

Oracle® Warehouse Builder
Installation and Administration Guide
11g Release 2 (11.2)
E17130-03

August 2010

Copyright © 2000, 2010, Oracle and/or its affiliates. All rights reserved.

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

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

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

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

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

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

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

This program contains Batik version 1.6.

Apache License

Version 2.0, January 2004

<http://www.apache.org/licenses/>

For additional information about the terms and conditions, search for "Apache License" in Oracle Warehouse Builder online help.

Contents

Preface.....	xiii
Audience.....	xiii
Documentation Accessibility	xiii
Related Documents	xiv
Conventions	xiv

Part I Installing and Configuring Oracle Warehouse Builder

1 Overview of Installation and Configuration Architecture

Choosing an Appropriate Installation Configuration	1-1
Architecture and Components of Oracle Warehouse Builder	1-1
Single Computer Topology.....	1-2
Separate Client and Database Server Topology	1-3
Multiple Database Topology	1-3
Separate Client, Control Center Service, and Database Server Topology	1-4
Control Center Agent Topology	1-5
About the Check for Updates Wizard for Oracle Warehouse Builder.....	1-6

2 Installing Oracle Warehouse Builder on the Server

Installing Warehouse Builder on Oracle Database 11g Release 2.....	2-1
Working with the Warehouse Builder Standalone Install Package	2-2
Installing Oracle Warehouse Builder Through Oracle Universal Installer	2-3
Installing Warehouse Builder in Oracle RAC Environments.....	2-7
Installing Warehouse Builder on Each Node of a Cluster	2-9
Checklist: Before Using the Universal Installer in an Oracle RAC Environment.....	2-9
For Windows Users Installing in an Oracle RAC Environment	2-9
For UNIX Users Installing in an Oracle RAC Environment.....	2-10
Ensuring the Availability of Service Names for Oracle RAC Nodes	2-11
Performing a Silent Install and Using Response Values.....	2-11
After the Installation is Complete.....	2-11

3 Configuring Oracle Warehouse Builder for Windows

Configuring OWB on Windows Installed as Part of Oracle Database 11g Release 2	3-1
Configuring OWB on Windows Installed as a Standalone Instance	3-2

About Configuring OWB Installed as a Standalone Instance.....	3-2
Cleaning an Oracle Warehouse Builder Repository	3-3
Creating an OWBSYS Schema with Repository Objects	3-3
Configuring the Control Center Service	3-4
Unlocking the OWBSYS and OWBSYS_AUDIT Accounts	3-5
Configuring Oracle Workflow to Support OWB Process Flows	3-5
Setting the OWB Home for Remote Installations	3-5
Creating the First Workspace in the Repository on Windows.....	3-6
4 Configuring Oracle Warehouse Builder for Linux	
Configuring OWB on Linux Installed as Part of Oracle Database 11g Release 2	4-1
Configuring OWB on Linux Installed as a Standalone Instance.....	4-2
About Configuring OWB Installed as a Standalone Instance.....	4-2
Cleaning an Oracle Warehouse Builder Repository	4-3
Creating an OWBSYS Schema with Repository Objects	4-3
Configuring the Control Center Service	4-4
Unlocking the OWBSYS and OWBSYS_AUDIT Accounts	4-4
Configuring Oracle Workflow to Support OWB Process Flows	4-5
Setting the OWB Home for Remote Installations	4-5
Creating the First Workspace in the Repository on Linux	4-5
5 Uninstalling Oracle Warehouse Builder	
About Uninstalling Oracle Warehouse Builder	5-1
Deleting the Workspace Users	5-2
Deleting the Workspace	5-6
Deleting the Schema Objects	5-9
Deleting a Repository from an Oracle Database.....	5-10
Uninstalling the Oracle Warehouse Builder Software.....	5-10
6 Moving to Oracle Warehouse Builder 11.2 from Earlier Releases	
Understanding MDL Migration and Repository Upgrades.....	6-1
Choosing Between MDL Migration and Repository Upgrade.....	6-1
MDL Migration Paths.....	6-2
Repository Upgrade Paths from OWB 10.1 or 9.n to release 11.2	6-3
Choosing a Repository Upgrade Scenario	6-3
In-Place Upgrade of OWB 10gR2 or 11gR1 Repository to 11gR2	6-4
Preparing for In-Place Repository Upgrade.....	6-4
Performing the In-Place Repository Upgrade.....	6-6
Upgrading a Repository to release 11.2 through MDL Files	6-10
Exporting an OWB Repository to a File.....	6-10
Importing a Repository Export MDL File to OWB Release 11.2	6-12
Migrating the Repository to a Different Oracle Database Instance.....	6-14
Upgrading Locations to OWB Release 11.2	6-15
Upgrading Locations to OWB Release 11.2 Using the Repository Browser.....	6-15
Changes to Repository Objects When Moving to OWB Release 11gR2	6-17

7 Configuring Repository Browser, Control Center Agent, and Other Components

Configuring the Repository Browser Environment	7-1
Changing the Session Timeout.....	7-1
Configuring Control Center Agent for Oracle Application Server	7-2
Modifying the OC4J Startup Script.....	7-2
Configuring Control Center Agent Server-side Logging.....	7-3
Configuring the Audit Data Source.....	7-3
Setting Control Center Agent Security Roles.....	7-4
Creating JMS Queues.....	7-9
Setting the PATH Environment Variable	7-11
Deploying the Control Center Agent to OC4J	7-12
Configuring the Control Center Agent Web Service (jrt-web.war).....	7-16
Enabling Control Center Agent Audit Table Logging	7-18
Configuring OWBSYS Audit Connection Pool.....	7-18
Configuring an Audit Table Data Source.....	7-21
Installing JDBC Drivers for OC4J.....	7-23
Overview of Control Center Agent Files	7-24
ODL Job Log Files	7-24
Control Center Agent Run-Time Log File	7-24
JobID File	7-24
Control Center Agent Settings	7-25
JRTAdministratorMBean	7-25
Installing Code Templates for Use from Control Center Agent	7-26
Integrating with E-Business Suite	7-26
Installing Third-Party Name and Address Data	7-27
Integrating with Oracle Workflow.....	7-28

Part II Administering Oracle Warehouse Builder

8 Managing Configurations

About Configuration Management	8-1
Creating Configurations Templates.....	8-2
Editing Configuration Templates.....	8-4
Activating Configurations	8-5
Configuring Objects	8-6

9 Managing the Control Center and Repository

Overview of OWB Control Center and Repository Management	9-1
Starting and Stopping the Control Center Service	9-2
Displaying and Managing Control Center Configuration.....	9-2
Managing Control Center Deployment and Execution Jobs	9-2
OWB Repository and Location Administration Utilities	9-3
OWB Administrative Utilities	9-3
abort_exec_request.....	9-3
abort_unit_request	9-3

deactivate_all	9-4
deactivate_deployment	9-4
deactivate_execution.....	9-4
delete_warehouse_object	9-5
display_platform_property	9-5
expedite_exec_request.....	9-5
grant_error_table_privileges	9-6
grant_upgrade_privileges.....	9-6
list_requests.....	9-6
print_exec_details.....	9-6
print_running_exec_details	9-7
purge_audit_template	9-7
reset_repository	9-8
rtrepos_report	9-8
service_doctor	9-8
set_ccs_home.....	9-8
set_platform_property.....	9-8
set_platform_property_from_file	9-9
set_repository_password	9-9
show_service.....	9-9
sqlplus_exec_background_template	9-9
sqlplus_exec_template.....	9-11
start_service	9-12
stop_service.....	9-12
UpdateControlCenter	9-13
UpdateLocation	9-13

10 Managing the Control Center Agent

Starting and Stopping the Control Center Agent	10-1
About the cca_admin Utility	10-2
Starting the cca_admin Utility	10-2
cca_admin Commands.....	10-2
cleanUpJobLogFilesForGivenTimeInterval.....	10-4
cleanUpNLeastRecentJobLogFiles.....	10-4
connect	10-4
disconnect.....	10-4
exit	10-4
getCacheJobStatusSize.....	10-5
getCacheLogNo	10-5
getKmPermissionsFile	10-5
getLogRoot	10-5
getMaxAbortTimeout	10-5
getMaxRuntimeFileSizeMB	10-6
getMaxRuntimeLogSizeMB	10-6
getRuntimeLogEncoding	10-6
getRuntimeLogFileLocation	10-6
getRuntimeLogLevel	10-6

help.....	10-6
isKmSecurityNeeded.....	10-7
isLogToAuditTable.....	10-7
isLogToOdl.....	10-7
isShowRuntimeLoggerLogHostAddr.....	10-7
isShowRuntimeLoggerLogHostId.....	10-7
isShowRuntimeLoggerLogMsgClassPath.....	10-8
isShowRuntimeLoggerLogMsgLogLevel.....	10-8
isShowRuntimeLoggerLogUserId.....	10-8
readSettings.....	10-8
reset.....	10-8
setCacheJobStatusSize.....	10-8
setCacheLogNo.....	10-9
setKmPermissionsFile.....	10-9
setKmSecurityNeeded.....	10-9
setLogRoot.....	10-9
setLogToAuditTable.....	10-10
setLogToOdl.....	10-10
setMaxAbortTimeout.....	10-10
setMaxRuntimeFileSizeMB.....	10-10
setMaxRuntimeLogSizeMB.....	10-10
setRuntimeLogEncoding.....	10-10
setRuntimeLogLevel.....	10-11
setShowRuntimeLoggerLogHostAddr.....	10-11
setShowRuntimeLoggerLogHostId.....	10-11
setShowRuntimeLoggerLogMsgClassPath.....	10-11
setShowRuntimeLoggerLogMsgLogLevel.....	10-11
setShowRuntimeLoggerLogUserId.....	10-12
settingsToString.....	10-12
showAllJobLogFile.....	10-12

11 Using Snapshots to Manage Metadata

About Snapshots and Metadata Loader.....	11-1
Using Snapshots.....	11-2
Creating Snapshots	11-3
Adding Components to a Snapshot	11-7
Managing Snapshots.....	11-10
Managing Snapshot Access Privileges.....	11-10
Comparing Snapshots	11-11
Converting a Full Snapshot to a Signature Snapshot	11-14
Restoring Workspace Objects From Snapshots	11-15
Exporting and Importing Snapshots	11-17
Deleting Snapshots	11-20
Snapshot Performance.....	11-21

12 Using the Metadata Loader

Exporting Metadata from the Design Center.....	12-1
Importing Metadata Using the Design Center.....	12-3
Importing Objects.....	12-4
Combining Import Modes and Matching Criteria	12-5
Import Different Base Languages.....	12-6
Different Base Languages in the Import File and Target Repository.....	12-6
Importing Supported Languages	12-6
Import Validation Rules.....	12-6
Upgrading Metadata from Earlier Releases	12-7
Changes to Workspace Objects When Upgrading from OWB 9.2.....	12-7
Changes to Workspace Objects When Upgrading from OWB 10.1, 10.2 or 11.1	12-8
Checking for Warnings and Error Messages	12-9
Design Center Metadata Management Dialog Boxes	12-9
Metadata Export Dialog Box	12-9
Export Advanced Options Dialog Box.....	12-10
Languages	12-11
Administration	12-11
Metadata Import Dialog Box	12-11
Import Advanced Options Dialog Box	12-14
Languages	12-14
Administration	12-15
Name Conflicts.....	12-15
File Summary Dialog Box	12-15
Metadata Upgrade Dialog Box.....	12-16
Metadata Progress Dialog Box	12-16
Metadata Loader Utilities	12-16
Metadata Export Utility.....	12-17
Metadata Import Utility	12-17
Multiple Session Concurrency and MDL.....	12-18
Metadata Loader Log File	12-18
About Metadata Loader Results	12-19
Managing Versions and History with the Metadata Loader.....	12-19
Accessing the Metadata Loader	12-19
About Metadata Loader Control Files	12-19

13 Managing Security

About Metadata Security	13-1
About the Security Service.....	13-2
Evaluating Metadata Security Strategies	13-3
Minimal Metadata Security Strategy (Default).....	13-3
Multiuser Security Strategy	13-3
Full Metadata Security Strategy	13-4
Registering Database Users.....	13-4
Registering Existing Database Users.....	13-5
Creating New Oracle Database Users	13-8
Changing Database Default Roles	13-12

Editing User Profiles.....	13-12
Roles	13-13
Default Object Privilege	13-13
Object Privileges.....	13-15
System Privileges	13-15
Support for a Multiple-user Environment	13-16
Read/Write Mode.....	13-17
Read-Only Mode	13-17
Defining Security Roles.....	13-17
Everyone Role.....	13-18
Administrator Role	13-18
Editing Role Profiles.....	13-19
Users.....	13-21
Applying Security Properties on Specific Metadata Objects	13-22
Security Tab.....	13-22
Example: Using Security Properties to Freeze a Project Design	13-23
Security Enforcement	13-23
Managing Passwords in Warehouse Builder.....	13-24
Credential Memory on Logon Panel.....	13-25
Changing Passwords that Access Warehouse Builder	13-25
Encrypting Passwords to Warehouse Builder Locations	13-25
General Steps for Troubleshooting Warehouse Builder.....	A-1
Inspecting Log Files in Warehouse Builder	A-2
Error Messages Related to Installation	A-3
Troubleshooting Other Installation Problems.....	A-12
Checking the Java Virtual Machine (JVM).....	A-13
Using OWB Client Logging to Control Logging and Trace Problems	A-13
Checking the System Path of Oracle Warehouse Builder	A-14
Configuring OWB for Oracle RAC	A-14

Index

List of Examples

10-1	Using cleanUpJobLogFilesForGivenTimeInterval Command	10-4
10-2	Using cleanUpNLeastRecentJobLogFiles command	10-4
10-3	Using connect Command	10-4
10-4	Using disconnect Command	10-4
10-5	Using exit Command.....	10-5
10-6	Using getCacheJobStatusSize Command	10-5
10-7	Using getCacheLogNo Command	10-5
10-8	Using getKmPermissionsFile Command	10-5
10-9	Using getLogRoot Command	10-5
10-10	Using getMaxAbortTimeout Command	10-5
10-11	Using getMaxRuntimeFileSizeMB Command.....	10-6
10-12	Using getMaxRuntimeLogSizeMB Command	10-6
10-13	Using getRuntimeLogEncoding Command.....	10-6
10-14	Using getRuntimeLogFileLocation Command.....	10-6
10-15	Using getRuntimeLogLevel Command.....	10-6
10-16	Using help Command	10-7
10-17	Using isKmSecurityNeeded Command	10-7
10-18	Using isLogToAuditTable Command.....	10-7
10-19	Using isLogToOdl Command	10-7
10-20	Using isShowRuntimeLoggerLogHostAddr Command	10-7
10-21	Using isShowRuntimeLoggerLogHostId Command	10-8
10-22	Using isShowRuntimeLoggerLogMsgClassPath Command	10-8
10-23	Using isShowRuntimeLoggerLogMsgLogLevel Command	10-8
10-24	Using isShowRuntimeLoggerLogUserId Command	10-8
10-25	Using readSettings Command	10-8
10-26	Using reset Command.....	10-8
10-27	Using setCacheJobStatusSize Command.....	10-9
10-28	Using setCacheLogNo Command	10-9
10-29	Using setKmPermissionsFile Command	10-9
10-30	Using setKmSecurityNeeded Command	10-9
10-31	Using setLogRoot Command	10-9
10-32	Using setLogToAuditTable Command.....	10-10
10-33	Using setLogToOdl Command	10-10
10-34	Using setMaxAbortTimeout command	10-10
10-35	Using setMaxRuntimeFileSizeMB Command	10-10
10-36	Using setMaxRuntimeLogSizeMB Command.....	10-10
10-37	Using setRuntimeLogEncoding Command	10-11
10-38	Using setRuntimeLogLevel Command	10-11
10-39	Using setShowRuntimeLoggerLogHostAddr Command	10-11
10-40	Using setShowRuntimeLoggerLogHostId Command	10-11
10-41	Using setShowRuntimeLoggerLogMsgClassPath Command	10-11
10-42	Using setShowRuntimeLoggerLogMsgLogLevel Command	10-11
10-43	Using setShowRuntimeLoggerLogUserId Command	10-12
10-44	Using settingsToString Command	10-12
10-45	Using showAllJobLogFiles Command	10-12

List of Figures

1-1	Single Computer Topology for Oracle Database 11g Release 2	1-2
1-2	Two Computer Topology; Separate Client and Database Server Installations	1-3
1-3	Multiple Database Topology	1-4
1-4	Separate Client, Control Service Center, and Database Server Topology	1-4
1-5	Control Center Agent Topology	1-6
8-1	Oracle Warehouse Builder Configuration Management	8-2
11-1	Metadata Change Management Window	11-10

List of Tables

6-1	Repository Upgrade Paths for OWB 10.1 and Earlier	6-3
11-1	Comparison of Snapshots and Metadata Loader.....	11-1
12-1	Import Mode Results.....	12-5
13-1	Privileges Required for the Execution of Operations	13-23
A-1	Naming Oracle RAC Service Nodes	A-14

Preface

This section contains the following topics:

- [Audience](#)
- [Documentation Accessibility](#)
- [Related Documents](#)
- [Conventions](#)

Audience

This document is intended for all persons responsible for installing Oracle Warehouse Builder, including:

- Data warehouse administrators
- System administrators
- Data warehouse and ETL developers
- Other MIS professionals

Documentation Accessibility

Our goal is to make Oracle products, services, and supporting documentation accessible to all users, including users that are disabled. To that end, our documentation includes features that make information available to users of assistive technology. This documentation is available in HTML format, and contains markup to facilitate access by the disabled community. Accessibility standards will continue to evolve over time, and Oracle is actively engaged with other market-leading technology vendors to address technical obstacles so that our documentation can be accessible to all of our customers. For more information, visit the Oracle Accessibility Program Web site at <http://www.oracle.com/accessibility/>.

Accessibility of Code Examples in Documentation

Screen readers may not always correctly read the code examples in this document. The conventions for writing code require that closing braces should appear on an otherwise empty line; however, some screen readers may not always read a line of text that consists solely of a bracket or brace.

Accessibility of Links to External Web Sites in Documentation

This documentation may contain links to Web sites of other companies or organizations that Oracle does not own or control. Oracle neither evaluates nor makes any representations regarding the accessibility of these Web sites.

Access to Oracle Support

Oracle customers have access to electronic support through My Oracle Support. For information, visit <http://www.oracle.com/support/contact.html> or visit <http://www.oracle.com/accessibility/support.html> if you are hearing impaired.

Related Documents

For more information, see the following documents in the Oracle Database 11g Release 2 documentation set:

- *Oracle Warehouse Builder Release Notes*
- *Oracle Warehouse Builder Concepts*
- *Oracle Warehouse Builder Sources and Targets Guide*
- *Oracle Warehouse Builder Data Modeling, ETL, and Data Quality Guide*
- *Oracle Warehouse Builder API and Scripting Reference*

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.
<i>italic</i>	Italic type indicates book titles, emphasis, or placeholder variables for which you supply particular values.
monospace	Monospace type indicates commands within a paragraph, URLs, code in examples, text that appears on the screen, or text that you enter.

Part I

Installing and Configuring Oracle Warehouse Builder

This part contains the following sections:

- [Chapter 1, "Overview of Installation and Configuration Architecture"](#)
- [Chapter 2, "Installing Oracle Warehouse Builder on the Server"](#)
- [Chapter 3, "Configuring Oracle Warehouse Builder for Windows"](#)
- [Chapter 4, "Configuring Oracle Warehouse Builder for Linux"](#)
- [Chapter 5, "Uninstalling Oracle Warehouse Builder"](#)
- [Chapter 6, "Moving to Oracle Warehouse Builder 11.2 from Earlier Releases"](#)
- [Chapter 7, "Configuring Repository Browser, Control Center Agent, and Other Components"](#)

Overview of Installation and Configuration Architecture

This section discusses the architecture and components of Oracle Warehouse Builder and provides installation and configuration procedures.

This section includes the following topics:

- [Choosing an Appropriate Installation Configuration](#)
- [Architecture and Components of Oracle Warehouse Builder](#)
- [About the Check for Updates Wizard for Oracle Warehouse Builder](#)

Choosing an Appropriate Installation Configuration

Depending on the business needs of your organization, you may be implementing different types of Oracle Warehouse Builder installations:

- The primary installation of Oracle Warehouse Builder, which includes the installation and configuration of the main design repository.
- Optional installations on client computers where developers and administrators run Oracle Warehouse Builder client software, which do not contain a repository.
- Optional installations on hosts that run the Control Center Agent service, but do not contain a repository.
- Optional installations on database server hosts, where ETL jobs execute and where the repository stores only run-time data.

See Also:

- ["Architecture and Components of Oracle Warehouse Builder"](#) on page 1 for an overview of examples of various common topologies
- [Chapter 2, "Installing Oracle Warehouse Builder on the Server"](#) for the software installation tasks
- [Chapter 3, "Configuring Oracle Warehouse Builder for Windows"](#)
- [Chapter 4, "Configuring Oracle Warehouse Builder for Linux"](#)

Architecture and Components of Oracle Warehouse Builder

This section describes some common installation configurations, and shows how the various components interact. The primary Oracle Warehouse Builder components include:

- The OWB Client (which includes the Design Center and the Control Center Manager)
- The OWB Browser
- Control Center Service
- Control Center Agent
- The Repository

See Also: *Oracle Warehouse Builder Concepts* for conceptual information about Warehouse Builder components

Depending on your business needs, there are many possible installation configurations. The five most common topologies, which are illustrated in this section, are:

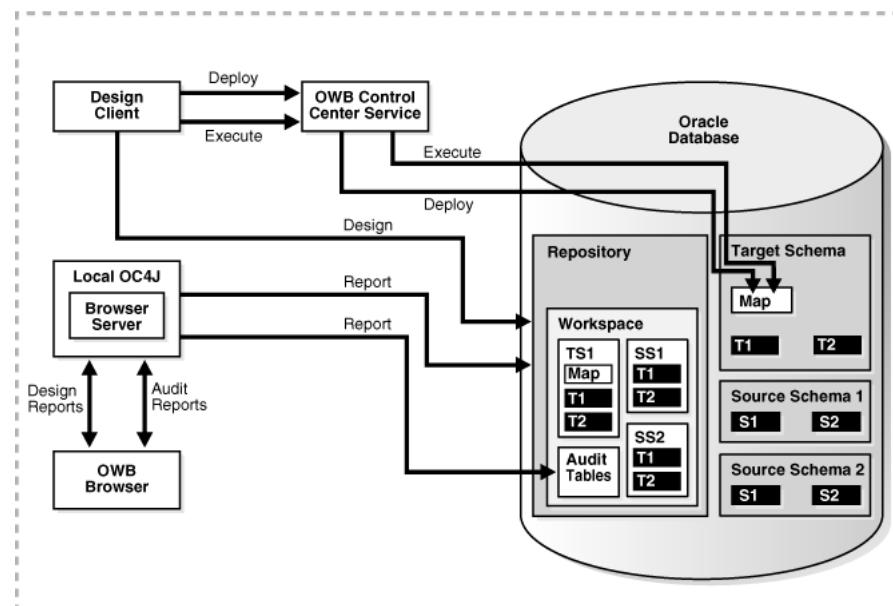
- Single Computer Topology
- Separate Client and Database Server Topology
- Multiple Database Topology
- Separate Client, Control Center Service, and Database Server Topology
- Control Center Agent Topology

Note: in any of these topologies, it is typical to have multiple design clients. For example, each OWB developer or administrator usually has a corresponding, installed OWB client, also called a design client.

