprise Repository had been delivered with a run-time edition of OBPM 10*g* R3. The currently delivered version of the workflows are incompatible with the OBPM 10g R3 engine and, therefore, requires a new installation of OBPM Enterprise version 10.3.1. Alternatively, you can upgrade existing OBPM installation to version 10.3.1 using BPM or upgrade existing OBPM installation to version 10.3.1 using Standard BPM upgrade procedure.

1.2.1 Upgrading Existing Oracle Enterprise Repository Workflows

You can upgrade existing Oracle Enterprise Repository workflows; these workflows contain all of the Oracle Enterprise Repository workflow processes, in addition to new processes surrounding UDDI operations. If you have deployed customized workflows have been deployed for use with Oracle Enterprise Repository, you must upgrade these customizations using OBPM 10.3 studio.

You must regenerate `workflow.xml` after Oracle Enterprise Repository 11*g* is deployed. This regeneration ensures that generated `workflow.xml` contains correct metadata, for example, categorizations and users. Once generated, it should be updated for correct registrar credentials and correct Oracle Enterprise Repository URL. Lastly, it should be updated for any custom event wiring, which is a manual step. It is also recommended that you thoroughly test it before deployment.

For information about installation of the workflows within OBPM 10.3, see *Oracle Fusion Middleware Configuration Guide for Oracle Enterprise Repository*.

> **Note:** The deployment of custom or customized workflows is outside the scope of this document.

2

Post Installation Tasks

This chapter describes the upgrade process tasks and requirements when upgrading from Oracle Enterprise Repository 10.3 to Oracle Enterprise Repository 11*g*.

This chapter contains the following sections:

- Section 2.1, "Migrating Custom Reports"
- Section 2.2, "Configuring Oracle Enterprise Repository Reports"
- Section 2.3, "Customizations"
- Section 2.4, "Oracle Enterprise Repository Legacy Eclipse Plug-in Uninstallation"
- Section 2.5, "Install Oracle Enterprise Repository Eclipse Plug-in on Client Workstations"
- Section 2.6, "Utilizing the Diagnostic Testing Tool"

2.1 Migrating Custom Reports

You must edit the `CustomReports.xml` file, as mentioned in the steps below:

1. Navigate to the `./WEB-INF/config/reports/` directory on the application server.

2. Open the **customreports.xml** file in a text editor.

3. Edit the meta-tags for report name, display name, description, and external to reference a custom report.

2.2 Configuring Oracle Enterprise Repository Reports

This section provides an overview of the Business Intelligence (BI) Publisher and the process of installing it.

Oracle Business Intelligence Publisher is required by Oracle Enterprise Repository to run the reports, which are available on the Oracle Enterprise Repository reports page.

You can install the BI Publisher Enterprise Server with the default installation option, which installs a OC4J server. You can also deploy the BI Publisher Enterprise Server on your other application servers, such as WebLogic, WebSphere. The recommended BI Publisher version is 10.1.3.4.1.

If BI Publisher is used only with Oracle Enterprise Repository, then the Oracle Enterprise Repository pre-configured BI Publisher content is used. The Basic type of BI Publisher installation is recommended. For information about installing BI Publisher, see *http://download.oracle.com/docs/html/B25825_01/toc.htm*.

> **Note:** The patch 9021224 is also required and must be applied to the BI Publisher server.

Oracle Enterprise Repository includes a collection of pre-configured BI Publisher content. This content needs to be deployed and configured before running any Oracle Enterprise Repository reports.

For more information about deploying Oracle Enterprise Repository pre-configured BI Publisher content, see "Configuring BI Publisher Enterprise Server" in the *Oracle Fusion Middleware Configuration Guide for Oracle Enterprise Repository*.

2.3 Customizations

An installation of Oracle Enterprise Repository 10.3 may have several customizations that are unique to that installation. These are not limited to, but may include:

- Open API programs written against the REX interface
- BPM Workflows
- Endpoint Event Subscriptions
- Eventing for WebSphere Application Server
- Clustering
  - See Section 2.3.2, "Move Settings to Database"
- Application Properties Moved to Database

2.3.1 General Instructions

For all of the customizations mentioned in the Overview section, the customizations should be repeated on Oracle Enterprise Repository 11*g*, following the successful upgrade.

2.3.2 Move Settings to Database

As part of clustering configuration and some other customizations, the Move Settings to Database option in Oracle Enterprise Repository copies the System Settings out of the property files in to the Database. During the Upgrade process some property files have been restored to their default configuration. These should be removed from the upgraded Oracle Enterprise Repository instance again.

The Oracle Enterprise Repository System Settings section contains all of the configuration options for the application deployment. When the Oracle Enterprise Repository application starts, the **database.properties** file is loaded and a connection to the database configured is established, allowing the application to load additional configuration settings. Oracle Enterprise Repository then opens a specific set of properties files found within the application's classpath. The system settings found within the files overwrite those that had been loaded from the database. The changes to system settings, when the Oracle Enterprise Repository Web interface saves them, are stored in the location from which the property had been loaded upon startup. Those properties that had been loaded from the database and not subsequently loaded from a properties file are again stored in the database.

Oracle Enterprise Repository supports and provides the option of moving settings to the database. This option is performed for the following purposes:

- Configuration of Oracle Enterprise Repository for deployment into a clustered application server configuration. For more information, see "Configuring and Clustering Oracle Enterprise Repository" in the *Oracle Fusion Middleware Installation Guide for Oracle Enterprise Repository*.