Single Computer Topology

When everything is running on the same computer, the database home for Oracle Database 11g Release 2 contains all the components needed to run Warehouse Builder, as illustrated in [Figure 1-1](#).

Figure 1-1 Single Computer Topology for Oracle Database 11g Release 2



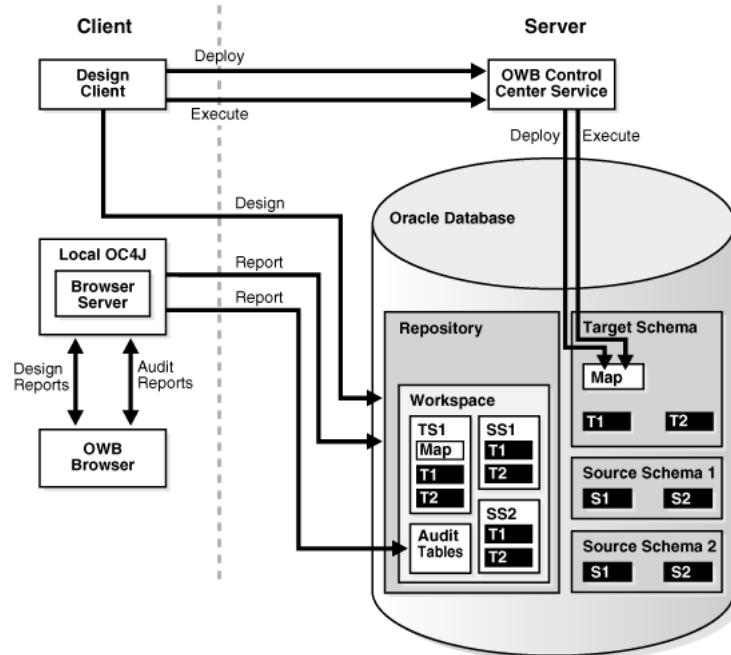
See Also:

- "Configuring OWB on Windows Installed as Part of Oracle Database 11g Release 2" on page 3-1
- "Configuring OWB on Linux Installed as Part of Oracle Database 11g Release 2" on page 4-1

Separate Client and Database Server Topology

If separate computers are being used to run the client and to host the repository, then all computers must have the OWB software installed, as illustrated in [Figure 1-2](#). For the database server installation, Oracle Database 11g Release 2 includes the necessary components. For the client, you must install the OWB software from the standalone install image.

Figure 1-2 Two Computer Topology; Separate Client and Database Server Installations

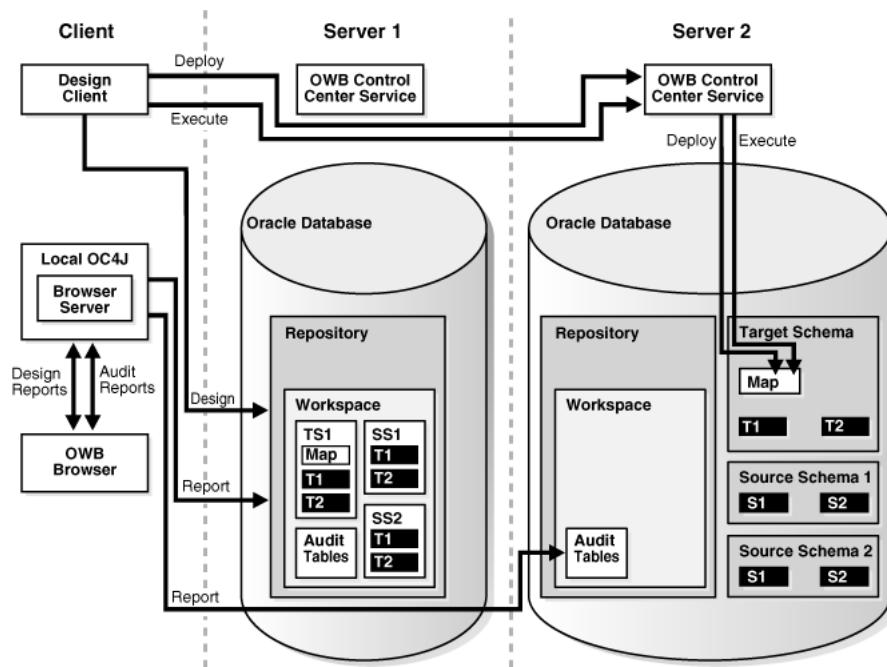


See Also: "Working with the Warehouse Builder Standalone Install Package" on page 2-2 for OWB client installation procedures

Multiple Database Topology

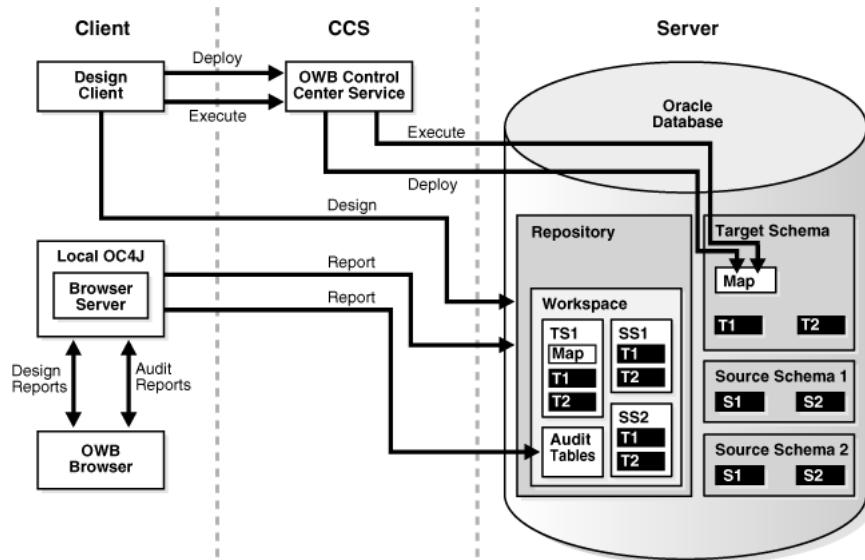
If a database on a separate computer is being used to run mappings designed on another database, then all computers must have the OWB software installed, as illustrated in [Figure 1-3](#). For the database servers, the installation for Oracle Database 11g Release 2 includes the necessary components. For the client, you must install the OWB software from the standalone install image.

You must prepare the OWB repository on all database server computers. As illustrated in [Figure 1-3](#), one acts as the design repository (Server1). The additional servers where mappings execute each have their own repository (shown as Server2), which in this case functions as a run-time repository. The Control Center Service on the execution database server deploys and executes the code.

Figure 1–3 Multiple Database Topology

Separate Client, Control Center Service, and Database Server Topology

If the OWB client, the Control Center Service, and the repository run on different computers, then all these computers must have the OWB software installed, as illustrated in [Figure 1–4](#). For the database server, the installation media for Oracle Database 11g Release 2 includes the necessary components. For the client, you must install the OWB software from the standalone install image. For the Control Center Service, you can use either type of installation. However, Oracle recommends that you use the standalone image because it installs only the components that OWB requires.

Figure 1–4 Separate Client, Control Service Center, and Database Server Topology

See Also:

- ["Installing Warehouse Builder on Oracle Database 11g Release 2"](#) on page 2-1
- ["Installing Warehouse Builder in Oracle RAC Environments"](#) on page 2-7
- ["Working with the Warehouse Builder Standalone Install Package"](#) on page 2-2 for OWB client installation procedures

Control Center Agent Topology

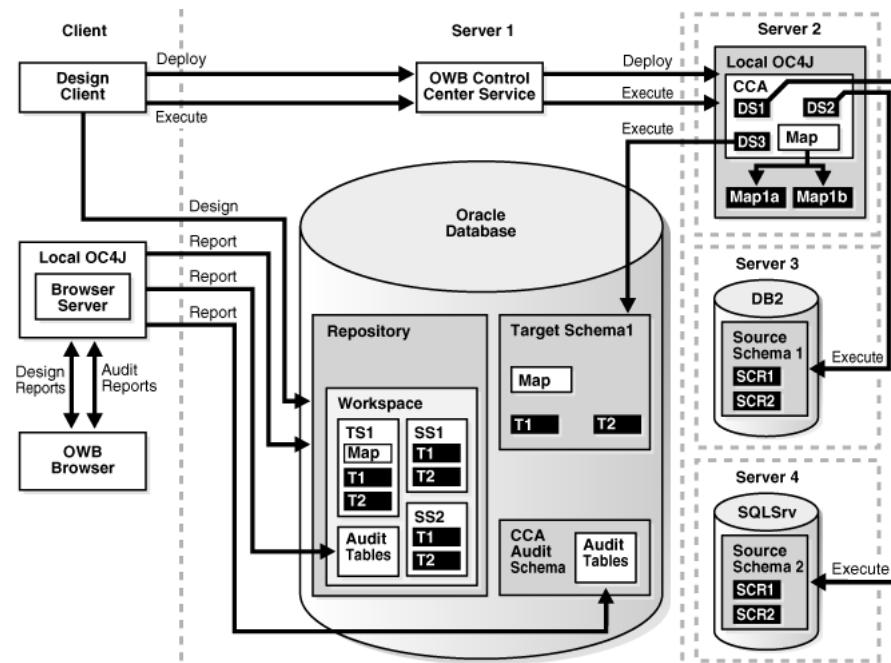
If separate computers are used to run the OWB client, the Control Center Agent service (CCA), and the repository, then all computers used to deploy or execute OWB code must have installed OWB software, as illustrated in [Figure 1–5](#). For the database server, the installation for Oracle Database 11g Release 2 includes the necessary components. For the client, you must install the OWB software from the standalone install image. For the Control Center Agent service, you can use either install. However, because the standalone image only installs the OWB-required components, Oracle recommends this option.

In some deployments, a Control Center Service is required on a host that does not have an Oracle Database installed. For example, if you use code template-based ETL mappings where part of the mapping executes on a non-Oracle database, then you can run a Control Center Agent on that non-Oracle database host. In such a case, you can install the OWB software on the non-Oracle database host.

Also note that:

- On the non-Oracle database host, the only component of the OWB software that is used is the Control Center Agent. The rest of the software is present but not used.
- The CCA may be located with the OWB client or Server1, in which case no additional installation would be necessary.
- In [Figure 1–5](#), the Map1 has three execution units (a, b, c). The first two are Load Code Templates (LCTs) and the other one is an Oracle Target Code Template.

Figure 1–5 Control Center Agent Topology



See Also:

- [Chapter 2, "Installing Oracle Warehouse Builder on the Server"](#)
- [Chapter 7, "Configuring Repository Browser, Control Center Agent, and Other Components"](#)

About the Check for Updates Wizard for Oracle Warehouse Builder

Oracle Warehouse Builder provides the Check for Updates wizard within Design Center to simplify locating and downloading new resources for Warehouse Builder as follows:

- Links to updated release notes, white papers, sample code, and other information resources.
- New code templates, platform definitions and, drivers to extend Warehouse Builder for new sources and targets.
- New experts, scripts, and other utilities that enhance Warehouse Builder or help address specific use cases.

Resources delivered through the Check for Updates wizard may consist of links to My Oracle Support, downloadable documents, or Zip archives. Installation instructions are included for client-side and server-side components with each resource as needed.

See Also:

- My Oracle Support at <https://support.oracle.com> and search on Warehouse Builder
- The product download page on Oracle Technology Network at <http://www.oracle.com/technology/software/products/warehouse/index.html>

Note: The OWB Repository Assistant can be launched from
OWB_HOME/owb/bin/unix/reposinst.sh on Linux or
OWB_HOME/owb/bin/windows/reposinst.bat on Windows.

Installing Oracle Warehouse Builder on the Server

This section discusses how to install Oracle Warehouse Builder on Windows platforms.

This section includes the following topics:

- [Installing Warehouse Builder on Oracle Database 11g Release 2](#)
- [Working with the Warehouse Builder Standalone Install Package](#)
- [Installing Oracle Warehouse Builder Through Oracle Universal Installer](#)
- [Installing Warehouse Builder in Oracle RAC Environments](#)
- [Performing a Silent Install and Using Response Values](#)
- [After the Installation is Complete](#)

Installing Warehouse Builder on Oracle Database 11g Release 2

Oracle Warehouse Builder is part of the standard installation on all hosts running Oracle Database 11g Release 2. When Oracle Database is installed, you must only unlock the OWBSYS and OWBSYS_AUDIT accounts. The OWBSYS schema contains all Warehouse Builder repository metadata, and the OWBSYS_AUDIT schema is used by the Warehouse Builder Control Center Agent to access the heterogeneous execution audit tables of the OWBSYS schema.

Managing Optional Features for Warehouse Builder Installations

Although many features in Warehouse Builder are included in Oracle Database licenses, there are features that require additional licensing. As an administrator, you can assist in managing license compliance by enabling or restricting user access to optional features. The management of optional features as described here is entirely voluntary.

When you first implement a Warehouse Builder repository, all optional features are enabled by default. You can modify these settings when you create the first workspace for the repository or you can modify the settings at a later time.

To manage the usage of optional features

1. Start the Repository Assistant. Select **Start, Programs, ORACLE_HOME, Warehouse Builder**, and then **Administration**.
2. If you are creating the first workspace for the repository, then the Repository Assistant automatically displays the Enable Optional Features page.

If you are modifying an existing repository, then on the Choose Operation page, select **Manage Optional Features**.

3. Select the options you want to make available to Warehouse Builder users.

The selections you make are enforced repository-wide and apply to all users. If a user is logged into the Warehouse Builder client when you make changes to on the Enable Optional Features page, then the changes are enforced for that user the next time they log in to the client.

Note: For more information about each option, refer to the appropriate Oracle licensing information guide, listed on Oracle Technology Network at
<http://www.oracle.com/technology/products/warehouse/htdocs/licensing.html>

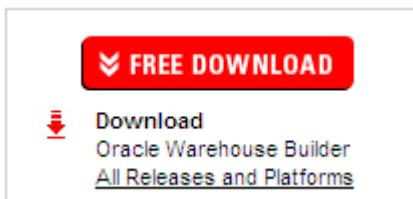
Working with the Warehouse Builder Standalone Install Package

There are several situations when you must download and install the OWB 11g Release 2 Standalone Install Package:

- You are installing OWB 11.2 and hosting an OWB Repository on an earlier release:
 Oracle Database 10g Release 1, Oracle Database 10g Release 2,
 or
 Oracle Database 11g Release 1.
- You run the OWB 11.2 Design Client on a developer's computer.
- You are installing OWB 11.2 on a computer that runs the Control Center Agent.

To obtain Oracle Warehouse Builder standalone software:

1. Navigate the browser to the following URL:
<http://www.oracle.com/technology/products/warehouse/>
2. On the right side of the page, click **All Releases and Platforms**.



3. Under Oracle Warehouse Builder Downloads, select **Accept** for the licence agreement.

Oracle Warehouse Builder Downloads

You must accept the [OTN License Agreement](#) to download this software.

[Accept License Agreement](#) | [Decline License Agreement](#)

The system displays the following message: "Thank you for accepting the OTN License Agreement; you may now download this software."

4. Under the Standalone Software download, click the down arrow beside the platform of your choice to download the ZIP file that contains Oracle Warehouse Builder.

For Windows: choose the **Microsoft Windows** option.

For Linux: choose the **Linux x86** option.

Oracle Warehouse Builder 11g Release 2 (11.2) Standalone Software

This software is automatically installed when you install Oracle Database 11g Release 2.

IMPORTANT: Do not download this standalone software unless you have any of the following needs:

- To install the software on a client computer.

A client computer is any computer that does not host Oracle Database 11g Release 2. If you want to run any Warehouse Builder components on a client computer, including the OMB Plus scripting utility and the Control Center, you must install the standalone software.

- To host the repository on Oracle Database 11g Release 1 or Oracle Database 10g Release 2
- To enable integration with Oracle Discoverer

 [for Microsoft Windows \(1,207,368,211 bytes\)](#)

 [for Linux x86 \(1,205,551,625 bytes\) \(cksum 163128023\)](#)

5. When the download is complete, extract the ZIP file into a temporary directory. For example,

For Windows: use C:\temp\OWB.

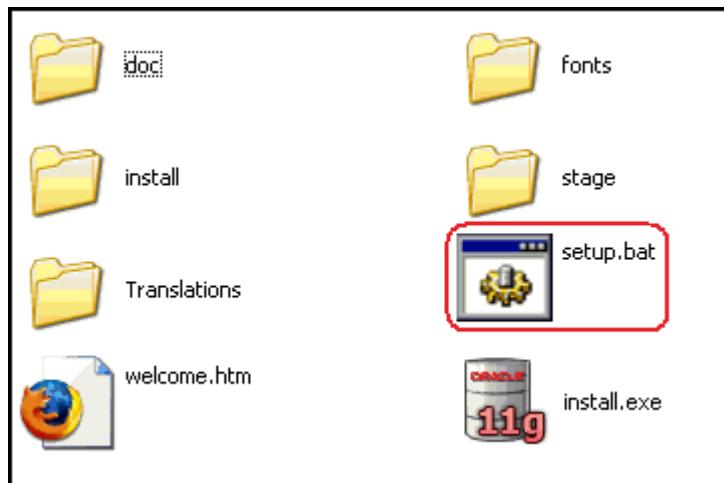
For Linux: use /tmp/OWB.

Installing Oracle Warehouse Builder Through Oracle Universal Installer

After you obtain the standalone installation files for Oracle Warehouse Builder and extract them, you must install the software.

To install Warehouse Builder using Oracle Universal Installer:

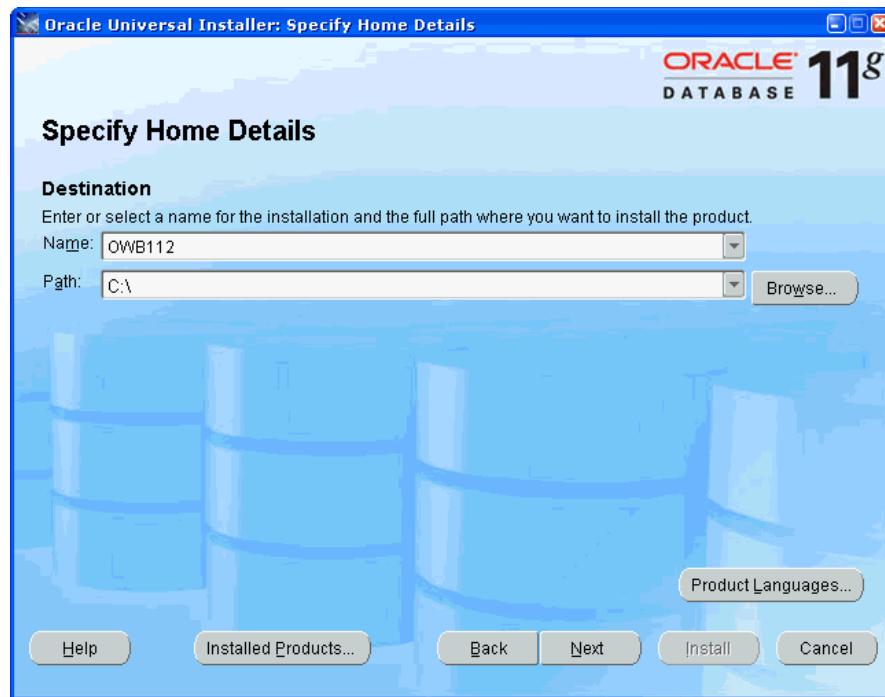
1. In Windows Explorer, navigate to the directory that contains the extracted Oracle Warehouse Builder installation files. In this discussion the sample directory is C:\temp\OWB.
2. Click the setup.bat file to start the Universal Installer.



3. On the Welcome screen, click Next.



4. On the Specify Home Details screen, specify a new Oracle home:
 - Enter the Oracle home **Name**, for example, OWB112.
 - Enter or browse to **Path** C:\.
 - Click **Next**.

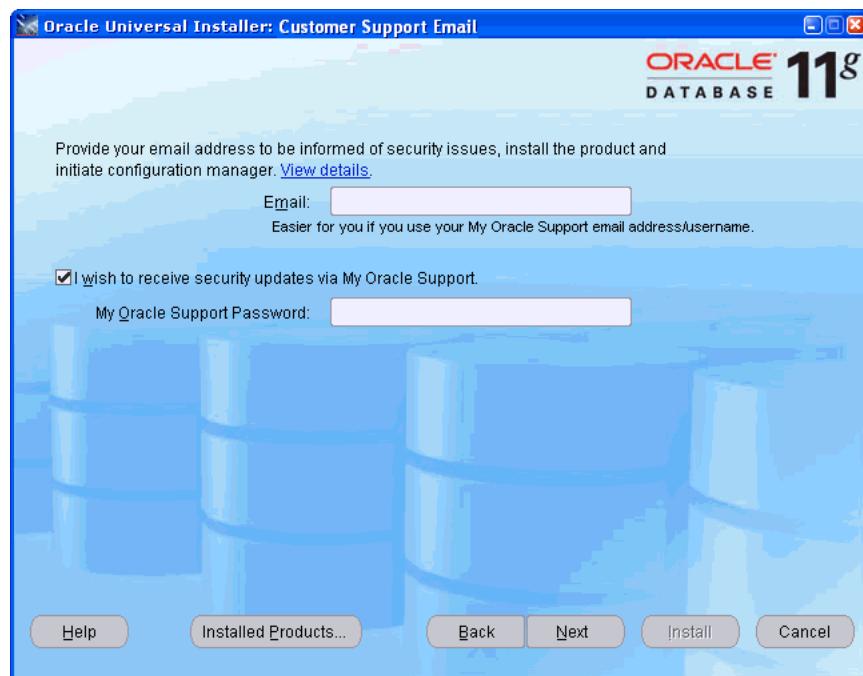


Tip: To ensure a successful installation:

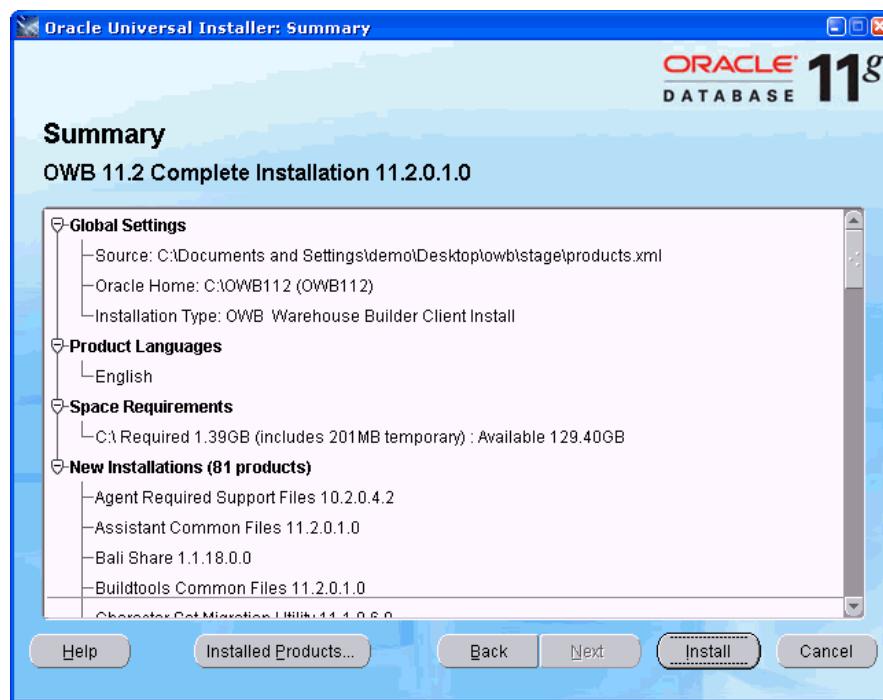
- Do not use an existing Oracle home.
- Do not use any spaces in the directory path.

5. [Optional] On the Customer Support Email screen, enter the **Email** and **Password** for your My Oracle Support account.

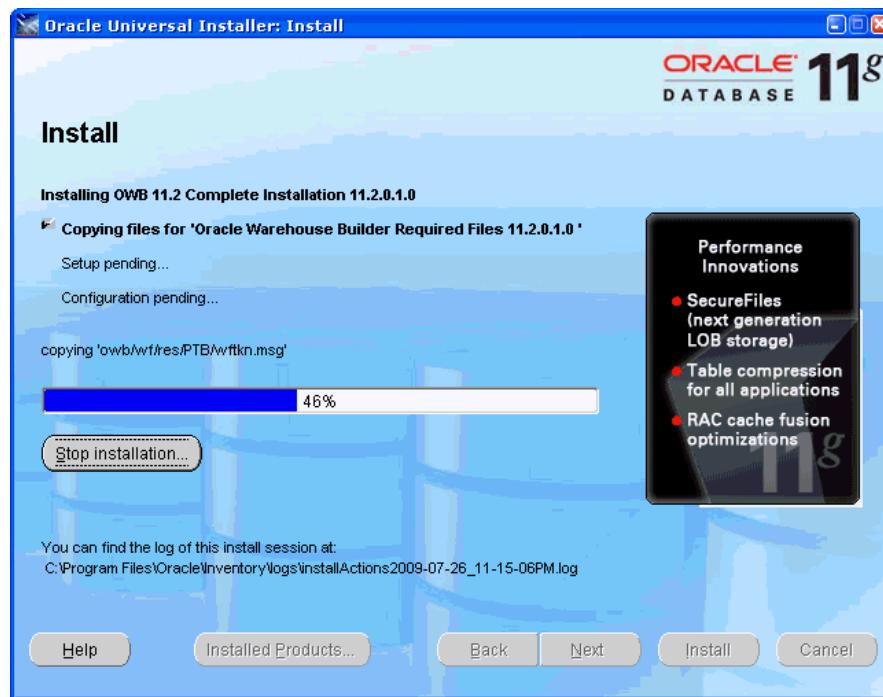
Click **Next**.



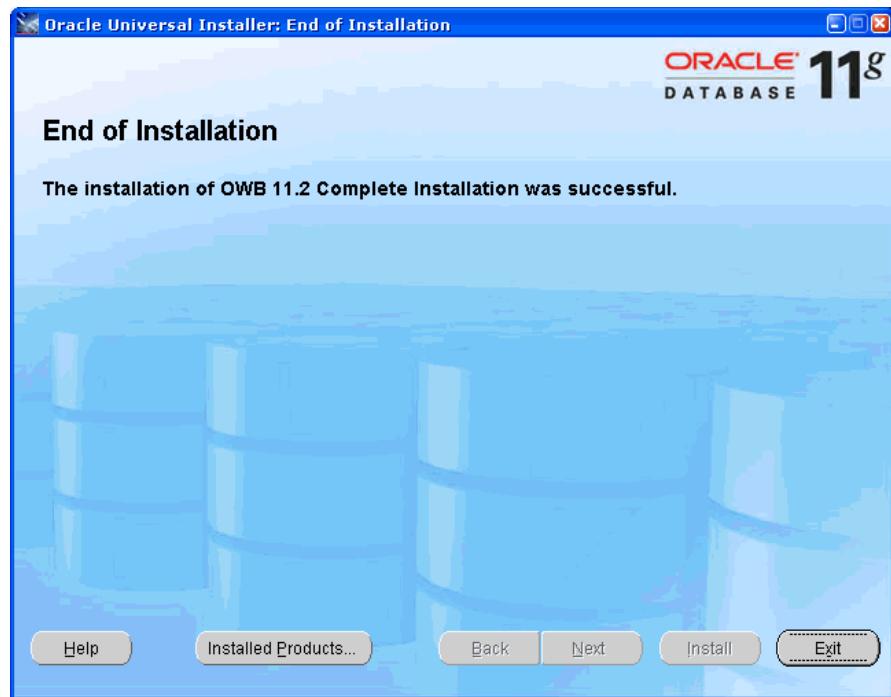
6. On the Summary screen, click **Install**.



7. The Install screen shows progress during the installation process.



8. At the End of Installation screen, click **Exit**. The Exit message appears asking if you really want to exit.



9. In the Exit message, click Yes.



Installing Warehouse Builder in Oracle RAC Environments

The overall process for installing on an Oracle RAC environment is similar to the general process for installing a single-instance database. However, there are a few specific details to observe as noted in this section.

To install and configure Warehouse Builder for Oracle RAC:

1. Create the Oracle RAC environment as described in *Oracle Real Application Clusters Installation Guide* specific to your platform. Cross references to generic installation guides are given after the last step of this procedure.
2. If you have not done so, review the most recent *Oracle Warehouse Builder Release Notes* available with the installation media.

See Also:

- Note ID 810394.1 "Oracle RAC and Oracle Clusterware Starter Kit and Best Practices" on My Oracle Support at
<https://support.oracle.com>
- Oracle Database release 11.2 documentation library page at
<http://www.oracle.com/pls/db112/homepage>
- The product download page on at
<http://www.oracle.com/technology/software/products/warehouse/index.html>

3. Prepare the host computers.

For each computer to host Warehouse Builder components, configure the `tnsnames.ora` file located in the `OWB_ORACLE_HOME\owb\network\admin` directory.

4. Configure `tnsnames.ora` for each Oracle Database server to be a Warehouse Builder data source or target. If you fail to configure `tnsnames.ora` for any host or database server, then you may encounter a repository connection error.

See Also: "[ORA-12154: TNS: Could not resolve service name.](#)" on page A-9 for troubleshooting

5. (Optional.) If necessary, install the Warehouse Builder standalone software. However, this is generally not necessary because Warehouse Builder is installed as part of every Oracle Database 11g Release 2 installation.
6. Define users and an owner for the repository. The Repository Assistant prompts you and guides you with procedures.

To start the Repository Assistant on Windows, from the Windows **Start** menu, select **Programs** and navigate to the Oracle product group you installed in the previous step. Select **Warehouse Builder**, **Administration**, and then **Repository Assistant**.

To start the Repository Assistant on UNIX, locate `OWB_ORACLE_HOME/owb/bin/` and run `reposinst.sh`.

7. Register each Oracle RAC node.

For each node, start the Repository Assistant and select the **Advanced Set Up** option. Connect to the node using `Host:Port:Sid`, where `Host` is the physical node name. Select the option for registering the Oracle RAC instance.

8. If the software is installed on separate disks, copy `rtrepos.properties` to each node in the cluster.

If you did not install to a shared disk, then you must manually copy the file `OWB_ORACLE_HOME/owb/bin/admin/rtrepos.properties` from the primary node to each node in the cluster.

9. Install the Warehouse Builder software on the client computers.

Repeat the procedure in ["Installing Warehouse Builder on Oracle Database 11g Release 2"](#) on page 2-1 for each computer to be used as a client.

10. After completing the installation process, you can start all the Warehouse Builder components.

Because you can connect to Warehouse Builder repositories using a net service name, you can embed Oracle RAC properties into the connect string to use Oracle RAC capabilities such as connect time failover, load balancing on server and load balancing of connections.

11. Refer to ["Ensuring the Availability of Service Names for Oracle RAC Nodes"](#) on page 2-11

The Control Center Service requires that service names for the individual nodes in the cluster be available. If these are not present after the Oracle RAC installation, then you must manually ensure the availability.

See Also:

- [Oracle® Real Application Clusters Installation Guide for Linux and UNIX](#)
- [Oracle® Real Application Clusters Installation Guide for Microsoft Windows x64 \(64-Bit\)](#)
- [Oracle Universal Installer and OPatch User's Guide for Windows and UNIX](#)

Installing Warehouse Builder on Each Node of a Cluster

Whether you are installing Warehouse Builder components onto a server or a client computer, you use the Oracle Universal Installer to install Warehouse Builder components.

For Oracle RAC, it is recommended that you install the Warehouse Builder components on each node of the cluster. The Control Center Service is required on to each node of the Oracle RAC cluster. You can achieve this in a single installation of the Warehouse Builder software if you install on a shared disk such as an OCFS or NTS shared disk.

Before launching Oracle Universal Installer, review and complete the ["Checklist: Before Using the Universal Installer in an Oracle RAC Environment"](#) on page 2-9.

Checklist: Before Using the Universal Installer in an Oracle RAC Environment

This section contains additional points to address before launching the Universal Installer:

- If you have not done so, review the latest Oracle Warehouse Builder Release Notes at <http://www.oracle.com/technology/>.
- The installed location must be the same directory path if using separate OWB_ORACLE_HOME installed disks, that is, local physical disks on each server.
- Close all other open applications.

For Windows Users Installing in an Oracle RAC Environment

To run the Oracle Universal Installer in an Oracle RAC environment:

1. Ensure that you are logged on to your system as a member of the Administrators group.

2. Insert the Oracle Warehouse Builder CD-ROM.
3. If your computer supports the autorun feature, then the autorun window automatically starts the Oracle Warehouse Builder installation.

If your computer does not support the autorun feature, locate the executable setup.exe in the root directory of the CD-ROM or downloaded software. Start the installer by launching the setup.exe program.
4. When prompted to specify the cluster node, you can select all hosts or the local node.

If you select local node, then you must install Warehouse Builder separately for each system unless installing to a shared disk.
5. When prompted, specify a home directory to be used only for the Warehouse Builder installation.

For example, specify a path such as C:\oracle\owb11_1 .
For the sake of brevity, the path you specify in this step is referred to as OWB_ORACLE_HOME throughout this guide.
6. Follow the on screen instructions.

When the software installation completes successfully, you can continue with the next step in "[Installing Warehouse Builder in Oracle RAC Environments](#)" on page 2-7.

For UNIX Users Installing in an Oracle RAC Environment

You can run Oracle Universal Installer from the CD-ROM. Do not run the Installer while the CD-ROM directory is the current directory because you cannot later unmount the next CD-ROM when prompted to do so.

To run the Oracle Universal Installer in an Oracle RAC environment:

1. To enable clusterware installation, ensure that you run an interactive secure shell such as /bin/ssh and have host user equivalency to all nodes.
2. Log in as the operating system user of the Oracle Database.

For example, log in as the oracle user.
Be sure you are not logged in as the root user when you start the Oracle Universal Installer (OUI). If you log in as root user when you start OUI, then only the root user has permissions to manage Oracle Warehouse Builder.
3. Start the installer by entering the following at the prompt:

```
cd mount_point  
.runInstaller
```
4. When prompted to specify the cluster node, you can select all hosts or the local node.

If you select local node, then you must install Warehouse Builder separately for each system unless installing to a shared disk.
5. As the installation proceeds, the Oracle Universal Installer prompts you to run several scripts. You must switch users and run the script as root.

When the software installation completes successfully, you can continue with the next step in "[Installing Warehouse Builder on Oracle Database 11g Release 2](#)" on page 2-1.

Ensuring the Availability of Service Names for Oracle RAC Nodes

1. List all of the instance or node names in the cluster. Issue the following command:

```
srvctl config database -d dbname
```

where dbname is the unique database name as specified by the initialization parameter db_name.

2. For a given instance, instn, add a service with the following command:

```
srvctl add service -d dbname -s instn -r instn
```

The resulting service name is instn.clusterdomainname. For example, if the instance name is owbrac1, then the service name could be owbrac1.us.oracle.com.

3. For a given instance, instn, start the service with the following command:

```
srvctl start service -d dbname -s instn
```

4. For a given instance, instn, verify the service is running with the following command:

```
srvctl status service -d dbname -s instn
```

5. Complete steps 2 through 4 for each node.

Performing a Silent Install and Using Response Values

An Oracle Universal Installer silent installation runs in the background and does not require input from the user. Instead of prompting the user to select a series of installation options, the software uses a pre-defined set of options stored in a response file or passed on the command line.

A response file specifies the information normally provided by the user to Oracle Universal Installer interface during an interactive installation session. Each answer is stored as a value for a variable identified in the response file. For example, values for Oracle home or install type can be set automatically within the response file. The response file template for the installation of your product can be found on your stage (CD-ROM) under the *root_of_CD/response* directory, such as:

Products.xml_Location/Response/product.installtype.rsp

To perform a silent installation of Warehouse Builder

- Open a Command window and navigate to the directory containing the response file that is used for the silent installation. Use the following command to start the silent installation:

For Windows, enter `setup -responseFile filename optional_parameters`

For Linux, enter `runInstaller -responseFile filename optional_parameters`

For this procedure, *filename* represents the name of the response file (including the complete path) and *optional_parameters* represents the parameters used for the silent installation.

After the Installation is Complete

Once Oracle Warehouse Builder software is installed on your computer, you must configure it.

For steps on configuring Oracle Warehouse Builder on *Linux*, see [Chapter 4, "Configuring Oracle Warehouse Builder for Linux"](#).

For steps on configuring Oracle Warehouse Builder on *Microsoft Windows*, see [Chapter 3, "Configuring Oracle Warehouse Builder for Windows"](#).

Configuring Oracle Warehouse Builder for Windows

This section discusses how to configure Oracle Warehouse Builder on Windows platforms.

This section includes the following topics:

- [Configuring OWB on Windows Installed as Part of Oracle Database 11g Release 2](#)
- [Configuring OWB on Windows Installed as a Standalone Instance](#)
- [Creating the First Workspace in the Repository on Windows](#)

Note: The initial installation of Oracle Warehouse Builder must be made on the computer that hosts the OWB Repository.

Configuring OWB on Windows Installed as Part of Oracle Database 11g Release 2

Oracle Warehouse Builder is part of every database installation on hosts running Oracle Database 11g Release 2.

Once Oracle Database 11g Release 2 is installed, you only have to unlock two accounts: OWBSYS and OWBSYS_AUDIT.

There are no additional installation tasks. However, you must configure Oracle Workflow to use process flows in Warehouse Builder.

See Also: "Integrating with Oracle Workflow" on page 7-28

To unlock OWBSYS and OWBSYS_AUDIT accounts:

1. At the command prompt, start SQL*Plus:

```
C:\>sqlplus
SQL*Plus: Release 11.2 - Production on Mon Jul 27 11:34:06 2009
```

```
Copyright (c) 1982, 2009, Oracle. All rights reserved.
```

2. When prompted for user name, log in as a user with administrative privileges.

```
Enter user-name: sys as sysdba
Enter password: password
```

```
Connected to:
Oracle Database 11g Enterprise Edition Release 11.2 - Production
```

With the Partitioning, OLAP and Data Mining options

3. Unlock the OWBSYS user account, and set its new password:

```
SQL> alter user owbsys identified by password account unlock;
```

User altered.

4. Unlock the OWBSYS_AUDIT user account, and set its new password:

```
SQL> alter user owbsys_audit identified by password account unlock;
```

User altered.

Configuring OWB on Windows Installed as a Standalone Instance

This section contains the following topics:

- [About Configuring OWB Installed as a Standalone Instance](#)
- [Cleaning an Oracle Warehouse Builder Repository](#)
- [Creating an OWBSYS Schema with Repository Objects](#)
- [Configuring the Control Center Service](#)
- [Unlocking the OWBSYS and OWBSYS_AUDIT Accounts](#)
- [Setting the OWB Home for Remote Installations](#)

About Configuring OWB Installed as a Standalone Instance

To properly configure Oracle Warehouse Builder and its repository on a host that runs an earlier version of the database, such as Oracle Database 10g Release 2, you must:

- Ensure that the database is fully installed and working; if it is not, see *Oracle Database Installation Guide for Microsoft Windows*.
- Note the host name, the port, and Oracle service name of this database. The example here uses the following connection information:
`localhost:1521:orcl`.
- Ensure that you have the username and password for an account with SYSDBA privileges. The SYSDBA privilege is necessary for making changes to the basic configuration of Oracle Database, Oracle Warehouse Builder, and OWB Repository.

To create an Oracle Warehouse Builder repository in your database, you must first create an OWBSYS schema, and then install repository objects within it. Oracle Warehouse Builder has command-level scripts to perform this task:

- The `clean_owbsys.sql` script drops the contents of any existing OWBSYS schema installations, but leaves the schema otherwise intact.
- The `cat_owb.sql` script checks if an OWBSYS schema exists, and creates it if necessary, and then installs the objects required by Oracle Warehouse Builder 11.2 repository.
- The `reset_owbcc_home.sql` script ensures that Oracle Warehouse Builder uses the 11.2 version of the Control Center Service.
- [Optional] The `remote_owb_install.sql` script, when executed on the server, ensures that a repository on a remote system can be installed; it sets the `REMOTE_OWB_HOME` variable for authentication by remote systems.

Starting with Oracle Warehouse Builder 11.1, the metadata repository is stored in the OWBSYS schema, inside a workspace. This contrasts with the terminology of Oracle Warehouse Builder 10.2, where a workspace corresponds to a repository.

Oracle Warehouse Builder repositories for version 11.2 and 11.1 cannot coexist on the same database; creating an 11.2 repository overwrites the contents of the 11.1 repository in the database. Oracle recommends that you back up the older OWB repositories before installing the 11.2 repository or upgrading to the 11.2 repository.

Cleaning an Oracle Warehouse Builder Repository

To clean an existing Oracle Warehouse Builder repository:

1. On the Microsoft Windows desktop, click **Start** and select **Run**.

2. Start SQL*Plus:

```
C:>sqlplus
SQL*Plus: Release 11.2 - Production on Mon Jul 27 12:05:12 2009

Copyright (c) 1982, 2009, Oracle. All rights reserved.
```

3. When prompted for user name, log in as `owbsys` user.

```
Enter user-name: owbsys
Enter password: password

Connected to:
Oracle Database 11g Enterprise Edition Release 11.2 - Production
With the Partitioning, OLAP and Data Mining options
```

4. Stop Oracle Warehouse Builder Control Center service.

```
SQL> @c:\OWB112\owb\rtp\sql\stop_service.sql
```

See "[stop_service](#)" on page 9-12 for more information.

5. Connect as user with administrative privileges:

```
SQL> connect sys as sysdba
Enter password: password
```

6. Run the `clean_owbsys.sql` script to drop all the objects in the existing OWBSYS schema:

```
SQL> @c:\OWB112\owb\UnifiedRepos\clean_owbsys.sql
```

Creating an OWBSYS Schema with Repository Objects

The process of creating (or refreshing) an OWBSYS schema generates the corresponding locked user account within the specified tablespace; in this example, the tablespace is USERS. When the script completes, you must unlock the account and set its password. The instructions in "[Configuring OWB on Windows Installed as Part of Oracle Database 11g Release 2](#)" on page 3-1 show how to unlock OWBSYS and OWBSYS_AUDIT accounts. Note that the OWBSYS schema is populated only when you run the Repository Assistant for the first time and create the first workspace in the repository.

See Also: "[Configuring OWB for Oracle RAC](#)" on page A-14

To set-up an OWBSYS schema with repository objects:

1. On the Microsoft Windows desktop, click **Start** and select **Run**.

2. At the command prompt, start SQL*Plus:

```
C:\>sqlplus
SQL*Plus: Release 11.2 - Production on Mon Jul 27 12:18:03 2009
```

```
Copyright (c) 1982, 2009, Oracle. All rights reserved.
```

3. When prompted for user name, log in as a user with administrative privileges.

```
Enter user-name: sys as sysdba
Enter password: password
```

```
Connected to:
```

```
Oracle Database 11g Enterprise Edition Release 11.2 - Production
With the Partitioning, OLAP and Data Mining options
```

4. Run the `cat_owb.sql` script to set up OWBSYS in a tablespace in your database. For example, this command sets up OWBSYS in the USERS tablespace:

```
SQL> @C:\OWB112\owb\UnifiedRepos\cat_owb.sql USERS
```

Configuring the Control Center Service

If you are installing OWB 11.2 on an older Oracle Database release, or if you integrating OWB with Oracle Business Intelligence Standard Edition (Oracle Discoverer), you must run the `reset_owbcc_home.sql` script to ensure that the Control Center Service functions correctly.