- A heightened security environment where an application EAR file is used for the deployment of Oracle Enterprise Repository. If the Oracle Enterprise Repository properties files are included within the EAR file upon deployment and the EAR is not exploded during deployment, then the Oracle Enterprise Repository properties are set in a read-only state since the Oracle Enterprise Repository application is unable to write to files located within the application EAR file

In the un-exploded EAR deployment option, the best practice instructions to configure Oracle Enterprise Repository and allow system settings to be modifiable are as follows:

1. Explode the application EAR file and the **oer-app.war** file contained within the EAR.

2. Add the correctly configured Oracle Enterprise Repository properties files into the *oer-app/WEB-INF/classes* directory.

3. (Optional) You may also add the JDBC Driver jar file(s) to the *oer-app/WEB-INF/lib* directory.

4. Use a ZIP or JAR program to re-create the **oer-app.war** file.

5. Use a ZIP or JAR program to re-create the **oer.ear** file (containing the **oer-app.war** and **oer-web.war** files).

6. Deploy the **oer.ear** file into your application server.

7. Start the Oracle Enterprise Repository application/server.

8. Access the Oracle Enterprise Repository application's Web interface, select **Admin** and **System Settings**. The Move Settings to Database link appears at the bottom of the System Settings page. Click the **Move Settings to Database** link.

9. Stop the Oracle Enterprise Repository application and undeploy the application EAR file.

10. Explode the Oracle Enterprise Repository EAR file and the **oer-app.war** file contained within the EAR.

11. Remove all properties files (except **database.properties**) from the *oer-app/WEB-INF/classes* directory.

12. Use a ZIP or JAR program to re-create the **oer-app.war** file.

13. Use a ZIP or JAR program to re-create the **oer.ear** file (containing the **oer-app.war** and **oer-web.war** files).

14. Deploy the oer.ear file into your application server.

15. Start the Oracle Enterprise Repository application/server.

16. Access the Oracle Enterprise Repository application's Web interface, select **Admin**, **System Settings**.

17. Change a setting within the system settings, for example, `enterprise.defaults.displayname.field = emailaddress`.

18. Save the change and confirm that the setting has been retained when the System Settings page reloads.

19. Change the setting, which had been change tested to its previous value and save the settings again.

20. Make any additional system settings changes as necessary within the Oracle Enterprise Repository Admin Web interface.

The application is now configured to be utilized within the secure EAR deployed environment.

> **Note:** If these files listed in the Clustering Guide exist in the backup of the `WEB-INF/classes` directory, then do not remove the file. You must remove those files that do not exist in the backup and were delivered by the upgrade process.

2.4 Oracle Enterprise Repository Legacy Eclipse Plug-in Uninstallation

To uninstall Oracle Enterprise Repository Eclipse plug-in, perform the following steps:

1. In Eclipse, click **Help**.

2. Click **Software Updates**.

3. Click **Manage Configuration**.

4. Select **Show Disabled Features**.

5. Select the **Oracle Enterprise Repository** feature.

6. Click **Disable Feature**.

> **Note:** Unless the **Oracle Enterprise Repository** feature is properly disabled before uninstallation, some Oracle Enterprise Repository files may remain in the Eclipse plug-ins directory. These files must be then manually deleted.

7. Follow the prompts to restart Eclipse.

8. After restart, select **Manage Configuration** in the Help menu.

9. Select the **Oracle Enterprise Repository** feature.

10. Select **Uninstall**.

11. Follow the prompts to restart Eclipse.

2.5 Install Oracle Enterprise Repository Eclipse Plug-in on Client Workstations

You can install Eclipse 3.5 (Galileo) from the standard Eclipse download site. The Oracle Enterprise Repository plug-in included the necessary Oracle Enterprise Pack for Eclipse (OEPE) plug-in dependencies with it. You can find the OEPE package at

http://www.oracle.com/technology/software/products/oepe/oepe_11113.html

To install workspace studio on client workstations, see *http://download-llnw.oracle.com/docs/cd/E12840_01/common/docs103/install/index.html*.

2.6 Utilizing the Diagnostic Testing Tool

The Diagnostics tool allows testing and troubleshooting of certain aspects of Oracle Enterprise Repository application. This section contains the following topics:

- Section 2.6.1, "Launching the Diagnostic Tool"

- Section 2.6.2, "Navigating the Diagnostics Tool"

- Section 2.6.3, "Enable the Oracle Enterprise Repository Diagnostics Tool"

- Section 2.6.4, "Creating Encrypted Passwords"

- Section 2.6.5, "Checking Product Version Information"

- Section 2.6.6, "Test Servlet Functionality"

- Section 2.6.7, "Test Required Libraries"

- Section 2.6.8, "Test Database Connectivity"

- Section 2.6.9, "List System Paths"

- Section 2.6.10, "Run Pre-Compile Servlet"

- Section 2.6.11, "Additional Functionalities"

2.6.1 Launching the Diagnostic Tool

To launch the Oracle Enterprise Repository Diagnostics tool, navigate to

```
http://host_name:port_number/application_name/diag/index.jsp
```

2.6.2 Navigating the Diagnostics Tool

In addition to the Diagnostics Home and Oracle Enterprise Repository links, the left sidebar of the Diagnostics tool includes links to several tests that can be run to check your installation of Oracle Enterprise Repository. Click any of these links to display the available tests in the main pane of the Diagnostics tool.

- Generic Web App

    - Installation Tests

        * Product Version Information

            Provides information on the installed version of the Oracle Enterprise Repository, the version of the installed Oracle Enterprise Repository database, upgrade dates, and the maintenance mode setting for the database.

        * Snoop Servlet

            Provides an output listing of all cookies, headers, CGI-Variables, Servlet Context Attributes, Session information and Session values set as a part of the request to the application server.

        * Test Required Libraries

            Provides information about the application server's configuration regarding required JAR library files, installed XML parsers, and the presence of the minimally required Oracle Enterprise Repository properties files.