To configure the Control Center service:

1. In Windows Explorer, open the directory for the Oracle home of Oracle Warehouse Builder release 11.2 to confirm the correct path for the Oracle home, paying particular attention to capitalization. In this example, the path is C:\OWB112.

While Windows is generally case insensitive, the path entered in the next steps must be an exact match to Oracle Warehouse Builder home directory, as displayed in the Windows Explorer address bar. To avoid errors, copy and paste the path.

2. On the Microsoft Windows desktop, click **Start** and select **Run**.
3. At the command prompt, start SQL*Plus:

```
C:\>sqlplus
SQL*Plus: Release 11.2 - Production on Mon Jul 27 12:31:53 2009
```

```
Copyright (c) 1982, 2009, Oracle. All rights reserved.
```

4. When prompted for user name, log in as a user with administrative privileges.

```
Enter user-name: sys as sysdba
Enter password: password
```

```
Connected to:
```

```
Oracle Database 11g Enterprise Edition Release 11.2 - Production
With the Partitioning, OLAP and Data Mining options
```

5. Run the `reset_owbcc_home.sql` script to force the Control Center service to run from Oracle Warehouse Builder 11.2 installation:

```
SQL> @C:\OWB112\owb\UnifiedRepos\reset_owbcc_home.sql C:\OWB112
```

Unlocking the OWBSYS and OWBSYS_AUDIT Accounts

To unlock OWBSYS and OWBSYS_AUDIT accounts:

1. On the Microsoft Windows desktop, click **Start** and select **Run**.
2. At the command prompt, start SQL*Plus:

```
C:\>sqlplus
SQL*Plus: Release 11.2 - Production on Mon Jul 27 12:37:08 2009
```

Copyright (c) 1982, 2009, Oracle. All rights reserved.

3. When prompted for user name, log in as a user with administrative privileges.

```
Enter user-name: sys as sysdba
Enter password: password
```

Connected to:

Oracle Database 11g Enterprise Edition Release 11.2 - Production
With the Partitioning, OLAP and Data Mining options

4. Unlock the OWBSYS user account, and set its new password:

```
SQL> alter user owbsys identified by password account unlock;
```

User altered.

5. Unlock the OWBSYS_AUDIT user account, and set its new password:

```
SQL> alter user owbsys_audit identified by password account unlock;
```

User altered.

Configuring Oracle Workflow to Support OWB Process Flows

Using process flows requires that you configure Oracle Workflow for use with Warehouse Builder. See "[Integrating with Oracle Workflow](#)" on page 7-28 for details on configuring Oracle Workflow.

Setting the OWB Home for Remote Installations

You must create a database directory `OWB_REMOTE_ADMIN` that is accessed by remote installations, so they can read the server's `OWB_HOME/owb/bin/admin` directory.

1. On the Microsoft Windows desktop, click **Start** and select **Run**.
2. At the command prompt, start SQL*Plus:

```
C:\>sqlplus
SQL*Plus: Release 11.2 - Production on Mon Jul 27 12:42:27 2009
```

Copyright (c) 1982, 2009, Oracle. All rights reserved.

3. When prompted for user name, log in as a user with administrative privileges.

```
Enter user-name: sys as sysdba
Enter password: password
```

Connected to:

Oracle Database 11g Enterprise Edition Release 11.2 - Production
With the Partitioning, OLAP and Data Mining options

4. Set the *OWB_REMOTE_HOME* variable using script *remote_owb_install.sql*:

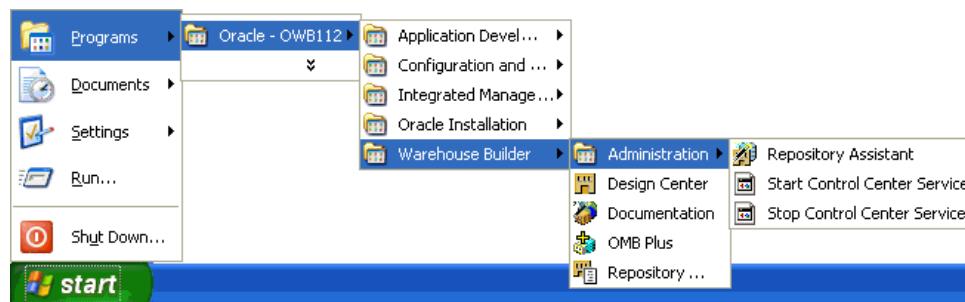
```
SQL> @remote_owb_install.sql OWB_REMOTE_HOME
```

Creating the First Workspace in the Repository on Windows

To use a repository, you must define an initial workspace, and the owner of that workspace. The following instructions use the Repository Assistant.

To create a workspace:

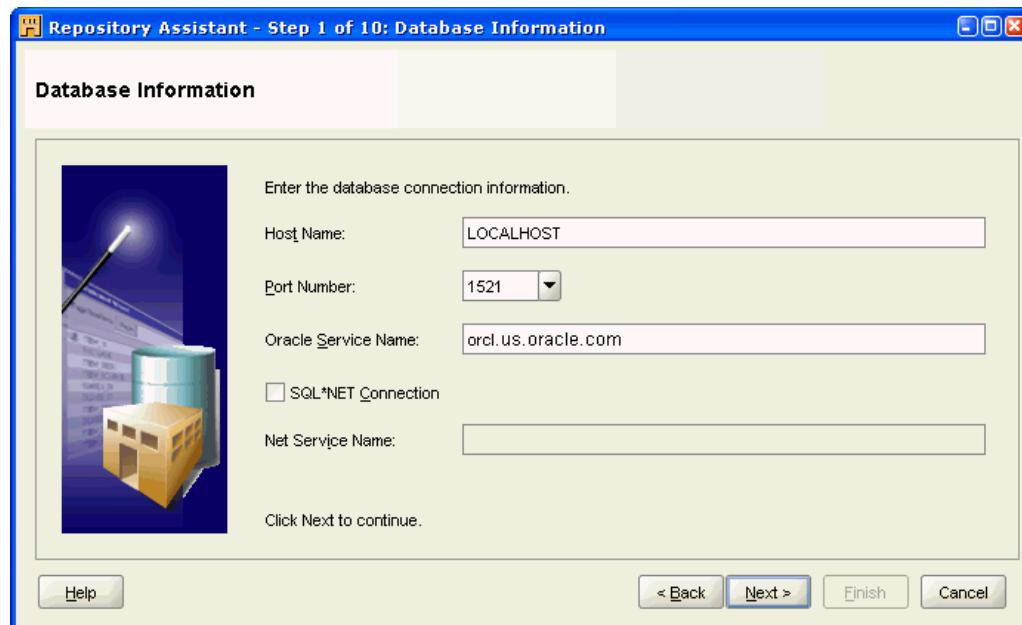
1. Start the repository assistant from the **Start** menu, selecting **Program Files**, then **Oracle - OWB112**, then **Warehouse Builder**, then **Administration**, then **Repository Assistant**.



2. On the Welcome screen of the Repository assistant, click **Next**.



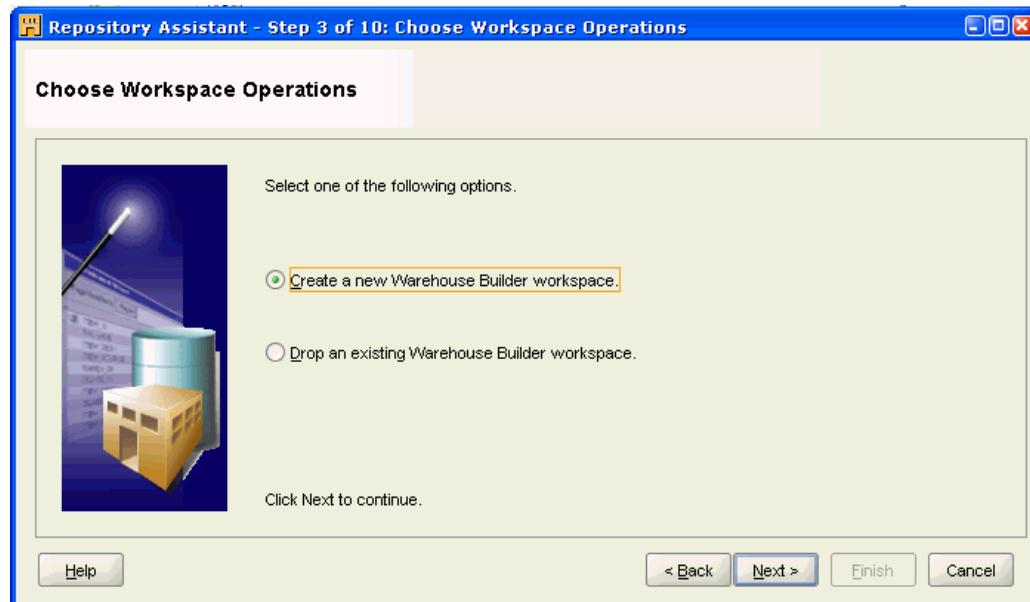
3. On the Database Information screen, enter the connection information (Host Name, Port Number, and Oracle Service Name) for the repository database, and click **Next**.



4. On the Choose Operation screen, select **Manage Warehouse Builder Workspaces**. Click Next.



5. On the Choose Workspace Operations screen, select **Create a New Warehouse Builder workspace**. Click Next.



6. On the New or Existing User screen, select **Create a workspace with a new workspace owner**. Click **Next**.

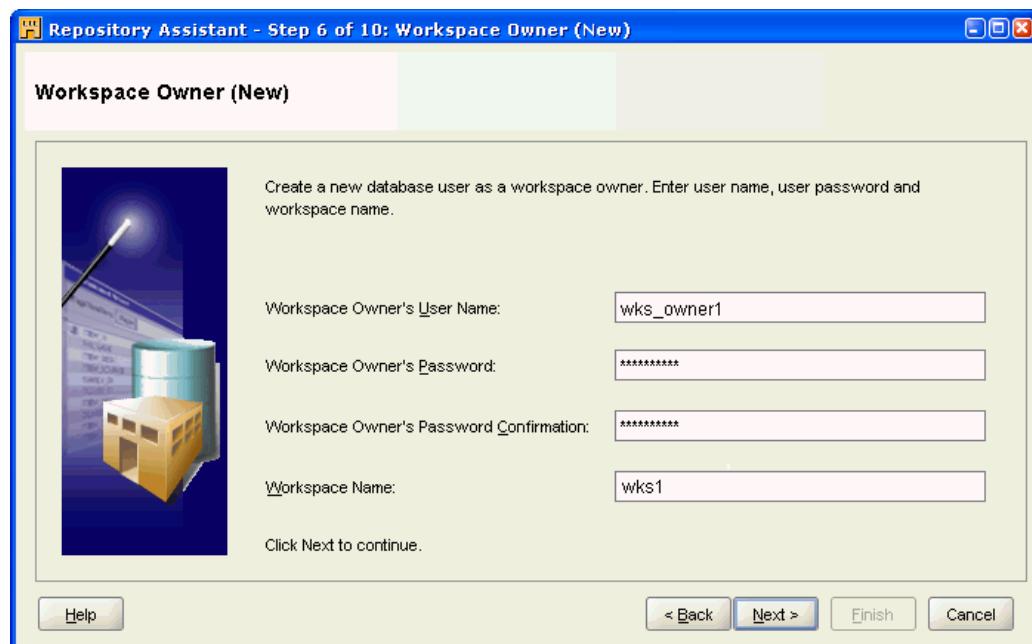


7. On the DBA Information screen, enter the **User Name** of a user with a **SYS** privilege, and that user's **Password**.

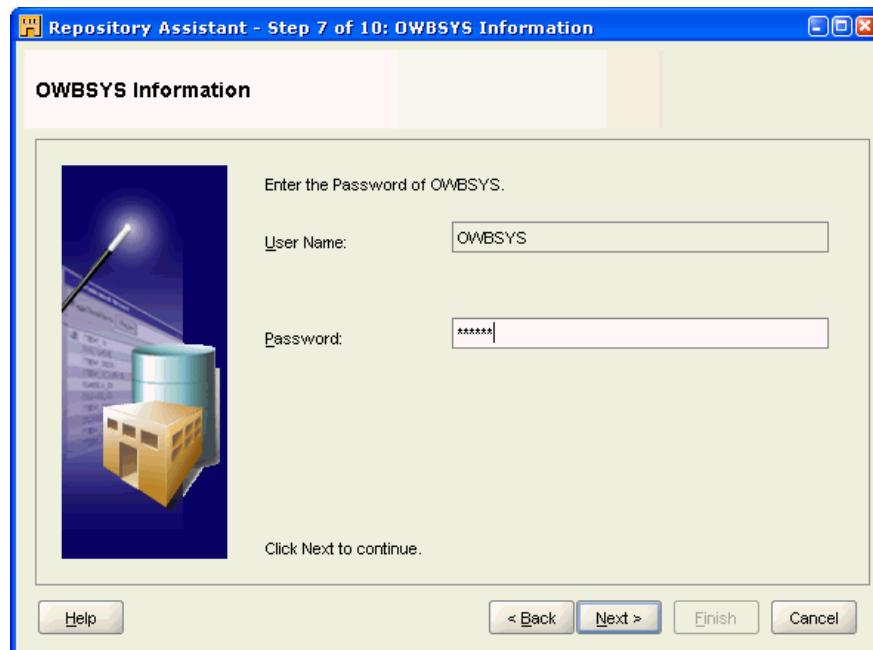
Click **Next**.



- 8.** In the Workspace Owner (New) screen, enter the following:
- Workspace Owner's User Name (wks_owner1)
 - Workspace Owner's Password
 - Workspace Owner's Password Confirmation
 - Workspace Name (wks1) (Note that workspace names cannot contain spaces)
- Click **Next**.



- 9.** On the OWBSYS Information screen, enter the OWBSYS **Password**.
Click **Next**.



10. [Optional] For remote installations (when you the schema is on a remote system), the system performs the verification of versions, and displays the dialog "Found OWB software with the compatible version installed locally on the database server computer with path REMOTE_OWB_HOME."

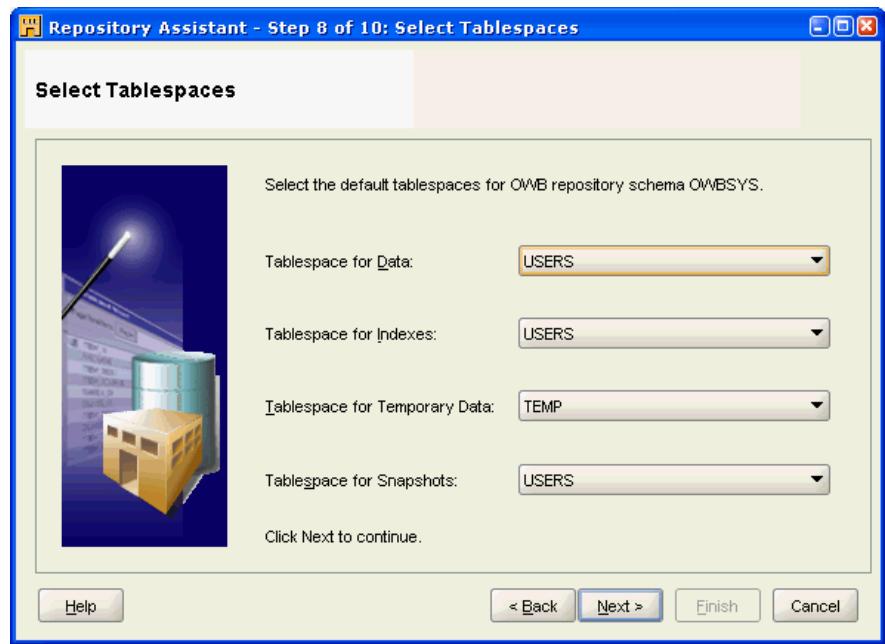
If the version is either incompatible or cannot be found, then the system displays the dialog "OWB software with the compatible version must be installed locally on the database server computer". If it is installed, acquire the DBA user credentials, and run the `remote_owb_install.sql` script on the Oracle Database server computer, as described in "[Setting the OWB Home for Remote Installations](#)" on page 3-5.

11. On the Select Tablespaces screen, enter the following:

- Tablespace for Data (USERS, or the name of the desired tablespace)
- Tablespace for Indexes (USERS, or the name of the same tablespace)
- Tablespace for Temporary Data (TEMP)
- Tablespace for Snapshots (USERS, or the name of the same tablespace)

Click **Next**.

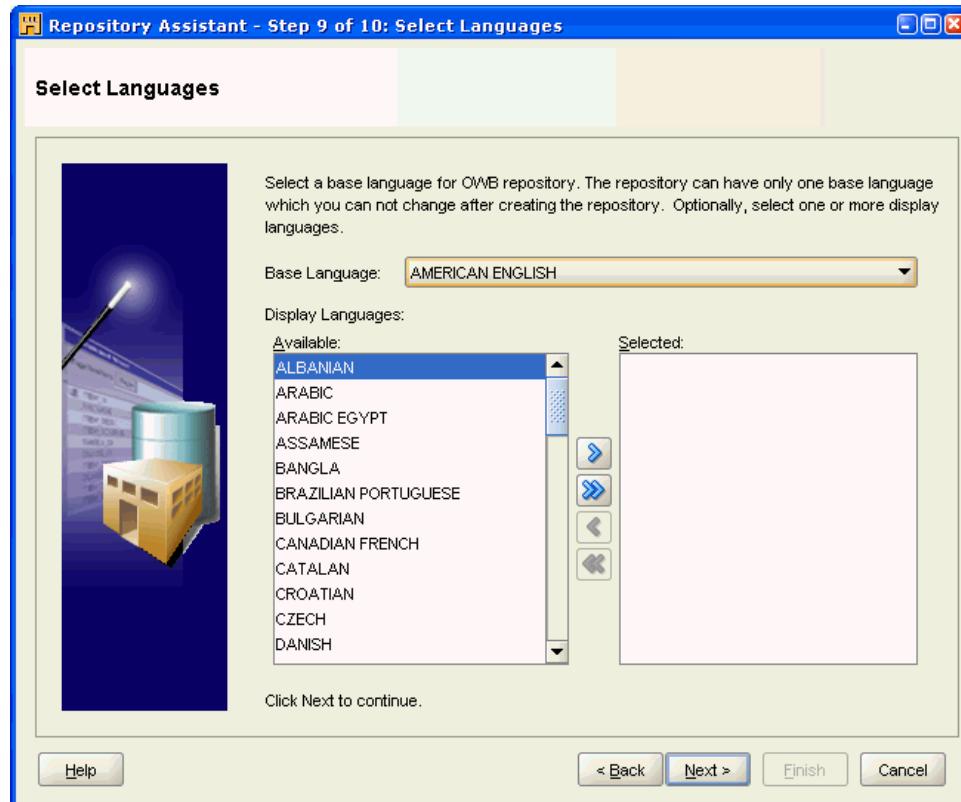
Note that this step enables you to change the default tablespace details. The OWB repository is stored in the OWBSYS schema, but you must identify what tablespaces are used to store OWBSYS and certain other objects, like snapshots. You may choose tablespaces other than USERS, depending on your storage management strategy.



12. On the Select Languages screen, select the Base Language from the menu.

[Optional] You may also select additional Display Languages from the list of Available languages on the left-hand side, and move them to the list of Selected languages on the right side.

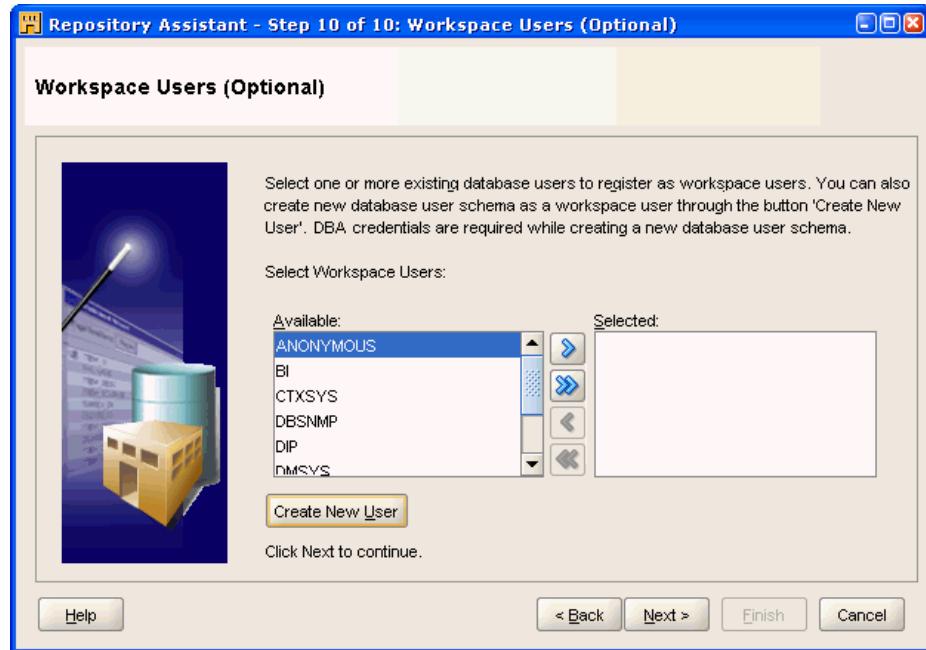
Click Next.



13. On the Workspace Users (Optional) screen, select the appropriate existing users from the **Available** list add them to the **Selected** list.

[Optional] Click **Create New User** to make new users who can access OWB. This takes you to step 14.

Click **Next**. This takes you to step 16.



14. [Optional] In the Create New Database User screen, enter the following:

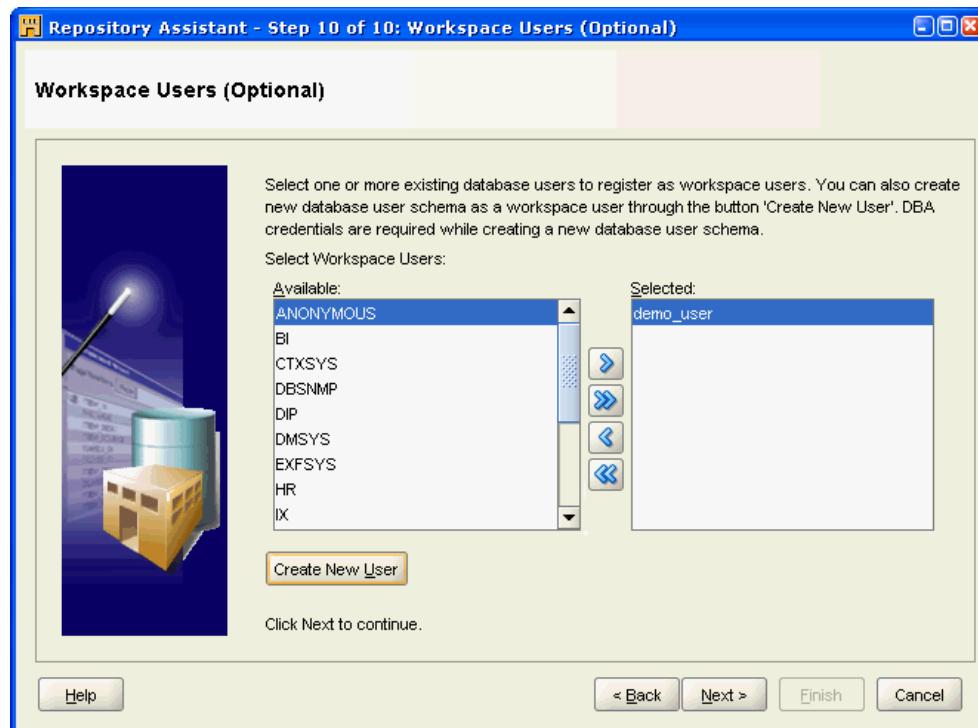
- User Name (demo_user)
- Password
- Re-enter Password

Click **OK**.

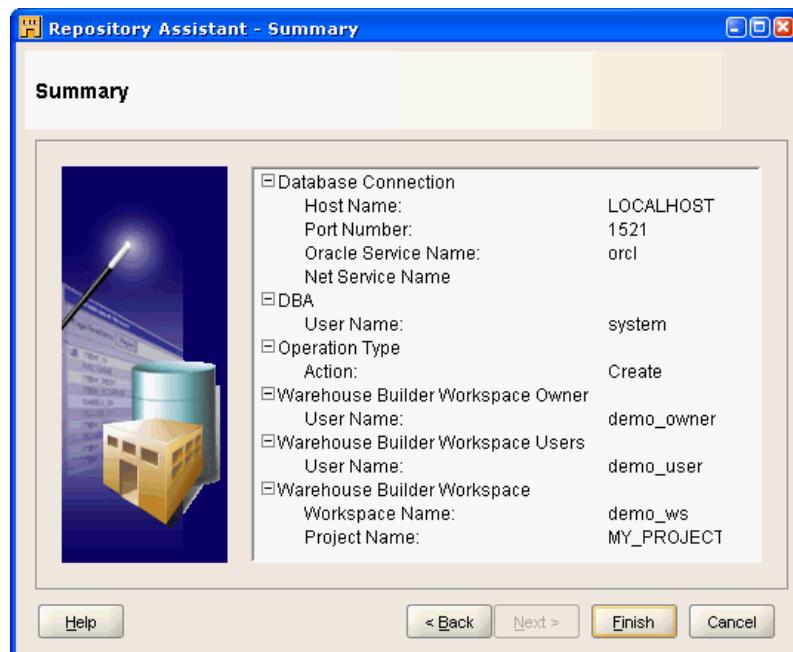
Note that DBA User Name (`system`) and DBA User Password are disabled fields and cannot be edited.



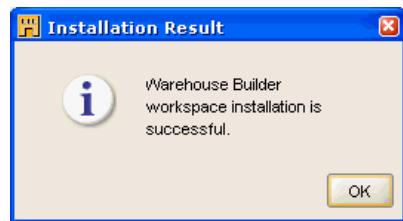
15. In the Workspace Users (Optional) screen, click **Next**.



16. On the Summary screen, click **Finish**.



17. Once the Installation Result screen appears, click **OK**.



4

Configuring Oracle Warehouse Builder for Linux

This section discusses how to configure Oracle Warehouse Builder on Linux platforms.

This section includes the following topics:

- [Configuring OWB on Linux Installed as Part of Oracle Database 11g Release 2](#)
- [Configuring OWB on Linux Installed as a Standalone Instance](#)
- [Creating the First Workspace in the Repository on Linux](#)

Note: The initial installation of Oracle Warehouse Builder must be made on the computer that hosts the OWB Repository.

Configuring OWB on Linux Installed as Part of Oracle Database 11g Release 2

Oracle Warehouse builder is part of every database installation on hosts running Oracle Database 11g Release 2.

Once Oracle Database 11g Release 2 is installed, you only have to unlock two accounts: OWBSYS and OWBSYS_AUDIT.

There are no additional installation tasks. However, you must configure Oracle Workflow if you want to use process flows.

See Also: ["Integrating with Oracle Workflow" on page 7-28](#)

To unlock OWBSYS and OWBSYS_AUDIT accounts:

1. At the command prompt, start SQL*Plus:

```
% sqlplus  
SQL*Plus: Release 11.2 - Production on Mon July 27 13:06:27 2009
```

```
Copyright (c) 1982, 2009, Oracle. All rights reserved.
```

2. When prompted for user name, log in as a user with administrative privileges.

```
Enter user-name: sys as sysdba  
Enter password: password
```

```
Connected to:  
Oracle Database 11g Enterprise Edition Release 11.2 - Production  
With the Partitioning, OLAP and Data Mining options
```

3. Unlock the OWBSYS user account, and set its new password:

```
SQL> alter user owbsys identified by password account unlock;
```

User altered.

4. Unlock the OWBSYS_AUDIT user account, and set its new password:

```
SQL> alter user owbsys_audit identified by password account unlock;
```

User altered.

Configuring OWB on Linux Installed as a Standalone Instance

This section contains the following topics:

- ["About Configuring OWB Installed as a Standalone Instance" on page 4-2](#)
- ["Cleaning an Oracle Warehouse Builder Repository" on page 4-3](#)
- ["Creating an OWBSYS Schema with Repository Objects" on page 4-3](#)
- ["Configuring the Control Center Service" on page 4-4](#)
- ["Unlocking the OWBSYS and OWBSYS_AUDIT Accounts" on page 4-4](#)

About Configuring OWB Installed as a Standalone Instance

To properly configure Oracle Warehouse Builder and its repository on a host that runs an earlier version of the database, such as Oracle Database 10g Release 2, you must:

- Ensure that the database is fully installed and working; if you it is not, see *Oracle Database Quick Installation Guide for Linux x86-64*.
- Note of the host name, the port, and Oracle service name of this database. The example here uses the following connection information:
`localhost:1521:orcl`.
- Ensure that you have username and password for an account with SYSDBA privileges. The SYSDBA privilege has the necessary privileges for making changes to the basic configuration of Oracle Database, Oracle Warehouse Builder, and OWB Repository.

To create an Oracle Warehouse Builder repository in your database, you must first create an OWBSYS schema, and then install repository objects within it. Oracle Warehouse Builder has command-level scripts to perform this task:

- The `clean_owbsys.sql` script drops the contents of any existing OWBSYS schema installations, but leaves the schema otherwise intact.
- The `cat_owb.sql` script checks if an OWBSYS schema exists, and creates it if necessary, and then installs the objects required by Oracle Warehouse Builder 11.2 repository.
- The `reset_owbcc_home.sql` script ensures that Oracle Warehouse Builder uses the 11.2 version of the Control Center Service.
- [Optional] The `remote_owb_install.sql` script, when executed on the server, ensures that a repository on a remote system can be installed; it sets the `REMOTE_OWB_HOME` variable for authentication by remote systems.

Starting with Oracle Warehouse Builder 11.1, the metadata repository is stored in the OWBSYS schema, inside a workspace. This contrasts with the terminology of Oracle Warehouse Builder 10.2, where a workspace corresponds to a repository.

Oracle Warehouse Builder repositories for version 11.2 and 11.1 cannot coexist on the same database; creating an 11.2 repository overwrites the contents of the 11.1 repository in the database. Oracle recommends that you back up the older OWB repositories before installing the 11.2 repository or upgrading to the 11.2 repository.

Cleaning an Oracle Warehouse Builder Repository

To clean an existing Oracle Warehouse Builder repository:

- At the command prompt, start SQL*Plus:

```
% sqlplus
SQL*Plus: Release 11.2 - Production on Mon July 27 13:14:43 2009
Copyright (c) 1982, 2009, Oracle. All rights reserved.
```

- When prompted for user name, log in as OWBSYS user.

```
Enter user-name: owbsys
Enter password: password

Connected to:
Oracle Database 11g Enterprise Edition Release 11.2 - Production
With the Partitioning, OLAP and Data Mining options
```

- Stop Oracle Warehouse Builder Control Center service.

```
SQL> @OWB112/owb/rtp/sql/stop_service.sql
```

See "[stop_service](#)" on page 9-12 for more information.

- Connect as user with administrative privileges:

```
SQL> connect sys as sysdba
Enter password: password
```

- Run the `clean_owbsys.sql` script to drop all the objects in the existing OWBSYS schema:

```
SQL> @OWB112/owb/UnifiedRepos/clean_owbsys.sql
```

Creating an OWBSYS Schema with Repository Objects

The process of creating (or refreshing) an OWBSYS schema generates the corresponding locked user account within the specified tablespace. In this example, the tablespace is USERS. When the script completes, you must unlock the account and set its password. The instructions in "["Configuring OWB on Linux Installed as Part of Oracle Database 11g Release 2"](#) on page 4-1 show how to unlock OWBSYS and OWBSYS_AUDIT accounts. Note that the OWBSYS schema is populated only when you run the Repository Assistant for the first time and create the first workspace in the repository.

See Also: "[Configuring OWB for Oracle RAC](#)" on page A-14

To set-up an OWBSYS schema with repository objects:

- At the command prompt, start SQL*Plus:

```
% sqlplus
```

```
SQL*Plus: Release 11.2 - Production on Mon July 27 13:22:11 2009
```

```
Copyright (c) 1982, 2009, Oracle. All rights reserved.
```

2. When prompted for user name, log in as a user with administrative privileges.

```
Enter user-name: sys as sysdba  
Enter password: password
```

```
Connected to:
```

```
Oracle Database 11g Enterprise Edition Release 11.2 - Production  
With the Partitioning, OLAP and Data Mining options
```

3. Run the `cat_owb.sql` script to set up OWBSYS in a tablespace in your database. For example, this command sets up OWBSYS in the USERS tablespace:

```
SQL> @OWB112/owb/UnifiedRepos/cat_owb.sql USERS
```

Configuring the Control Center Service

If you are installing OWB 11.2 on an older Oracle Database release, or if you integrating OWB with Oracle Business Intelligence Standard Edition (Oracle Discoverer), you must run the `reset_owbcc_home.sql` script to ensure that the Control Center Service functions correctly.

To configure the Control Center service:

1. Confirm the correct Oracle home path, paying particular attention to capitalization. In this example, the path is OWB112.

The path entered in the next steps must be an exact match to the Oracle Warehouse Builder home directory, including letter case match.

2. At the command prompt, start SQL*Plus:

```
% sqlplus  
SQL*Plus: Release 11.2 - Production on Mon July 27 13:47:31 2009
```

```
Copyright (c) 1982, 2009, Oracle. All rights reserved.
```

3. When prompted for user name, log in as a user with administrative privileges.

```
Enter user-name: sys as sysdba  
Enter password: password
```

```
Connected to:
```

```
Oracle Database 11g Enterprise Edition Release 11.2 - Production  
With the Partitioning, OLAP and Data Mining options
```

4. Run the `reset_owbcc_home.sql` script to force the Control Center Service to run from Oracle Warehouse Builder 11.2 installation:

```
SQL> @OWB112/owb/UnifiedRepos/reset_owbcc_home.sql OWB112
```

Unlocking the OWBSYS and OWBSYS_AUDIT Accounts

To unlock OWBSYS and OWBSYS_AUDIT accounts:

1. At the command prompt, start SQL*Plus:

```
% sqlplus  
SQL*Plus: Release 11.2 - Production on Mon July 27 13:57:03 2009
```

Copyright (c) 1982, 2009, Oracle. All rights reserved.

2. When prompted for user name, log in as a user with administrative privileges.

```
Enter user-name: sys as sysdba
Enter password: password
```

```
Connected to:
Oracle Database 11g Enterprise Edition Release 11.2 - Production
With the Partitioning, OLAP and Data Mining options
```

3. Unlock the OWBSYS user account, and set its new password:

```
SQL> alter user owbsys identified by password account unlock;
```

User altered.

4. Unlock the OWBSYS_AUDIT user account, and set its new password:

```
SQL> alter user owbsys_audit identified by password account unlock;
```

User altered.

Configuring Oracle Workflow to Support OWB Process Flows

Using process flows requires that you configure Oracle Workflow for use with Warehouse Builder. See "[Integrating with Oracle Workflow](#)" on page 7-28 for details on configuring Oracle Workflow.

Setting the OWB Home for Remote Installations

You must create a database directory *OWB_REMOTE_ADMIN* that is accessed by remote installations, so they can read the server's *OWB_HOME/owb/bin/admin* directory.

1. At the command prompt, start SQL*Plus:

```
% sqlplus
SQL*Plus: Release 11.2 - Production on Mon July 27 14:06:12 2009
```

Copyright (c) 1982, 2009, Oracle. All rights reserved.

2. When prompted for user name, log in as a user with administrative privileges.

```
Enter user-name: sys as sysdba
Enter password: password
```

```
Connected to:
Oracle Database 11g Enterprise Edition Release 11.2 - Production
With the Partitioning, OLAP and Data Mining options
```

3. Set the *OWB_REMOTE_HOME* variable using script *remote_owb_install.sql*:

```
SQL> @OWB112/owb/UnifiedRepos/remote_owb_install.sql OWB_REMOTE_HOME
```

Creating the First Workspace in the Repository on Linux

To use a repository, you must define an initial workspace, and the owner of that workspace. The following instructions use the Repository Assistant.

To create a workspace:

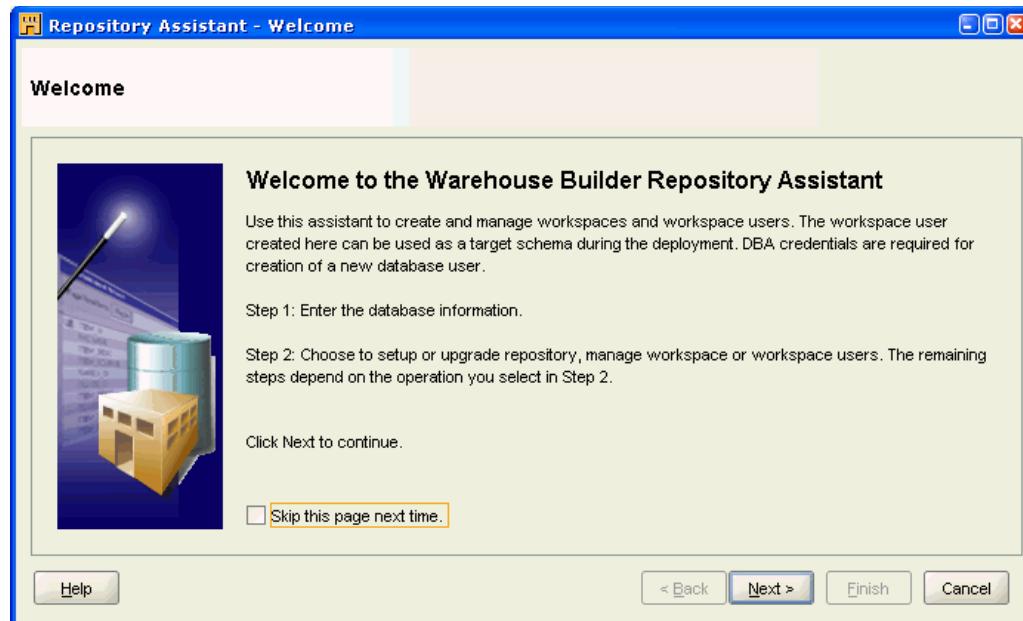
1. At the command prompt, navigate to the UNIX bin directory:

```
% cd OWB112/owb/bin/unix  
% pwd  
OWB112/owb/bin/unix
```

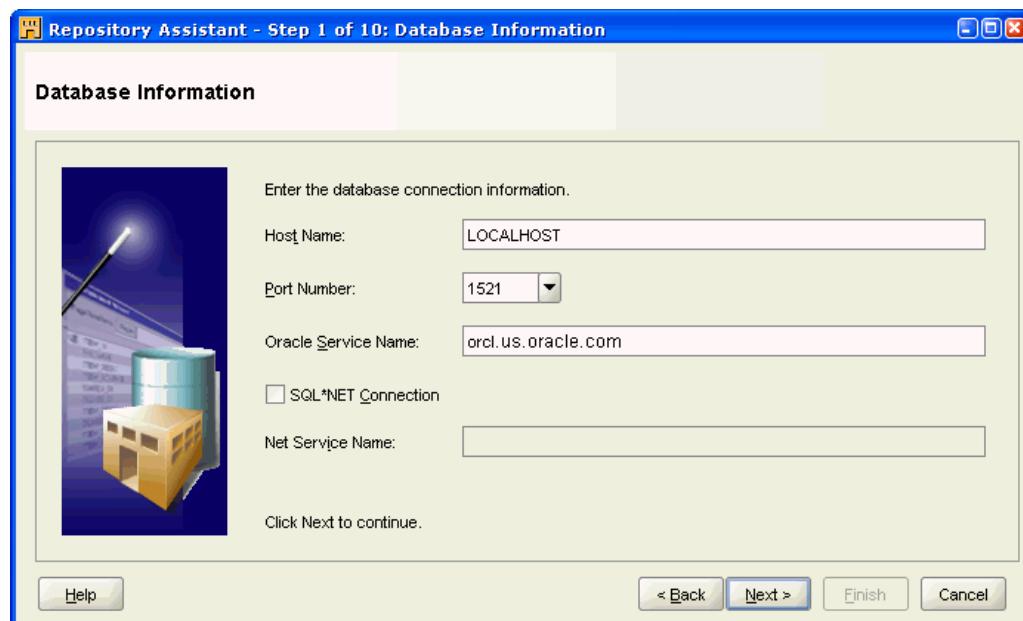
2. Start the repository assistant.

```
% ./reposinst.sh
```

3. On the Welcome screen of the Repository assistant, click **Next**.



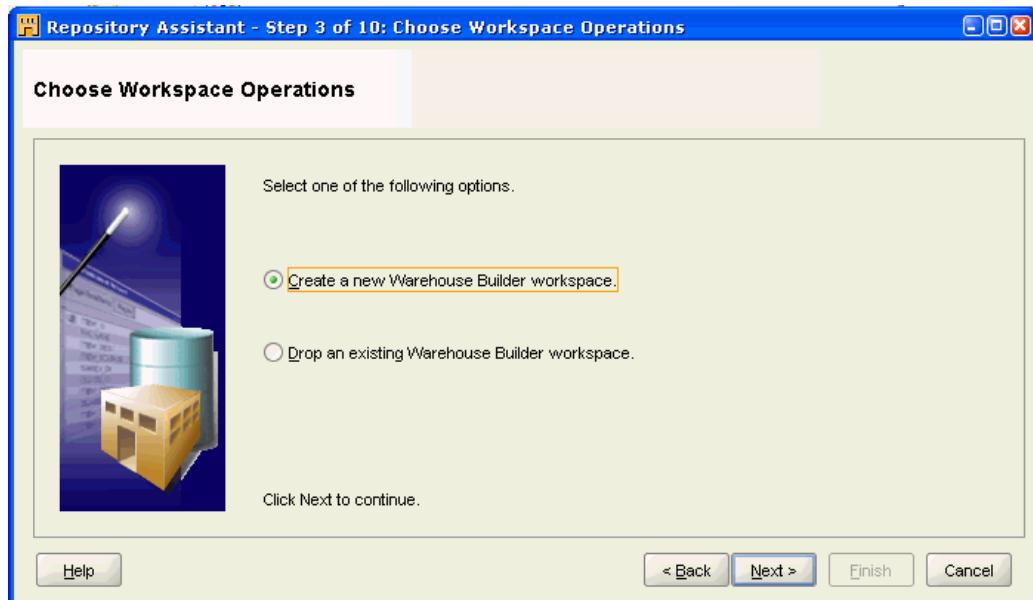
4. On the Database Information screen, enter the connection information (Host Name, Port Number, and Oracle Service Name) for the repository database, and click **Next**.



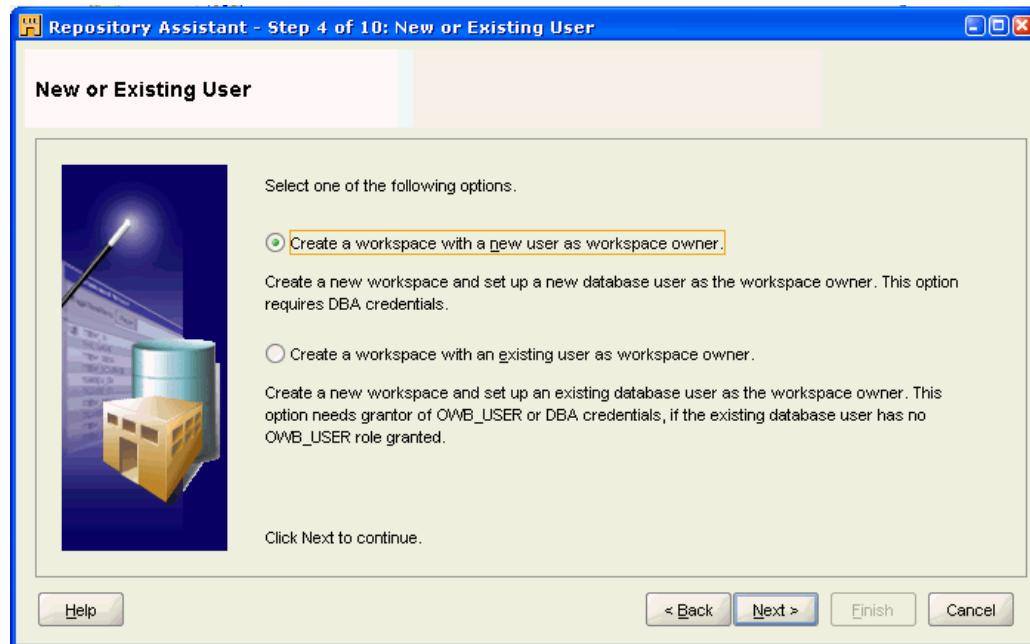
5. On the Choose Operation screen, select **Manage Warehouse Builder Workspaces**. Click Next.