        * Run Active Diagnostic Test

            Runs a JUnit test against the installed Oracle Enterprise Repository application. Many of these tests are required to allow Oracle Enterprise Repository to build and parse XML data.

* Test Database

    Provides insight into the JDBC driver functionality, including the number of database side functions that are expected to be supported. Also tests the connection string set in the database.properties file, provides additional information regarding the version of the JDBC driver, and may also indicate deficiencies within the database tables.

* List System Paths

    Lists the pathways (local and absolute URLs) configured within the `cmee.properties` file and, also those stored within the database. Also helps to indicate why some parts of the application deployment function while others do not (such as images not loading).

- Tools

    * JSP Precompiler

        Forces the application server to perform an immediate JIT compile of JSP pages within the Oracle Enterprise Repository application. This process greatly improves the performance of page loads if performed after each application server restart

    * Encrypt Strings for Passwords

        Performs a Hash on a clear-text password string supplied in the text field. This feature is used for encrypting passwords to be stored within application properties files. This tool should not be used to encrypt user passwords. Contact Oracle Support for the proper procedure to update a password in the database.

- Oracle Enterprise Repository Repository Tests

    - Runtime - Configuration

        * List Enabled System Settings

            Lists all values set in the Oracle Enterprise Repository database and, also those set in properties files.

        * Edit System Paths

            Allows an administrator to make permanent changes to the `cmee.properties` file on the application server.

    - Runtime - Assets

        * List Types

            Produces XML output that describes all Asset Types stored within the Oracle Enterprise Repository database.

        * Edit Asset Custom Data

            Allows a registrar to modify XML data contained within an asset in the event that an asset's metadata is otherwise uneditable. Do not use this tool unless specifically directed by Oracle Support personnel.

        * Get XML: Show Type ID

            Produces the XML description of a specific Asset Type based on the unique ID of the Asset Type record.

        * Get XML: Type ID As String

Produces the XML description of a specific Asset Type based on the unique ID of the Asset Type record as one continuous string.

* Get XML: Type XML Schema

This tool produces the XML schema of a specific Asset Type based on the unique ID of the Asset Type record.

* Get XML: Type XML Schema as String

This tool produces the XML schema of a specific Asset Type based on the unique ID of the Asset Type record as one continuous string.

* Get XML: Asset ID

- Produces the XML data of an Asset based on the unique ID of the Asset.

- The include extended metadata option produces additional information related to Policies and other data.

* Get XML: Asset ID as String

Produces the XML data of an Asset based on the unique ID of the Asset as one continuous string.

- Runtime - Security

■ Optional Tests

- SOAP Services

Test the SOAP URL for Rex API service using

http://server:port/oer_app_name/service/FlashlineRegistry?wsdl

This process produces the Oracle Enterprise Repository Web Service WSDL file.

- Logging Options

■ To access the Oracle Enterprise Repository login page, click the **Oracle Enterprise Repository** link in the sidebar of the Diagnostics page.

■ To activate the debugging mode for the current session, click the turning logging on link in the main pane.

2.6.3 Enable the Oracle Enterprise Repository Diagnostics Tool

Starting with Oracle Enterprise Repository 11*g* Release 1 (11.1.1.6.3), the diagnostics pages are disabled, by default. To enable this feature, you must pass a JVM parameter, during the application server startup.

If you are using the WebLogic Server, the parameter is provided as a part of the *Oracle_HOME/user_projects/domains/DOMAIN_NAME/bin/setOERDomainEnv.sh* or `*.cmd` file. Uncomment the `OER_DIAG_OPTION` environment variable and restart the `oer_server`.

If you are using WebSphere, then you must add the following JVM Parameter to the server(s) running Oracle Enterprise Repository:

`-DdiagPagesEnabled=true`

This parameter should be appended to any existing values within the WebSphere administration server path of:

Application servers, server1, Process Definition, Java Virtual Machine, Generic JVM Arguments field

2.6.4 Creating Encrypted Passwords

Oracle Enterprise Repository 11*g* Release 1 (11.1.1.6.3) requires that all cleartext passwords are encrypted. The existing ALER/Oracle Enterprise Repository installations support but do not mandate encrypted passwords in property files.

Oracle Enterprise Repository is now distributed with a password encryption command-line tool, which is useful in encrypting passwords for the Oracle Enterprise Repository REX OpenAPI processes and, also for updating properties files.

This utility can be found in *BEA_HOME/repostioryXXX/core/tools/solutions/XX.X.X.X.X-OER-PasswordTools.zip*.

This zip file has two scripts (for the Windows and Unix/Linux platforms, respectively):

- **encrypt.bat(.sh)** - encrypt an xml configuration file's password elements

- **encryptpassword.bat(.sh)** - encrypt a single password from the command line

An example xml file for use with the `encrypt.bat/.sh` file is:

```
<passwords>
<!-- database.properties -->
<password>example_db.password</password>
<!-- ldap.properties -->
<password>example_ldap.bindPassword</password>
<!-- activemq.xml -->
<password>activemq</password>
<!-- EndPointEventSubscription.xml -->
<password>aler_workflow_pass</password>
</passwords>
```

The output of either of these utilities must be used within the appropriate properties or xml configuration file instead of the clear-text password that are present currently. This restriction is applicable to the Oracle Enterprise Repository server-side application deployment and, also Oracle Enterprise Repository included in client-side packages such as the Harvester and Exchange Utility.

A utility included within the ALER/Oracle Enterprise Repository diagnostics page enables you to encrypt existing passwords, thus enabling the upgrade process to proceed.

To create encrypted passwords, perform the following steps:

1. Navigate to the Diagnostics page using *http://host_name:port_number/application_name/diag/index.jsp*. The Oracle Enterprise Repository Diagnostics page is displayed.

2. Click **Tools**, **Encrypt Strings for Passwords**. The Oracle Enterprise Repository Diagnostics Password Encryption page is displayed.