6. On the Choose Workspace Operations screen, select **Create a New Warehouse Builder workspace**. Click Next.

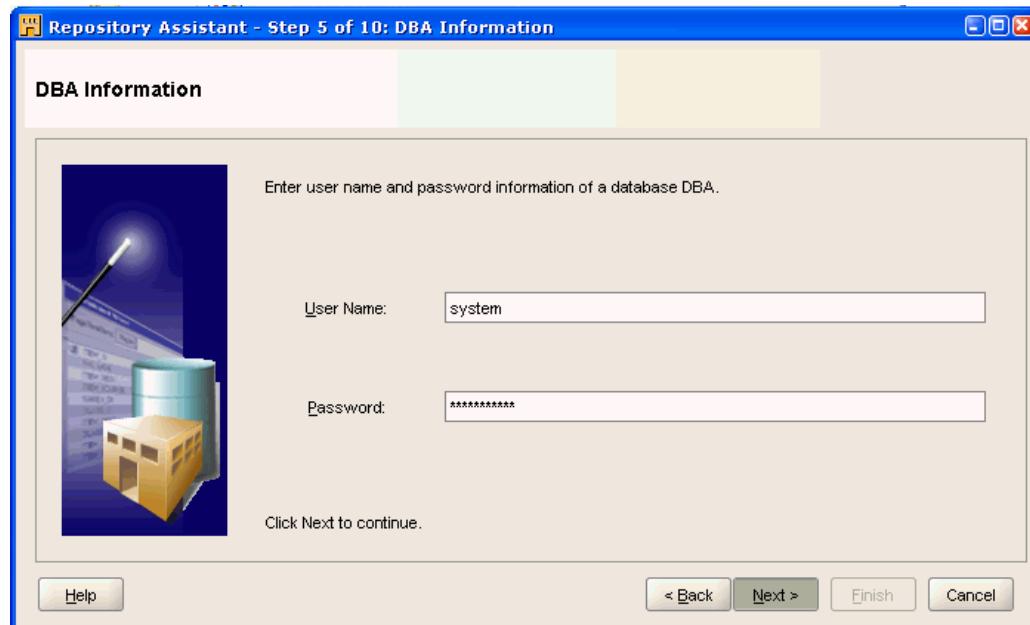


7. On the New or Existing User screen, select **Create a workspace with a new workspace owner**. Click Next.



8. On the DBA Information screen, enter the **User Name** of a user with a `CREATE USER` privilege, and that user's **Password**.

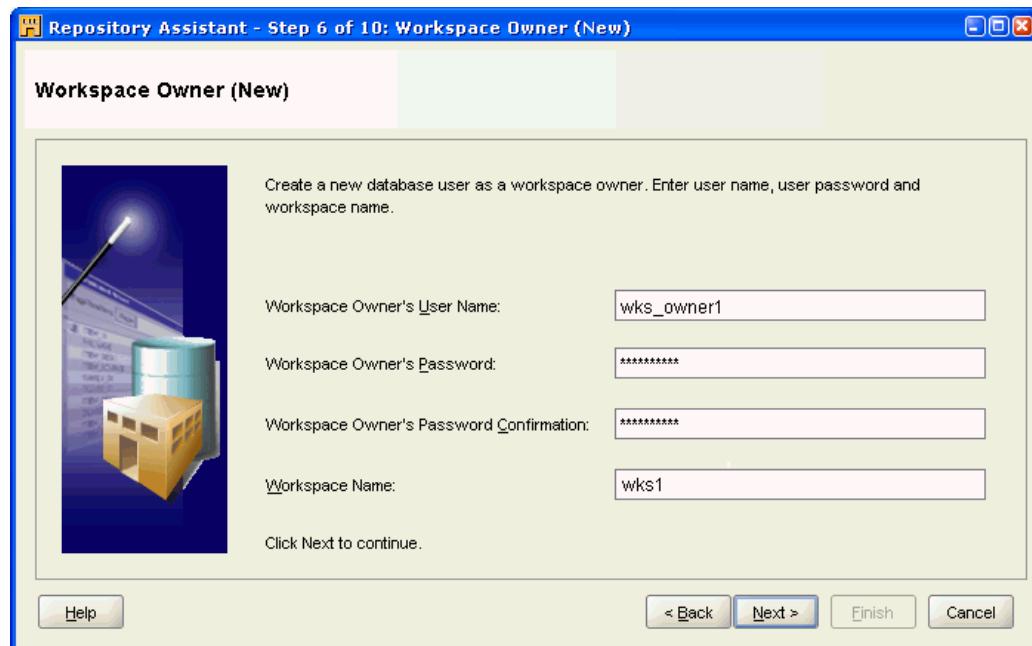
Click **Next**.



9. In the Workspace Owner (New) screen, enter the following information:

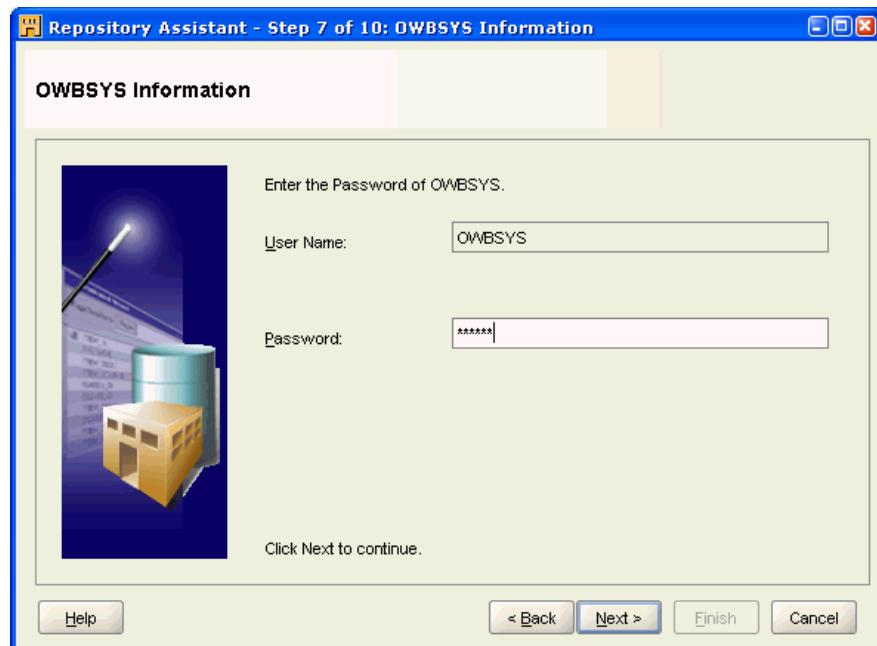
- Workspace Owner's User Name (wks_owner1)
- Workspace Owner's Password
- Workspace Owner's Password Confirmation
- Workspace Name (wks1) (Note that workspace names cannot contain spaces)

Click **Next**.



10. In the OWBSYS Information screen, enter the OWBSYS Password. Click Next.

Note that you need these credentials to seed the OWBSYS schema.



11. [Optional] For remote installations (when the schema is on a remote system), the system performs the verification of versions, and displays the message: "Found OWB software with the compatible version installed locally on the database server with path REMOTE_OWB_HOME."

If the version is either incompatible or cannot be found, then the system displays the message: "OWB software with the compatible version must be installed locally on the database server." If the compatible version is installed, then acquire the DBA user credentials and run the `remote_owb_install.sql` script from the

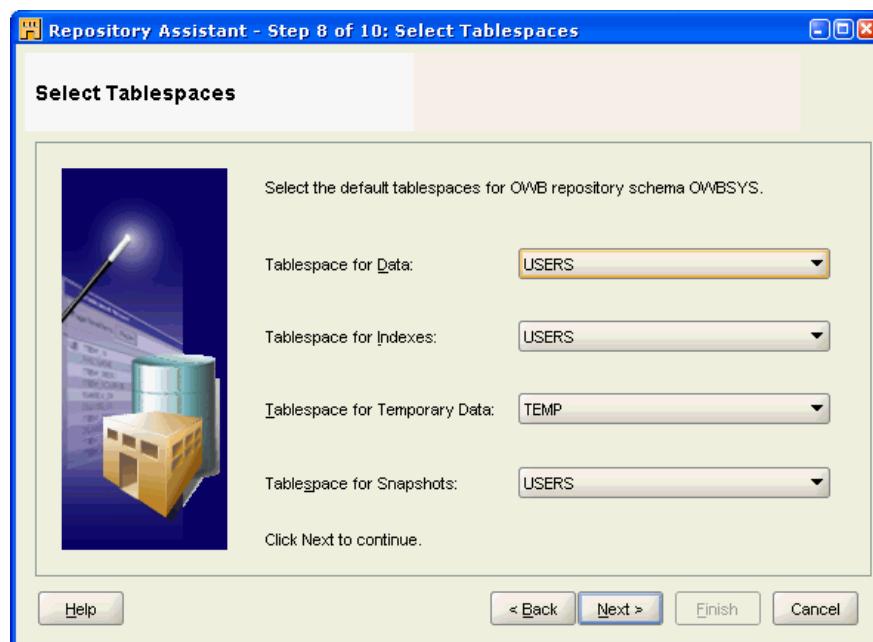
system where Oracle Database is installed, as described in "Setting the OWB Home for Remote Installations" on page 4-5.

12. On the Select Tablespaces screen, enter the following information:

- Tablespace for Data (USERS, or the name of the desired tablespace)
- Tablespace for Indexes (USERS, or the name of the same tablespace)
- Tablespace for Temporary Data (TEMP)
- Tablespace for Snapshots (USERS, or the name of the same tablespace)

Click **Next**.

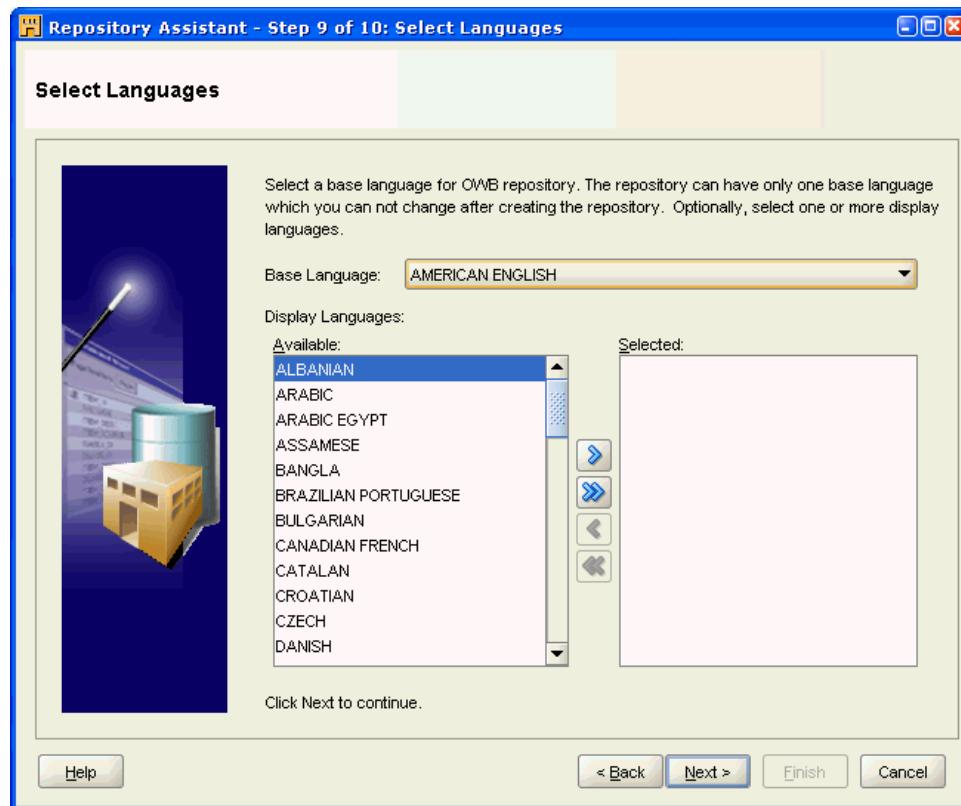
Note that this step allows you to change the default tablespace details. The OWB repository is stored in the OWBSYS schema, but you must identify what tablespaces are used to store OWBSYS and certain other objects, like snapshots. You may choose tablespaces other than USERS, depending on your storage management strategy.



13. On the Select Languages screen, select the Base Language from the menu.

[Optional] You may also select additional Display Languages from the list of Available languages on the left side, and move them to the list of Selected languages on the right side.

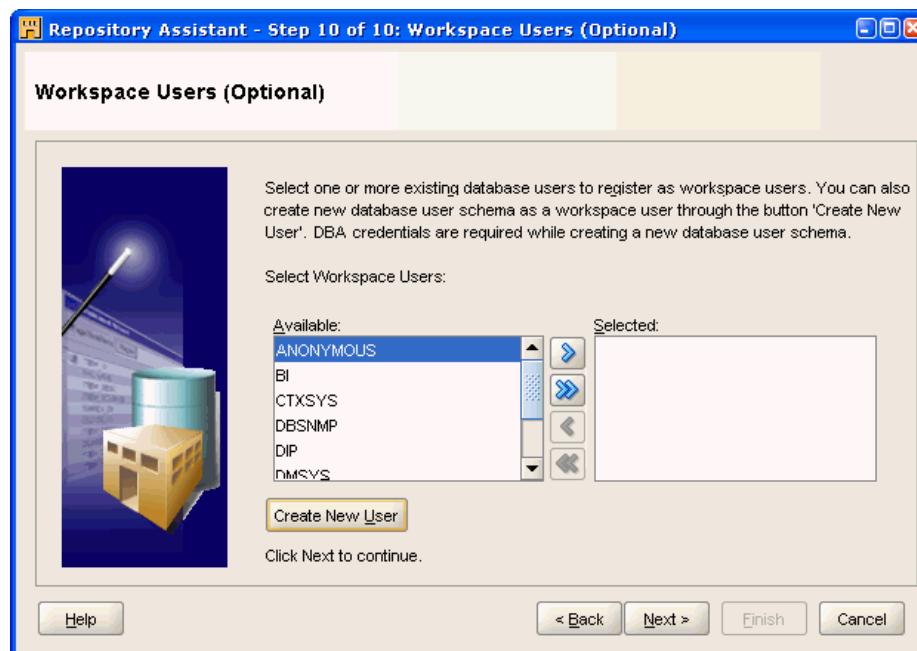
Click **Next**.



14. On the Workspace Users (Optional) screen, select the appropriate existing users from the Available list add them to the Selected list.

[Optional] Click **Create New User** to make new users who can access OWB. This takes you to step 15.

Click **Next**. This takes you to step 17.



15. [Optional] In the Create New Database User screen, enter the following:

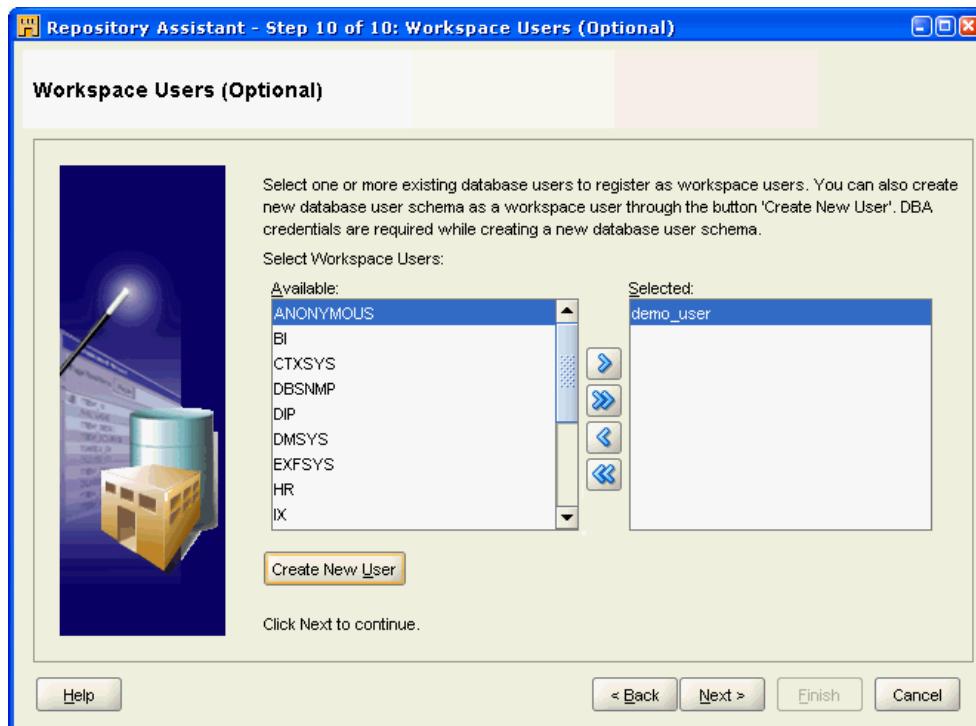
- User Name (`demo_user`)
- Password
- Re-enter Password

Click **OK**.

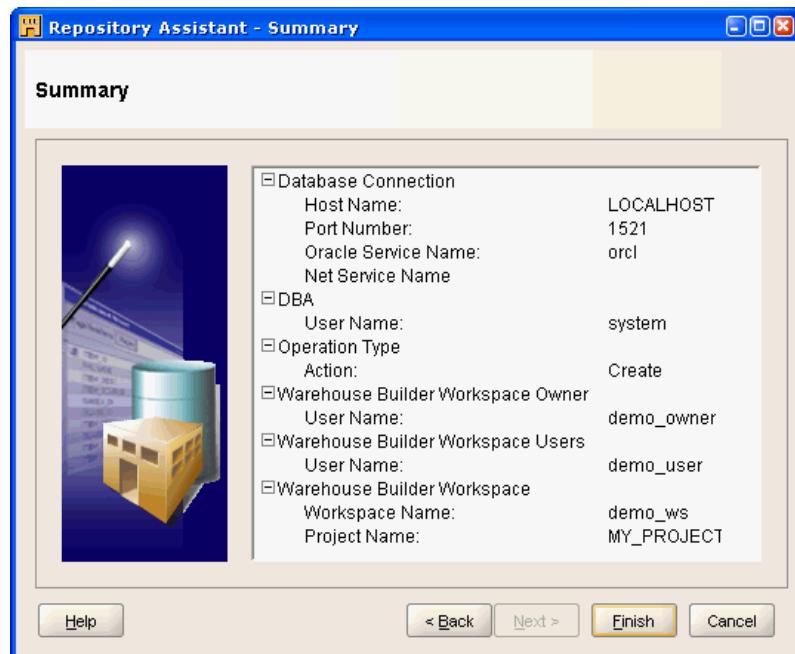
Note that DBA User Name (`system`) and DBA User Password are disabled fields and cannot be edited.



16. In the Workspace Users (Optional) screen, click **Next**.



17. On the Summary screen, click **Finish**.



18. Once the Installation Result screen appears, click **OK**.



Uninstalling Oracle Warehouse Builder

This section discusses how to uninstall Oracle Warehouse Builder.

This section includes the following topics:

- ["About Uninstalling Oracle Warehouse Builder" on page 5-1](#)
- ["Deleting the Workspace Users" on page 5-2](#)
- ["Deleting the Workspace" on page 5-6](#)
- ["Deleting the Schema Objects" on page 5-9](#)
- ["Deleting a Repository from an Oracle Database" on page 5-10](#)
- ["Uninstalling the Oracle Warehouse Builder Software" on page 5-10](#)

About Uninstalling Oracle Warehouse Builder

You cannot uninstall Oracle Warehouse Builder in Oracle Database 11g R2.

To uninstall earlier versions of Oracle Warehouse Builder, you have two general options:

- Removing only the client components

To remove OWB components from a host where there is no repository, follow the instructions in ["Uninstalling the Oracle Warehouse Builder Software" on page 5-10](#). If you want to uninstall all Warehouse Builder components including the repository, then you must have SYSDBA privileges to the repository.

- Removing both server and client components from your environment

To remove all components, you must complete the following tasks, in that sequence. Otherwise, you may have to manually delete some components:

1. Use the Advanced Setup option in the Repository Assistant to delete all workspace users, as described in ["Deleting the Workspace Users" on page 5-2](#).
2. Use the Advanced Setup option in the Repository Assistant to delete the workspace owner, as described in ["Deleting the Workspace" on page 5-6](#).
3. [Optional] Delete the OWB schema objects, as described in ["Deleting the Schema Objects" on page 5-9](#)
4. Delete the repository, as described in ["Deleting a Repository from an Oracle Database" on page 5-10](#)
5. Start the Universal Installer to uninstall the software components, as described in ["Uninstalling the Oracle Warehouse Builder Software" on page 5-10](#).

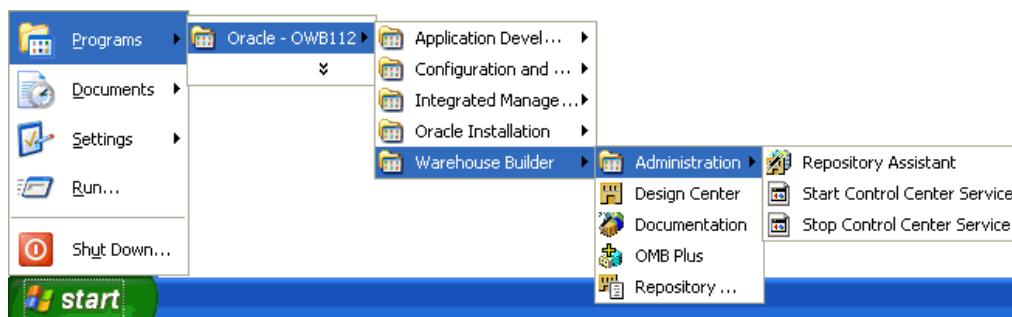
Deleting the Workspace Users

Before you can uninstall a workspace owner, you must first delete the associated workspace users. When you delete a workspace user, you un-register and remove the user from the repository. Deleting the user from the Warehouse Builder repository does not delete or alter the corresponding user account in the Oracle Database.