3. Open the Oracle Enterprise Repository **database.properties** file within your existing Oracle Enterprise Repository deployment and enter the **db.password** property value into the diagnostics page, and click the **Submit Query** button.

The resulting string should replace the db.password field value. You must escape the '=' symbols within the encrypted password value. This is accomplished by using the backslash character immediately before the equals character. For example,

Cleartext:

`db.password=example_word`

Encrypted:

```
db.password=v2_1.16MCs4Mc8hfvRvVS/WaEhQ\=\=
```

Repeat these steps for the **ldap.properties** file for the **ldap.bindPassword** property, if this property contains a value.

2.6.5 Checking Product Version Information

To check the product version information, perform the following steps:

1. Click **Installation Tests** under Generic Web App.

2. Click **Product Version Information** in the main pane.

    A new browser window opens to display information on the internal product version, database schema version, and information about the most recent upgrade/install.

2.6.6 Test Servlet Functionality

To test the servlet functionality in Oracle Enterprise Repository, perform the following steps:

1. Click **Installation Tests** under Generic Web App.

2. Click **Snoop Servlet** in the main pane.

    A new browser window opens to display information on the parameters passed from the Web client to server.

2.6.7 Test Required Libraries

To test the required libraries, perform the following steps:

1. Click **Installation Tests** under Generic Web App.

2. Click **Test Required Libraries** in the main pane.

    A new browser window opens to display information on the property files and system properties essential to proper functionality.

2.6.8 Test Database Connectivity

To test the database connectivity in Oracle Enterprise Repository, perform the following steps:

1. Click **Installation Tests** under Generic Web App.

2. Click **Test Database** in the main pane.

    A new browser window opens to display information on container access, the availability of JDBC drivers and the presence of important Oracle Enterprise Repository data tables.

2.6.9 List System Paths

To list the system paths in Oracle Enterprise Repository, perform the following steps:

1. Click **Installation Tests** under Generic Web App.

2. Click **List System Paths** in the main pane.

    A new browser window opens to display all Oracle Enterprise Repository paths.

2.6.10 Run Pre-Compile Servlet

To run the pre-compiler servlet, perform the following steps:

1. Click **Tools** under Generic Web App.

2. Click **JSP Precompiler** in the main pane.

   A new browser window opens to display a list of system JSPs that have been pre-compiled for speedier first time access.

   > **Note:** This servlet produces errors within the error log; reset logs before starting the application server.

2.6.11 Additional Functionalities

The additional Diagnostics testing tool functionality includes:

- Testing browser redirection compatibility to both a standard URL or local file.

- Listing all enabled Oracle Enterprise Repository Settings.

- Listing all Types in the system.

- Displaying XML for a specific Type (both editor and viewer).

- Displaying xml for a specific asset.

- Clustering Status shows a list of all nodes in the cluster.

Ensure that client browser and JRE installations have appropriately supported versions for Oracle Enterprise Repository.

> **Note:** The Firefox 3.6 browser requires Java SE 6 Update 10 or later. Otherwise, Java-based web applications do not work. For more information, see *http://www.java.com/en/download/faq/firefox_newplugin.xml*.

3

Data Migration Tool

This chapter describes how to use the data migration utility to migrate assets from earlier versions of Oracle Enterprise Repository.

This chapter contains the following sections:

- Section 3.1, "Overview"
- Section 3.2, "Using the Data Migration Tool"
- Section 3.3, "Configuring the Data Migration Tool"

3.1 Overview

The Data Migration Tool migrates the source asset and artifact file information to new OER 11g Asset Models (Harvested Models), when the file information artifacts are remote and links them with a special relationship (harvester-migrator), for tracking. Here is a sample view of a pre and post migration asset model.

Figure 3–1   Ad-hoc Created Asset View–(Pre-Migration View)
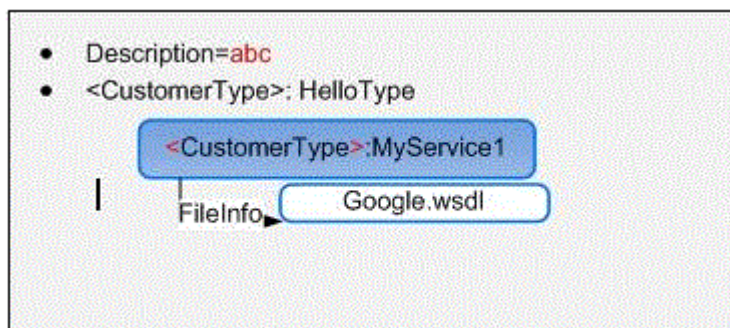
Figure 3–2   OER 11g Asset View – (Post-Migration View)
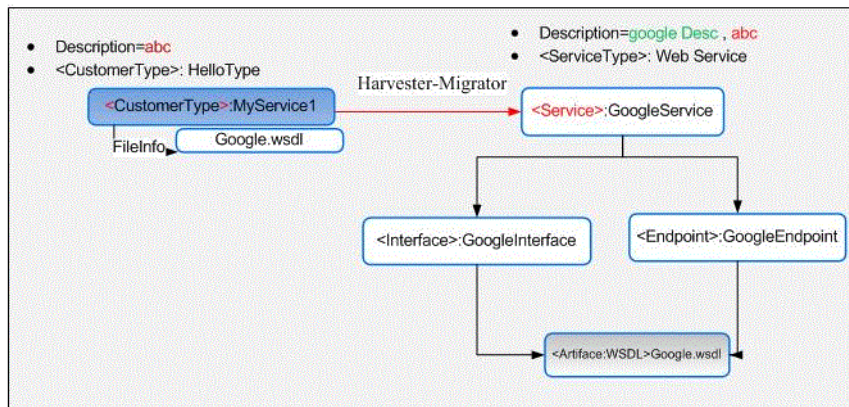
Key Features of the Data Migration Tool

The Data Migration Tool assists you in these tasks:

- Migrating ad-hoc created assets to OER11g Asset model, given the source asset has valid file info with WSDL/XSD artifacts (all harvester supported artifacts <provide link>).

- Migrating all the ALER 3.x & OER 10.x Harvested Assets to confirm to OER11g Harvester model.

- Migrating most of the key mandatory-data from source asset to migrated root service asset (ref the <known issue> section for unsupported data migration elements)

- Migrating any Custom Type Asset Create by Customer to 11g Harvester model to take the all benefits provided in OER 11g SOA governance feature.

- Migrating custom-data defined on out-of-the-box or custom asset types to destination assets (*)

- Setting the source asset to Retire /Delete at the end of post migration.

3.1.1 Best Practices

In the Oracle Enterprise Repository 11g R1 release, the harvester was extended to support Oracle SOA Suite 11g and Oracle Service Bus 10g. The Exchange Utility was enhanced to support Oracle Service Bus. The Enterprise Manager Integration utility was added to integrate with metrics in Enterprise Manager 10g. To support these products, the Oracle Enterprise Repository asset model was extended, which is known as the 11g R1 Asset Model.

This section describes the key points about Oracle Enterprise Repository content migration in an upgrade:

- The custom types are left intact when upgrading to 11*g*. Also, the extensions, for example, the addition of metadata attributes, but not necessarily deletions, made in the type manager to standard out-of the box asset types also remain intact. The instances of these types with their extended metadata are ordinarily preserved in an upgrade.

- Some or all instances of service assets/artifacts that you previously harvested or captured at design time must be harvested or reharvested with deployment/endpoint information. This is true, for example, if you and your

customer require Oracle Enterprise Repository/Oracle Service Registry use cases such as consumption in Oracle SOA Suite/JDeveloper, registry/repository exchange between Oracle Enterprise Repository and Oracle Service Registry, and other more advanced lifecycle use cases that are beyond simple visibility in Oracle Enterprise Repository).

- To behave properly in Oracle Enterprise Repository 11*g*, artifact files must either be hosted remotely on an HTTP/HTTPS/FTP site or harvested directly from deployment instances. In addition, endpoints must be present from concrete WSDL in order for Oracle Enterprise Repository to publish to Oracle Service Registry.

  Typically, this re-harvesting is recommended for any services/artifacts that have locally cached artifact files in file info.

- If you have manually entered service assets/artifacts (non-harvested) of the same types that are typically used by the harvester, then you can manually plug-in a remote reference to the concrete WSDL hosted on an HTTP/HTTPS/FTP site before an upgrade to Oracle Enterprise Repository 11*g* and the harvester Migration Tool (`11.1.1.x.x-OER-Migrate.zip`) creates the new Harvester model.

3.1.2 Prerequisites

Before using the migration tool, you must perform the following prerequisites:

- The Oracle Enterprise Repository server instance must first be upgraded to 11*g* R1, and must be started.

- The Oracle Enterprise Repository database must be backed up, so it can be restored if there are any problems with the migration.

- The migration tool requires the Java JRE or JDK version 6 or higher.

- The migration tool connects to Oracle Enterprise Repository using its WebService API. So it can be run on any computer that can access the Oracle Enterprise Repository server. It runs much faster if run on the same computer as the Oracle Enterprise Repository server.

- The Oracle Enterprise Repository System Setting, `cmee.extframework.enabled`, must be set to `true`. This can be set from the Web UI in the Admin, System Settings page.

- Only assets with a status of *Active* will be migrated.

3.1.3 High Level Use Cases

You can use the migration tool to:

- Migrate assets that were imported into Oracle Enterprise Repository through the ALER 3.x WSDL importer in the Web console.

- Migrate assets that were imported into Oracle Enterprise Repository through the ALER 3.x UDDI importer in the Web console.

- Migrate assets that were imported into Oracle Enterprise Repository through the ALER 3.x BPEL importer in the Web console.

- Migrate assets that were imported into Oracle Enterprise Repository through the ALER 3.x ALRR-XU (version 3.x).

■ Migrate assets that were imported into Oracle Enterprise Repository through the ALER 3.x ad-hoc "Submit an Asset" functionality in the Web console.

■ Migrate assets that were imported into Oracle Enterprise Repository through the 10*g* R3 Harvester.

■ Migrate assets that were imported into Oracle Enterprise Repository through the 10*g* R3 Exchange Utility.

■ Migrate WSDL assets that were imported into Oracle Enterprise Repository through the Open API (REX) WSDL import functionality.

■ Preview the results of a migration, without committing any changes to Oracle Enterprise Repository.

3.1.4 Migration Tool Functionality

This section describes the migration tool functionality for ALER 3.x assets and Oracle Enterprise Repository 11*g* assets.

3.1.4.1 Migration Tool Functionality - ALER 3.x Assets

In the past, Oracle Enterprise Repository had a variety of tools for importing metadata. In version 3.x of ALER, these included Web and Open API (REX) utilities for importing from WSDL, BPEL, and UDDI and the command-line RRXU tool. These import service metadata into an asset model called the ALER 3.x Asset Model, as shown.