To delete workspace users:

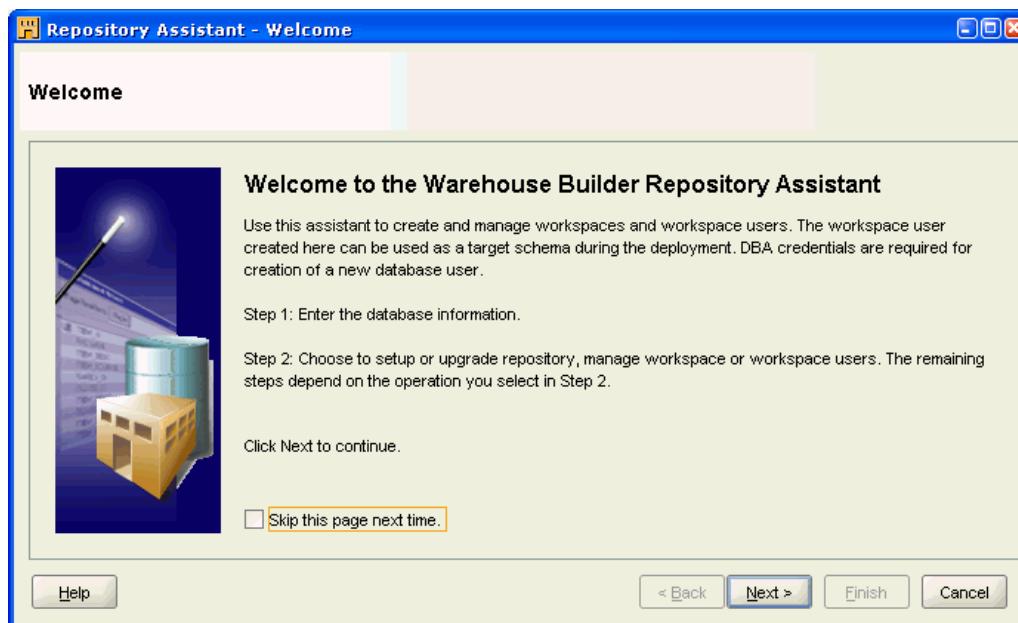
1. Start the Oracle Warehouse Builder Repository Assistant.

For Windows, select **Start, Programs, Oracle-OWB112, Warehouse Builder, Administration**, and then click **Repository Assistant**.



For Linux, in the `OWB_HOME/owb/bin/unix` directory, run the `reposinst.sh` script.

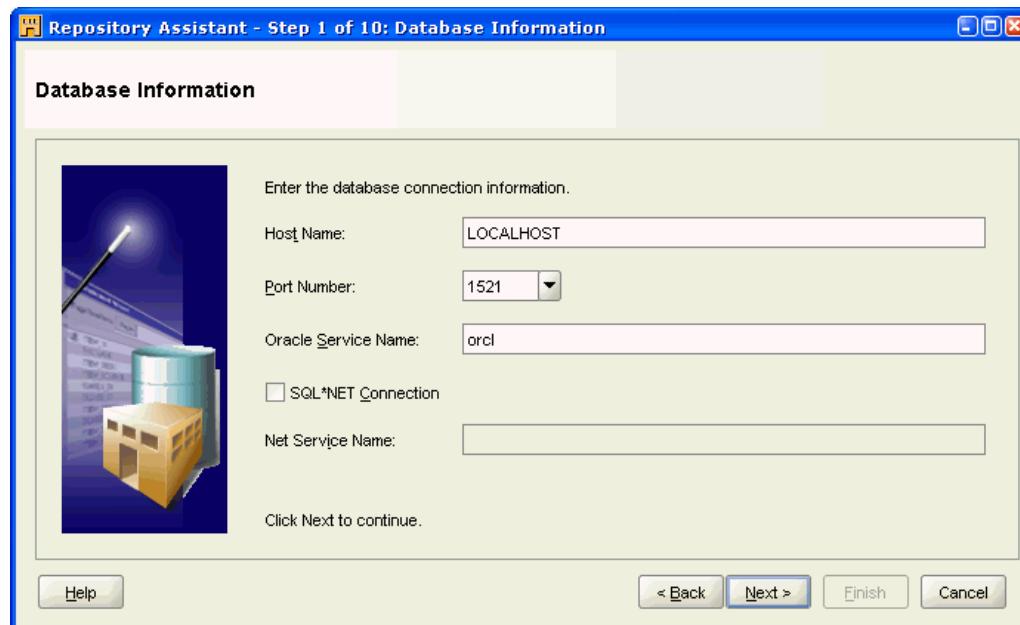
2. On the Welcome screen, click **Next**.



3. On the Database Information page, enter the **Host Name**, **Port Number**, and **Oracle Service Name**. Click **Next**.

Note that these details are automatically filled in based on the values entered in the install and configuration stage.

In this example, we are using the values of LOCALHOST, 1521, and orcl, respectively.



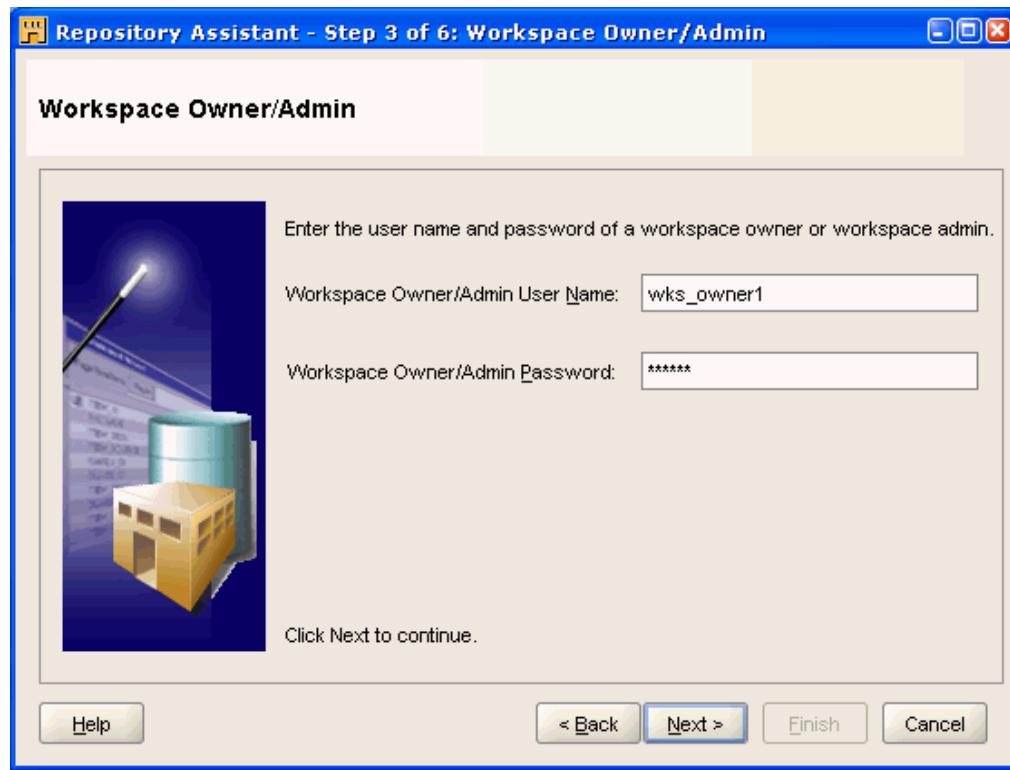
4. On the Choose Operation screen, select **Manage Warehouse Builder workspace users** option.

Click **Next**.



5. On the Workspace Owner/Admin screen, enter the **Workspace Owner / Admin User Name** (wks_owner1), and **Workspace Owner / Admin Password**.

Click **Next**.

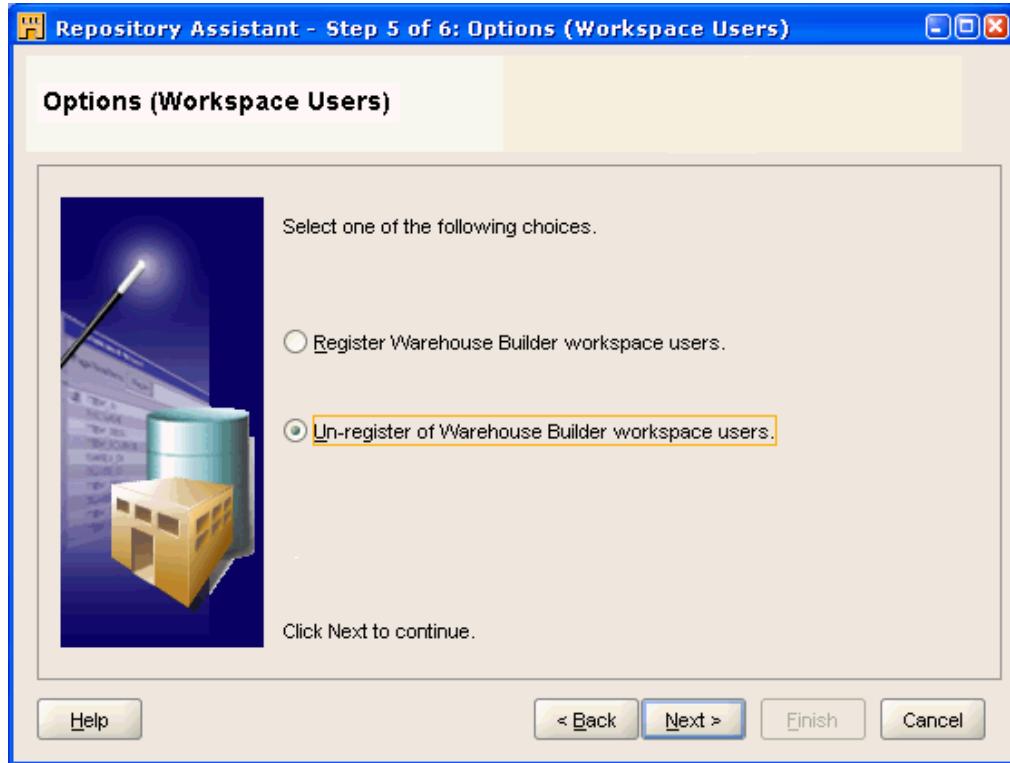


6. On the Workspaces screen, select the **Workspace Name** (WKS_OWNER1.WKS1).
Click **Next**.

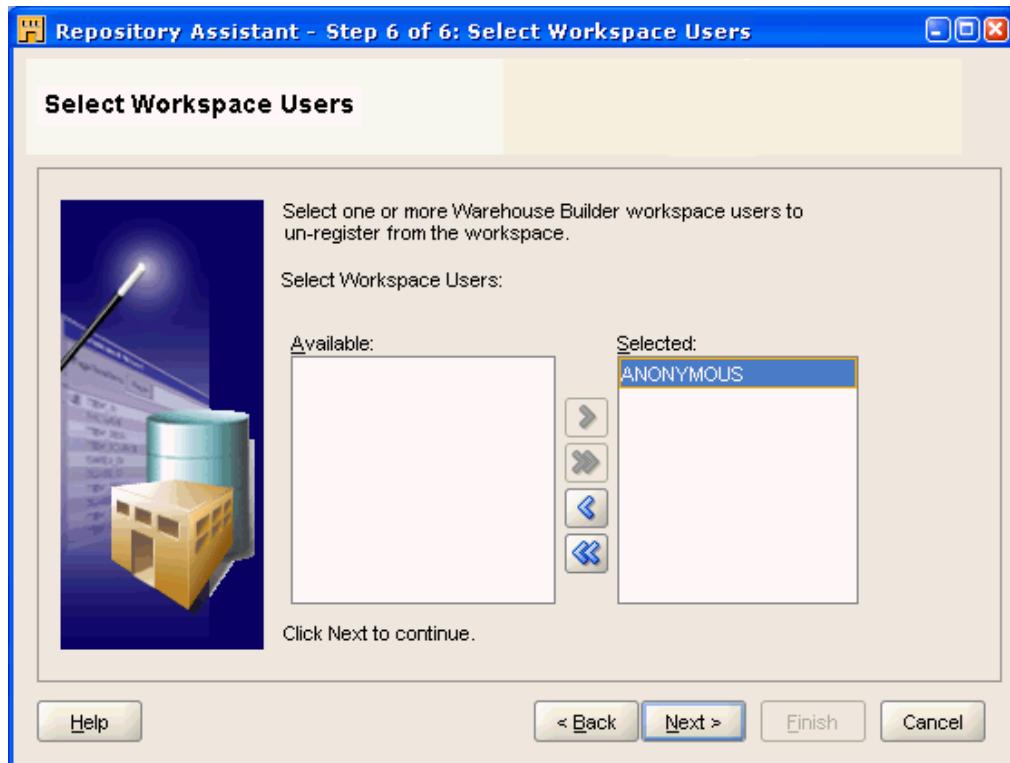


7. On the Options (Workspace Users) screen, select **Un-register of Warehouse Builder workspace users**.

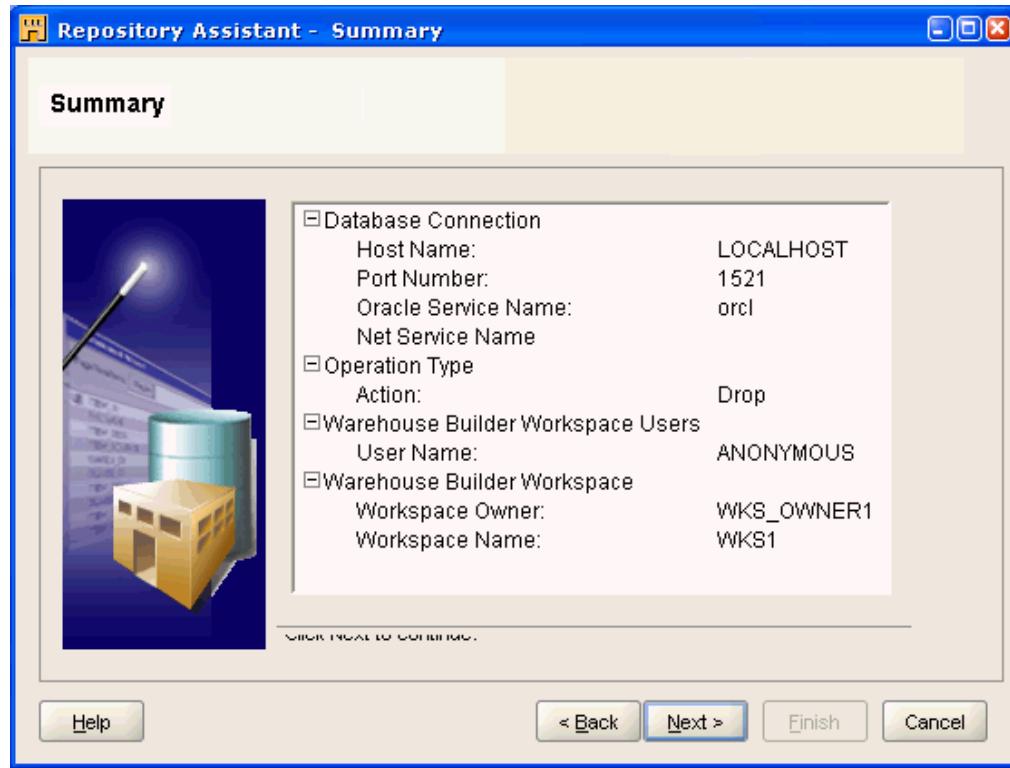
Click Next.



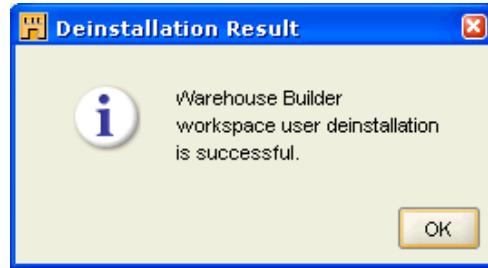
8. On the Select Workspace Users screen, move the users you want to delete from the Available list to the Selected list. Click Next.



9. In the Summary screen, click **Finish**.



- On the Deinstallation Result screen, click **OK**.



Deleting the Workspace

After deleting the workspace users, you can delete the workspace. When you delete a workspace owner, you un-register and remove the owner from the repository. Deleting the owner from the repository does not delete or alter the owner account in the Oracle Database.

To delete the workspace:

- Repeat steps 1 through 3 in "Deleting the Workspace Users" on page 5-2.
- On the Choose Operation screen, select **Manage Warehouse Builder workspaces**.

Click **Next**.



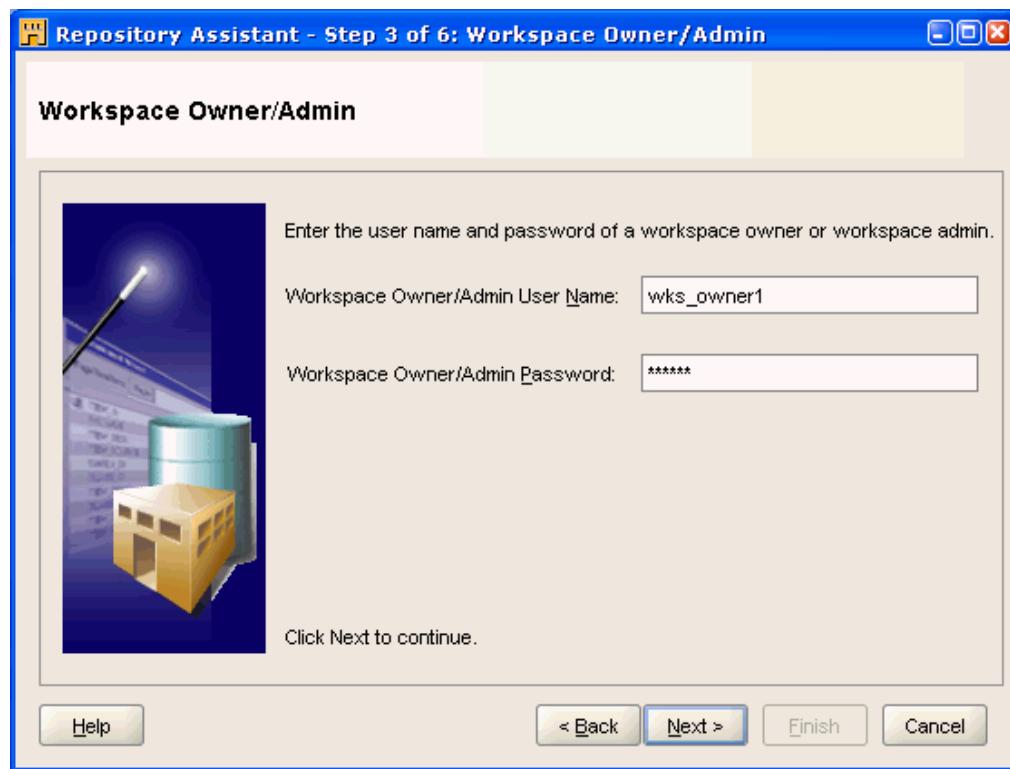
3. On the Choose Workspace Operations screen, select **Drop and existing Warehouse Builder workspace**.

Click Next.



4. On the Workspace Owner/Admin screen, enter the **Workspace Owner / Admin User Name** (wks_owner1), and **Workspace Owner / Admin Password**.

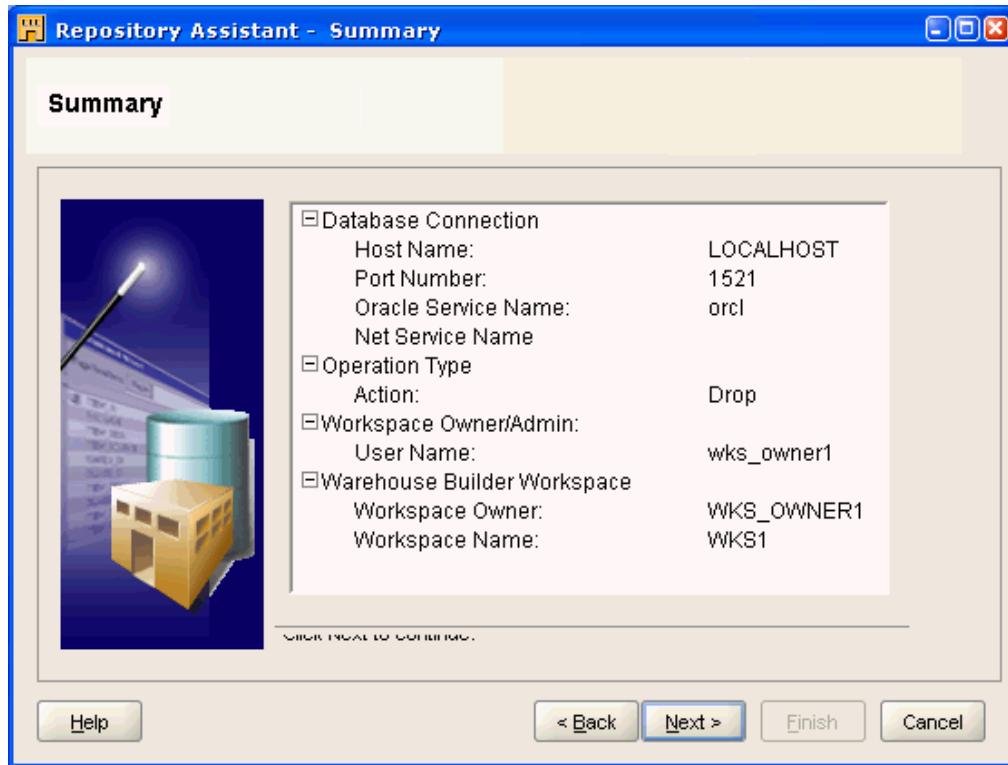
Click **Next**.