Figure 3–3   ALER 3.x Asset Model
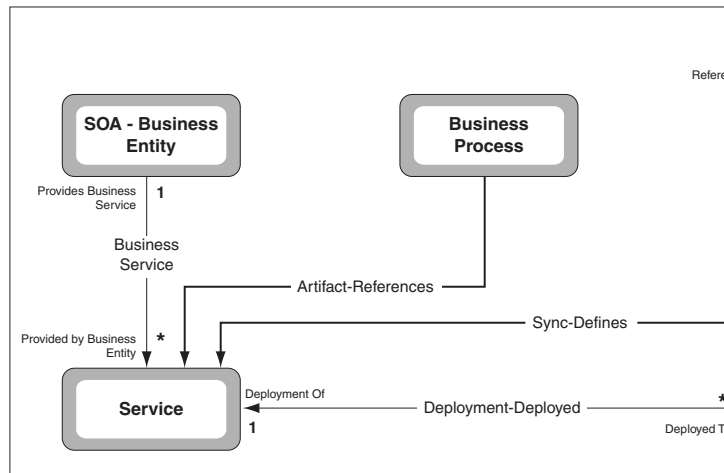
Service Assets

The contents of the WSDL that is pointed to by the Service FileInfo is introspected from its download URI. An interface asset is created, and related to the service. An Artifact: WSDL asset is introspected from its download URI. An Artifact: WSDL asset is created and related to the endpoint. A relationship, Harvester-Migrator, is created

between the original Service asset and the new Service asset that is created by the introspection.

Business Process Assets

The contents of the BPEL that is pointed to by a business process asset is introspected from its download URI. An Artifact: BPEL asset is created and related to the business process. The business process asset has its asset type changed to Business Process: BPEL. A relationship, Harvester-Migrator, is created between the original BPEL asset and the new Business Process: BPEL asset that is created by the introspection.

Imports

Any WSDLs or XSDs imported by the WSDL results in new artifact assets and relationships, as in the Harvester (or points to existing assets if their fingerprints match).

PartnerLinks

Any WSDLs used by a BPEL in partnerlinks results in new artifact assets and relationships, as in the Harvester.

FileInfos

The FileInfo is removed from the service and endpoint assets. The new Artifact: WSDL asset contains the updated FileInfo.

Fingerprints

The SFID (fingerprint) is removed from the service and endpoint assets. Any Artifact: WSDL assets and Artifact: XSD assets contain a new SFID, using the Harvester's SFID algorithm.

Internal Names

Any non-artifact asset that is migrated gets a new internal name, which is used for duplicate checking.

Names

Pre-existing service, endpoint, and business process assets keep their original name. New assets are named according to the Harvester's naming rules.

WSDL Summary

WSDL Summary metadata entries are created on the migrated assets, as in Harvester.

Manifest Metadata

Manifest metadata entries are created on the migrated assets, as in Harvester, to support the new download functionality in Oracle Enterprise Repository.

Ad-hoc Assets (through "Submit an Asset"): Limited Support

Assets that were imported into Oracle Enterprise Repository in an ad-hoc manner are migrated as described in Section 3.3.2.1, "Asset Types to Migrate", only if they conform to the ALER 3.x Asset model. For example, if an ad-hoc service asset contains a FileInfo that points to a WSDL, it is migrated as described above. But an asset of another type, for example, "TestCase", are not migrated.

3.1.4.2 Migration Tool Functionality - Oracle Enterprise Repository **10*g* R3 Assets**

Fingerprints

Recalculates and store the SFID for artifact assets from the original artifact files, using the 11gR1 fingerprinting code.

Manifests

Updates the `internal.introspector.manifest.store` CMF entry and FileInfo with the correct artifact location according to the following 11*g* R1 harvester rules:

- Artifacts that were harvested from remote URLs have the FileInfo updated to point to the remote location (to match the Manifest).

- Artifacts that were harvested from local files have the Manifest updated to point to the repository download URL (to match the FileInfo).

- Deletes obsolete `internal.artifact.store` CMF entries for artifacts that were harvested from remote URLs.

Exchange Stores

Updates the `internal.alrr.exchange.store` CMF entry to conform to the latest Exchange Utility rules.

- Sets `<tmodel>` entries in the `internal.alrr.exchange.store` CMF entry on Service and Endpoint assets, with the UDDI keys of porttype and binding TModels. (Connects to the UDDI registry to get these).

- Sets the `<uddiRegistries>` custom data table on Endpoint assets, including the `registry-name`, `registry-url`, and `service-key` elements. The `registry-name` and `registry-url` elements are set from the original `internal.alrr.exchange.store` CMF entry. The service-key is read from the UDDI registry during migration.

- Sets the <uddi> custom data elements on Service assets, including the business-key and service-key elements. These are read from the UDDI registry during migration.

- Sets the <uddi> custom data elements on Business Entity assets, specifically the business-key element. This is read from the UDDI registry during migration.

Harvester Properties

Converts the `internal.introspector.store` CMF entry to the new format used by 11gR1 harvester.

- Sets the `intname` harvester property for abstract assets, according to the 11gR1 harvester rules.

- Calculates and sets the following harvester properties, which are automatically set by 11gR1 harvester, on the appropriate assets: `Namespace`, `Short Name`, `SCA Name`, `Endpoint URI`, `Deployment` URL, Transport Type, Interface Type, Service Type.

- Updates the `wsdlLocation` element in the WSDLSummary, based on the download URL calculated in "Manifests" on page 3-6.

Asset Type

Updates the asset types according to the 11*g* R1 model:

- Changes Endpoint: WebService assets to Endpoint

■ Changes Interface: WebService assets to Interface

3.2 Using the Data Migration Tool

This section contains the following topics:

■ Section 3.2.1, "Running from Command Line"

3.2.1 Running from Command Line

The migration tool is available in the `11.1.1.x.x-OER-10gMigrate.zip` file. Unzip this file to a directory on your computer. The migration tool can be run in the command line using the **migrate.bat** utility (for Windows) or **migrate.sh** (for Linux and Unix).

Before running migrate.bat or migrate.sh, ensure that the environment variables mentioned in Table 3–1 are set. In Windows, from a command window, you can type `set X` to see the value of the variable X, and `set X=abc` to set the value of `FOO` to `abc`.

Table 3–1   Command Line Script

Environment Variable	Description
JAVA_HOME	Ensure that the `JAVA_HOME` environment variable points to an installed java runtime (JRE) or SDK. This must be Java version 5 or higher.
JAVA_OPTS	Ensure that you set your `JAVA_OPTS` parameter as follows:
	set `JAVA_OPTS=-Dhttp.proxyPort=80 -Dhttp.proxyHost=www-proxy.us.oracle.com -Dhttp.nonProxyHosts= "*.oracle.com|localhost"`
	`JAVA_OPTS` refers to the extra options to the java executable. In normal cases, there is no need to set this variable. However, a common exception when you need to set this variable is when your computer is inside a firewall, and you need to use an HTTP proxy to access external servers.
	See Also: *http://java.sun.com/javase/6/docs/technotes/guides/net/proxies.html*

You can preview the results of a migration in the Data Migration tool, without committing any changes to Oracle Enterprise Repository, by using the command-line options. Table 3–2 describes the command-line options for the Data Migration tool.

Table 3–2   Command Line Options for the Data Migration Tool

Environment Variable	Description
-preview	If true, the data migration tool runs in the preview mode. A detailed information about successes and failures are logged. No changes are committed to Oracle Enterprise Repository.
	If false, the data migration tool runs in the production mode. A detailed information about successes and failures are logged, and the changes are committed to Oracle Enterprise Repository.

3.3 Configuring the Data Migration Tool

This section contains the following topics:

■ Section 3.3.1, "Setting the Repository Connection Information for the Command-line Utility"

■ Section 3.3.2, "Advanced Configuration"

- Section 3.3.3, "Known Issues"

3.3.1 Setting the Repository Connection Information for the Command-line Utility

Open the `HarvesterSettings.xml` file located at `<Data Migration Home>` and modify the following XML chunk to point to the Oracle Enterprise Repository instance with the right credentials.

```
<repository>
  <uri>http://localhost:8080/oer/services/FlashlineRegistry</uri>
  <credentials>
    <user>admin</user>
    <password>*****</password>
    <enableTransaction>false</enableTransaction>
    <triggerEvent>false</triggerEvent>
  </credentials>
  <timeout>120000</timeout>
</repository>
```

> **Note:** You should run the Migration Tool as a user with the Basic Access Settings for **Assets - View**, **Edit**, **Accept**, and **Register**.

The passwords in this file must be encrypted before running the migrate script. To encrypt the passwords, use the `encrypt.bat/encrypt.sh` script, which is located in *<Oracle_home>/tools/solutions/11.1.1.x.x-OER-PasswordTools.zip*.

3.3.2 Advanced Configuration

The section describes the following advanced configuration options for the migration tool:

- Section 3.3.2.1, "Asset Types to Migrate"
- Section 3.3.2.3, "Logging"

3.3.2.1 Asset Types to Migrate

By default, the migration tool is configured to migrate the assets of the types that are created by default by the ALER 3.x and 10*g* R3 tools. However, these legacy importers can be configured to use different asset types, through the `types.properties` file.

If the assets were imported using nondefault asset types, you must configure the migration tool to use the same nondefault asset types. These can be configured by editing the following XML chunk in the `HarvesterSettings.xml` file:

```
<query>
    <repositoryQuery>
      <typesToMigrate>
        <assetType.service>Service</assetType.service>
        <assetType.endpoint>Endpoint: Web Service</assetType.endpoint>
        <assetType.endpoint>Endpoint</assetType.endpoint>
        <assetType.interface>Interface: Web Service</assetType.interface>
        <assetType.interface>Interface</assetType.interface>
        <assetType.process>Business Process: BPEL</assetType.process>
        <assetType.process>Business Process</assetType.process>
        <assetType.processDeployment>Deployment:
        BPEL</assetType.processDeployment>
        <assetType.xsd>Artifact: XSD</assetType.xsd>
```

```
        <assetType.wsdl>Artifact: WSDL</assetType.wsdl>
        <assetType.bpel>Artifact: BPEL</assetType.bpel>
        <assetType.xslt>Artifact: XSLT</assetType.xslt>
    </typesToMigrate>
</repositoryQuery>
```

> **Note:** You should migrate all Assets during a single run of the
> migration tool. Individually migrating specific Asset types can cause
> problems such as having the same internal names.

A new attribute has been added that determines what the "source" asset's active status
will be after the migration has completed.

For example, when migrating "CustomerServiceType" to "Service", you can say you
want all assets of type "CustomerServiceType" to be set to "Retired".

```
<assetType.service sourceActiveStatus="retired"
name="CustomerServiceType" />
```

3.3.2.2 Custom Data Migration

Configuration of custom data is done using the provided customDataMigration.xml
file. Configuration must occur for each asset type that will be migrated.

Determine what the "source" asset type is, and what the "destination" (11gR1) asset
type is, for example, "MyCustomServiceType" is the source, and it will be migrated to
"Service", the destination asset type.

In the customDataMigration.xml, this is defined as:

```
<migratingAssetTypes>
    <migrationAssetType>
        <sourceTypeName>MyCustomServiceType</sourceTypeName>
        <destinationTypeName>Service</destinationTypeName>
```

Once the asset types are established, determine which custom data elements to
migrate. To do this, define the XML Mappings for each element for both the source
and the destination.

The XML Mapping names can be obtained doing the following:

1. Launch the Asset Editor.

2. Launch the Type Manager.

3. Navigate to File->Export Schema and select the desired **Asset Type**.

The schema will display all of the XML Mappings for the asset type, specifically within
the element, `<xs:element name="custom-data">`. For example: `<xs:element
name="development-hours" >`, where the name attribute is the XMLMapping.

Configuration requires that you determine the XML Mapping for the custom data
element in the source and ALSO the XML Mapping that you want to migrate the
source element to.

The example below migrates custom data from the MyCustomServiceType asset type
to the Service asset type. For every asset of that type, the value stored in the
"development-hours" field will be migrated to the corresponding "development-hours"
field in the Service asset type, as well as "estimated-time-to-use". In the
customDataMigration.xml, this is defined as:

```
<migratingAssetTypes>
  <migrationAssetType>
   <sourceTypeName>MyCustomServiceType</sourceTypeName>
   <destinationTypeName>Service</destinationTypeName>
    <customDataFields>
     <customDataField>
      <sourceElement>
       <xmlMapping name="development-hours" />
      </sourceElement>
      <destinationElement>
       <xmlMapping name="development-hours" />
      </destinationElement>
     </customDataField>
     <customDataField>
      <sourceElement>
       <xmlMapping name="estimated-time-to-use" />
      </sourceElement>
      <destinationElement>
       <xmlMapping name="estimated-time-to-use" />
      </destinationElement>
     </customDataField>
    </customDataFields>
```

In this sample, the source and destination XML Mappings are the same, however that is not always the case. The configuration allows the ability to migrate data into any defined XML Mapping field.

> **Note:** If the defined XML Mapping field in the destination asset does not exist, it must be added in the Type Manager before running the migration tool. Failure to define it results in a failure while running the migration tool.

Migrating Custom Data Fields that are Tables

There are certain custom data fields that are tables, which means they have custom data fields within custom data fields. These types can be migrated, but the setup is more complicated. A table, called "SourceTable1", has two fields within it, "SourceText1" and "SourceDate1". When defining these XML Mapping fields in the configuration file, the highest level XML Mapping must be defined first, in this case, "SourceTable1". The children of that field do not need to be in any order, however, they must correspond correctly when defining the destination XML Mappings.

```
<migratingAssetTypes>
  <migrationAssetType>
   <sourceTypeName>MyCustomServiceType</sourceTypeName>
   <destinationTypeName>Service</destinationTypeName>
    <customDataFields>
     <customDataField>
      <sourceElement>
       <xmlMapping name="SourceTable1" />
       <xmlMapping name="SourceText1" />
       <xmlMapping name="SourceDate1" />
      </sourceElement>
      <destinationElement>
       <xmlMapping name="DestinationTable1" />
       <xmlMapping name="DestinationText1" />
       <xmlMapping name="DestinationDate1" />
      </destinationElement>
     </customDataField>
```

```
        </customDataFields>
```

This example provides an alternative to the example above:

```
<sourceElement>
 <xmlMapping name="SourceTable1" />
 <xmlMapping name="SourceDate1" />
 <xmlMapping name="SourceText1" />
</sourceElement>
<destinationElement>
 <xmlMapping name="DestinationTable1" />
 <xmlMapping name="DestinationDate1" />
 <xmlMapping name="DestinationText1" />
</destinationElement>
```

This example illustrates an error because the SourceText1 field will be migrated to DestinationDate1.

```
<sourceElement>
 <xmlMapping name="SourceTable1" />
 <xmlMapping name="SourceText1" />
 <xmlMapping name="SourceDate1" />
</sourceElement>
<destinationElement>
 <xmlMapping name="DestinationTable1" />
 <xmlMapping name="DestinationDate1" />
 <xmlMapping name="DestinationText1" />
</destinationElement>
```

Migrating Custom Data Fields with Acceptable Value Lists

Custom Data fields with acceptable value lists (AVLs) can also be migrated, however there is one extra step involved. The custom data field has its XML Mapping, but it also references the XML Mapping of the actual AVL.

This can also be found within the exported asset type schema.

```
<xs:element name="avl-dropdown">
  <xs:complexType>
    <xs:simpleContent>
      <xs:extension base="AcceptableValueList_50018">
        <xs:attribute name="id" type="xs:long" use="optional"/>
      </xs:extension>
    </xs:simpleContent>
  </xs:complexType>
</xs:element>
```

This is translated in the custom data configuration like this:

```
<sourceElement>
 <xmlMapping name="avl-dropdown"
  multipleValueListItemMapping="AcceptableValueList_50018"/>
</sourceElement>
<destinationElement>
 <xmlMapping name="destination-avl-dropdown"
  multipleValueListItemMapping="Destination-AcceptableValueList_50020"/>
</destinationElement>
```

3.3.2.3 Logging

The migration tool uses log4j for logging the detailed tasks performed and the log file is placed in the < Migration Tool Home>. The logging options can be changed by updating the `log4fl.properties` file located in the <Migration Tool Home>.

3.3.3 Known Issues

This section describes the following known issues in the Data Migration tool:

- The Data Migration tool does not support assets that are created using the ALSync framework, from ALSB or ALDSP. ALSB assets must be re-introspected using Harvester. Harvesting from ALDSP is currently not supported.

- The Data Migration tool, when migrating exceptionally large data sets, may take hours to run. Some application servers may time out the sessions and cause the migration to fail. To prevent this timeout, you must add the following entry to the `web.xml` file, within the `<webapps>` element, where the number you supply is in minutes:

```
<session-config>
<session-timeout>120</session-timeout>
</session-config>
```

- The Data Migration tool ignores the assets with local referenced artifacts, such as `Fileinfo local with  rep://`.

- Migration of following metadata element Types are NOT supported.

  - Name

  - Version

  - Service Type

  - Uploaded Submission Files

  - Consuming Projects

  - Asset Usage

  - Review

  - Metadata Files

  - Assembly Properties Viewer

- Migration of SFID is NOT supported.

- Migration of UNIQUE element is not supported between the same type by logging warning message.

Index