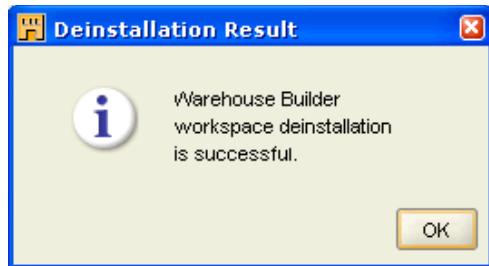
5. On the Workspaces screen, select the **Workspace Name** (WKS_OWNER.WKS1).
Click **Next**.



- On the Summary screen, click **Finish**.



- On the Deinstallation Result screen, click **OK**.



Deleting the Schema Objects

When you delete a workspace user or the workspace owner, you un-register and remove that owner from the repository. This operation, however, does not delete that user or owner, or alter the account in the Oracle Database.

To permanently delete the workspace users, workspace owners, and the Warehouse Builder-related roles and synonyms permanently from Oracle Database, use Oracle Enterprise Manager. To enable the OWB user to run Enterprise manager, you must grant the appropriate permissions.

To grant access to Enterprise Manager:

- At the command prompt, start SQL*Plus.

```
C:\>sqlplus
SQL*Plus: Release 11.2 - Production on Mon July 27 15:45:32 2009
```

Copyright (c) 1982, 2009, Oracle. All rights reserved.

2. When prompted for user name, log in as a user with administrative privileges.

```
Enter user-name: sys/as sysdba  
Enter password: password
```

```
Connected to:  
Oracle Database 11g Enterprise Edition Release 11.2 - Production  
With the Partitioning, OLAP and Data Mining options
```

3. Use the GRANT command:

```
SQL> GRANT SELECT any dictionary to "&OWB repository user";
```

Deleting a Repository from an Oracle Database

To delete an OWB 11g repository and associated objects from an Oracle Database 10g R2, use the SQL script `clean_owbsys.sql`, in the `OWB_HOME/owb/UnifiedRepos` directory.

Note that these instructions are for deleting earlier OWB installations. OWB cannot be deleted from Oracle Database 11g R2 as the schemas remain; however, running the `clean_owbsys.sql` script removes all the objects inside the schema.

To delete the repository:

1. At the command prompt, start SQL*Plus:

```
C:\>sqlplus  
SQL*Plus: Release 11.2 - Production on Mon July 27 15:58:47 2009  
  
Copyright (c) 1982, 2009, Oracle. All rights reserved.
```

2. When prompted for user name, log in as a user with administrative privileges.

```
Enter user-name: sys/as sysdba  
Enter password: password
```

```
Connected to:  
Oracle Database 11g Enterprise Edition Release 11.2 - Production  
With the Partitioning, OLAP and Data Mining options
```

3. Run the `clean_owbsys.sql` script to drop the OWBSYS user and OWB-related roles from the database:

```
SQL> @OWB112/owb/UnifiedRepos/clean_owbsys.sql C:/OWB112
```

Note that if you previously used the Repository Assistant to create a workspace, or a workspace owner and database user, these they still exist in the database after running the `clean_owbsys.sql` script.

Uninstalling the Oracle Warehouse Builder Software

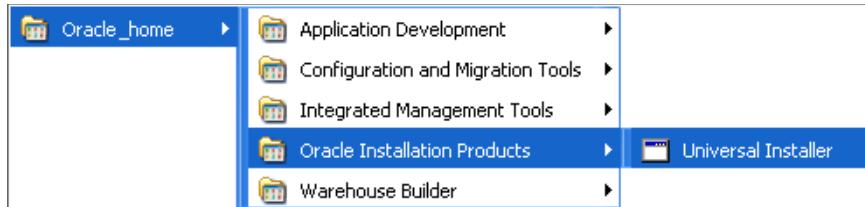
To uninstall OWB software, you must use the Oracle Universal Installer.

Note that uninstalling OWB deletes the files in `OWB_HOME`, namely the scripts necessary for cleaning up the schema and the repository. Therefore, this should be the last process you perform, after steps described in "["Deleting the Schema Objects"](#) on page 5-9 and "["Deleting a Repository from an Oracle Database"](#) on page 5-10.

To uninstall the OWB software:

1. Start Oracle Universal Installer.

For Windows, select Start, Programs, then select OWB_HOME, Oracle Installation Products, and then click Universal Installer.



For Linux, run the script runInstaller.sh in the OWB_HOME/oui/bin directory.

2. On the Oracle Universal Installer: Welcome page, click **Deinstall Products**.
3. On the Inventory page, under the **Contents** tab, in the box titled "You have the following Oracle products installed," select the **OWB home**.
Click **Remove**.
4. On the Confirmation page, click **Yes**.
The deinstallation process begins.
5. Once the process is complete, on the Inventory page, click **Close**.
6. On the Oracle Universal Installer: Welcome page, click **Cancel**.

This closes the Oracle Universal Installer.

Moving to Oracle Warehouse Builder 11.2 from Earlier Releases

Oracle Warehouse Builder enables you to upgrade repositories from Oracle Warehouse Builder 10g Release 2 and Oracle Warehouse Builder 11g Release 1 to Oracle Warehouse Builder 11g Release 2.

This section describes the different methods of moving to OWB release 11.2 from earlier releases, the upgrade and migration paths and upgrade process, and changes to upgraded ETL design objects to make them compatible with OWB release 11.2.

This section includes the following topics:

- ["Understanding MDL Migration and Repository Upgrades" on page 6-1](#)
- ["Choosing a Repository Upgrade Scenario" on page 6-3](#)
- ["In-Place Upgrade of OWB 10gR2 or 11gR1 Repository to 11gR2" on page 6-4](#)
- ["Migrating the Repository to a Different Oracle Database Instance" on page 6-14](#)
- ["Upgrading Locations to OWB Release 11.2" on page 6-15](#)
- ["Changes to Repository Objects When Moving to OWB Release 11gR2" on page 6-17](#)

Understanding MDL Migration and Repository Upgrades

The two ways to move projects from earlier releases into OWB release 11.2 are migrating objects and upgrading repositories.

This discussion uses the following definitions:

- *Upgrading a repository* is the process of taking an entire repository, including the control center audit data such as deployment and execution history, from an earlier OWB release and moving it to OWB release 11.2. This process is performed from within Repository Assistant.
- *Migrating objects*, in the context of moving between OWB releases, is the process of exporting existing OWB design-time objects from an earlier OWB release, and importing the resulting MDL file into OWB release 11.2. This process is performed from OWB Design Center.

Choosing Between MDL Migration and Repository Upgrade

The Repository Assistant in OWB release 11.2 provides ways to upgrade OWB release 10.2 and release 11.1 repositories directly. The advantages of upgrading an entire repository are:

- You can move up to OWB release 11.2 without regenerating code or redeploying your mappings, for maximum stability, then decide on objects to regenerate or redeploy based on whether you need specific release 11.2 enhancements.
- Upgrading can bring forward all design-time and runtime audit metadata from the previous repository, such as the history of deployments and executions of your ETL mappings, process flows, and other objects.
- All projects and global objects in a repository are processed at once. (For OWB 11gR1 repositories, all projects in all workspaces are processed at once.)

The advantages of migrating individual projects or other objects include:

- Speed, if you have large volumes of runtime metadata that you do not need to preserve.
- Simplicity, in cases where the upgrade path for the complete repository is more complex than the migration path. In particular, OWB release 11.2 can directly import MDL for projects from OWB 9.2 and 10.1, but cannot upgrade the entire repository from these earlier releases. For example, upgrading an entire repository from OWB 10.1 or older may require creating one or more new OWB environments from intermediate releases, to be used for these intermediate upgrades.

MDL Migration Paths

OWB release 11.2 can directly import MDL for projects or other objects from OWB 9.2.0.n or higher. Importing metadata from MDL files from earlier releases is performed in the OWB release 11.2 Design Center. The process is described in [Chapter 12, "Using the Metadata Loader."](#) The process is like importing any other MDL file. However, you will be prompted for a filename for an intermediate MDL file (which is your file upgraded to the new format for OWB release 11.2). This file is generated, saved at a location you specify, and then immediately imported.

For releases older than 9.2.0.n, the supported migration path requires an intermediate step:

1. Import the MDL to an intermediate 10.2.0.5 or 11.1.0.7 database. These versions should be able to read any older MDL.

The MDL file upgrade process is performed during the import process. OWB generates an upgraded MDL file; you will be prompted for a filename.

The upgraded MDL file is written to the filename you specify, and is then imported into your intermediate database. You can ignore this and simply copy the intermediate file to a new location.

2. Import that upgraded MDL into OWB 11.2 using Design Center, as described in [Chapter 12, "Using the Metadata Loader."](#)

Note: The latest 10.1, 10.2 or 11.1 patch release is strongly recommended for cases where an intermediate release is used in the upgrade path. In some cases, earlier patch releases may not import or upgrade metadata from all earlier releases of the product.

Repository Upgrade Paths from OWB 10.1 or 9.n to release 11.2

OWB release 11.2 can directly upgrade repositories from OWB 10.2.0.n or OWB 11.1.0.n. Repositories from older releases cannot be upgraded directly. OWB 10.2 becomes an intermediate step in the upgrade path.

OWB 10gR2 combined the separate design-time and runtime repositories from earlier releases into a single unified repository, which is used in all subsequent releases. The OWB 10gR2 Repository Assistant contains the functionality required to merge previously separate design-time and runtime repository contents from OWB 10.1.0.4 into the new combined repository model. If your source is 9.2.0.n or earlier, then you must first upgrade to OWB 10.1.0.4.

Therefore, if upgrading both runtime and design-time metadata from a 10.1 or earlier release, the stages required are:

- Prepare an OWB 10.2 environment
- Upgrade the separate design-time and run-time repositories from the earlier releases into the combined OWB 10.2 repository in that environment
- Upgrade the resulting 10.2 repository to OWB 11.2.

a cross reference needed here

The following table summarizes the repository upgrade paths for releases prior to OWB 10.2:

Table 6–1 Repository Upgrade Paths for OWB 10.1 and Earlier

Current Release	Repository Upgrade Path
10.1.0.n	To 10.2.0.n, then to 11.2
9.2.0.n or Older	To 10.1.0.4, then to 10.2.0.n, then to 11.2

To perform these repository upgrades, you will require working environments for the intermediate versions on the upgrade path.

For example, if you are upgrading a 9.2 repository, you must set up an OWB 10.1.0.4 environment and an OWB release 10.2.0.5 environment for use during the upgrade steps, then upgrade from release 9.2 to release 10.1.0.4 and upgrade the result to OWB release 10.2.0.5.

See the documentation for those older releases for details on the intermediate upgrades.

If you are moving to OWB 11.2 from OWB 10.1 or earlier, then it is usually simpler to migrate objects on a project-by-project basis into a clean OWB release 11.2 workspace. Export the projects to MDL in Design Center using the process for those releases, and import the MDL objects into OWB release 11.2, using the steps in [Chapter 12, "Using the Metadata Loader."](#)

cross-ref please.

Choosing a Repository Upgrade Scenario

The Repository Assistant for OWB 11.2 supports two major upgrade scenarios:

- Repository upgrade through MDL files, where the complete repository from OWB 10.2 or OWB 11.1 is exported to an MDL file, and then that MDL file is later imported into a database with an OWB 11.2 repository schema

- An in-place repository upgrade, in which case the new OWB 11.2 repository is immediately created in the same database where the OWB 10.2 or 11.1 repository was stored.

Note the following:

- Upgrading through MDL files is the only option when moving your entire repository to a new database. (Exporting the entire repository to MDL is also a good option for backing up the repository.)
- An in-place repository upgrade of OWB 11.1 overwrites the OWB 11.1 repository with the OWB 11.2 repository, because both are stored in the same repository schema.
- Internally, an in-place repository upgrade actually performs a repository export to MDL and repository import from MDL. The advantage of in-place repository upgrade is that the user performs fewer manual steps than in the upgrade through MDL files when targeting the same database.

In-Place Upgrade of OWB 10gR2 or 11gR1 Repository to 11gR2

An in-place upgrade of an OWB repository is the simplest method of a repository upgrade where the target release 11.2 repository is stored in the same database instance that contained the source repository. You can perform an in-place upgrade of either a 10g R2 or 11g R1 repository that exists on the same database instance as your Oracle Database 11g R2 installation.

For OWB 10gR2, Repository Assistant exports the specified repository to an MDL file, then imports that MDL file into a new release 11.2 workspace in the OWBSYS schema in the same database. The workspace name is the same as the schema that contained the 10gR2 repository.

For OWB 11gR1, Repository Assistant exports the whole repository from the OWBSYS schema to an MDL file, drops all contents of the OWB 11gR1 repository (as if you had run `clean_owbsys.sql`), creates a new OWB release 11.2 repository in OWBSYS, and re-imports the repository from MDL.

Caution: If your target database already has an OWB 11gR1 OWBSYS schema, then back it up before starting the actual migration using the 11.2 Repository Assistant. This provides you a way to recover from any possible failures during the upgrade process. The process described in "Preparing for In-Place Repository Upgrade" on page 6-4 overwrites the OWBSYS schema (if it exists) with the OWB 11.2 schema.

Preparing for In-Place Repository Upgrade

To perform the tasks in this section, you must have SYSADMIN privileges, which enable you to run prerequisite scripts.

Note: Check the latest OWB release notes at
<http://www.oracle.com/technetwork/developer-tools/warehouse/overview/index.html> before starting the upgrade process.

The steps for preparing a repository for in-place repository upgrade are slightly different for Windows and Linux.

To prepare for an in-place repository upgrade on Windows

1. Ensure that you make a back-up copy of the OWBSYS schema, using the backup strategy employed in your organization.

2. On the Microsoft Windows desktop, click **Start** and select **Run**.

3. In the Run window, for **Open**, enter cmd.

Click **OK**.

4. At the command prompt, start SQL*Plus:

```
C:\>sqlplus
```

5. Connect as a user with administrative privileges:

```
SQL> connect sys as sysdba
Enter password: password
```

6. You have two options for the OWBSYS schema:

- If the OWBSYS schema does not exist on the Oracle Database instance, then run the cat_owb.sql script to create it:

```
SQL> @C:\OWB112\owb\UnifiedRepos\cat_owb.sql
```

- If the OWBSYS schema exists on the Oracle Database instance, and you are upgrading from OWB 11g Release 1, then run the cat_owb_reposupgrade.sql script to upgrade it:

```
SQL> @C:\OWB112\owb\UnifiedRepos\cat_owb_reposupgrade.sql
```

7. Run the reset_owbcc_home.sql script to reset the *OWB_HOME* directory:

```
SQL> @C:/OWB112/owb/UnifiedRepos/reset_owbcc_home.sql
```

Note that this entry is case-sensitive, does not accept a trailing slash, and requires forward slashes only, regardless of the operating system.

To prepare for an in-place repository upgrade on Linux

1. Ensure that you make a back-up copy of the OWBSYS schema, using the backup strategy employed in your organization.

2. At the command prompt, start SQL*Plus:

```
% sqlplus
SQL*Plus: Release 11.2 - Production on Mon July 27 16:37:21 2009
```

```
Copyright (c) 1982, 2009, Oracle. All rights reserved.
```

3. Connect as a user with administrative privileges:

```
SQL> connect sys as sysdba
Enter password: password
```

4. You have two options:

- If the OWBSYS schema does not exist on the Oracle Database instance, run the cat_owb.sql script:

```
SQL> @OWB112/owb/UnifiedRepos/cat_owb.sql
```

Enter the tablespace name when you are prompted for it.

- If the OWBSYS schema exists on the Oracle Database instance, and you are upgrading from OWB 11g Release 1, then run the `cat_owb_reposupgrade.sql` script:

```
SQL> @OWB112/owb/UnifiedRepos/cat_owb_reposupgrade.sql
```

You are prompted for the tablespace name; enter it.

5. Run the `reset_owbcc_home.sql` script to reset the `OWB_HOME` directory:

```
SQL> @OWB112/owb/UnifiedRepos/reset_owbcc_home.sql
```

Note that this entry is case-sensitive, does not accept a trailing slash, and requires forward slashes only, regardless of the operating system.

6. Unlock the OWBSYS and OWBSYS_AUDIT accounts, as described in the following sections:

- *For Windows*, "[Unlocking the OWBSYS and OWBSYS_AUDIT Accounts](#)" on page 3-5
- *For Linux*, "[Unlocking the OWBSYS and OWBSYS_AUDIT Accounts](#)" on page 4-4

Performing the In-Place Repository Upgrade

When upgrading from an OWB 11g R1 repository, you are upgrading an existing OWBSYS schema on an existing repository. When upgrading from OWB 10.2, your upgrade process will create a new workspace in the OWB 11.2 repository, with the same name as the schema containing the OWB 10.2 repository. In either case, the process is similar.

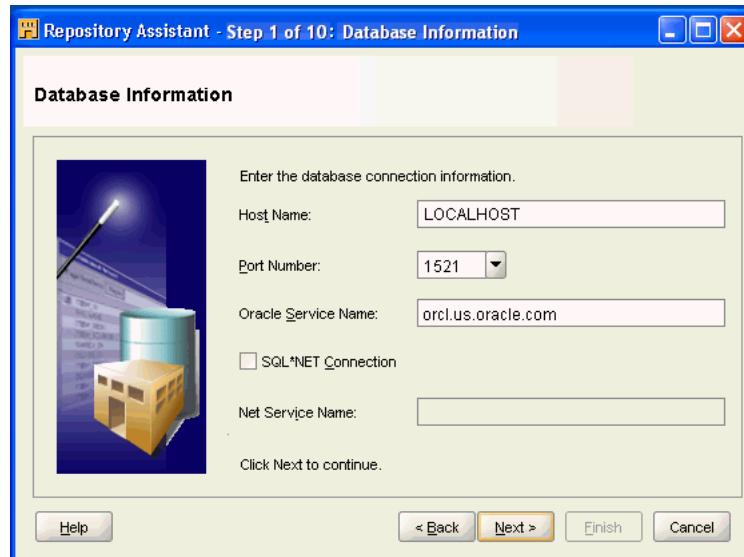
To perform an in-place repository upgrade

1. Start the Repository Assistant.

In Windows, from the **Start** menu, select **Program Files**, then **Oracle - OWB112**, then **Warehouse Builder**, then **Administration**, then **Repository Assistant**.

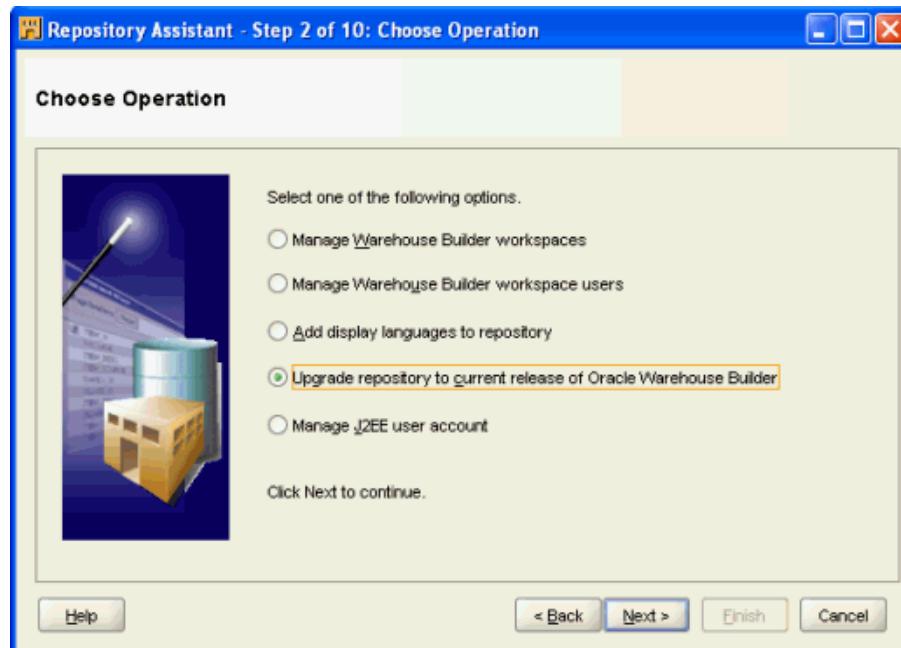
In Linux, from the command line, run
`OWB_HOME/owb/bin/unix/reposinst.sh`.

2. On the Welcome page of the Repository Assistant, click **Next**.
3. On the Database Information page, enter the host name, port number, and service name where OWB is installed.



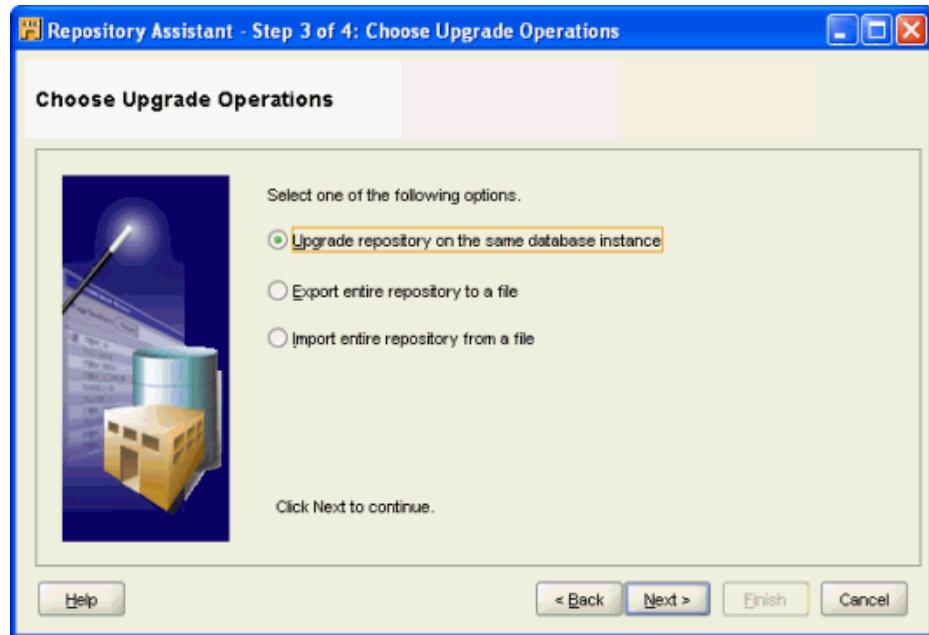
Click **Next**.

4. On the Choose Operation page, select **Upgrade repository to current release of Oracle Warehouse Builder**.



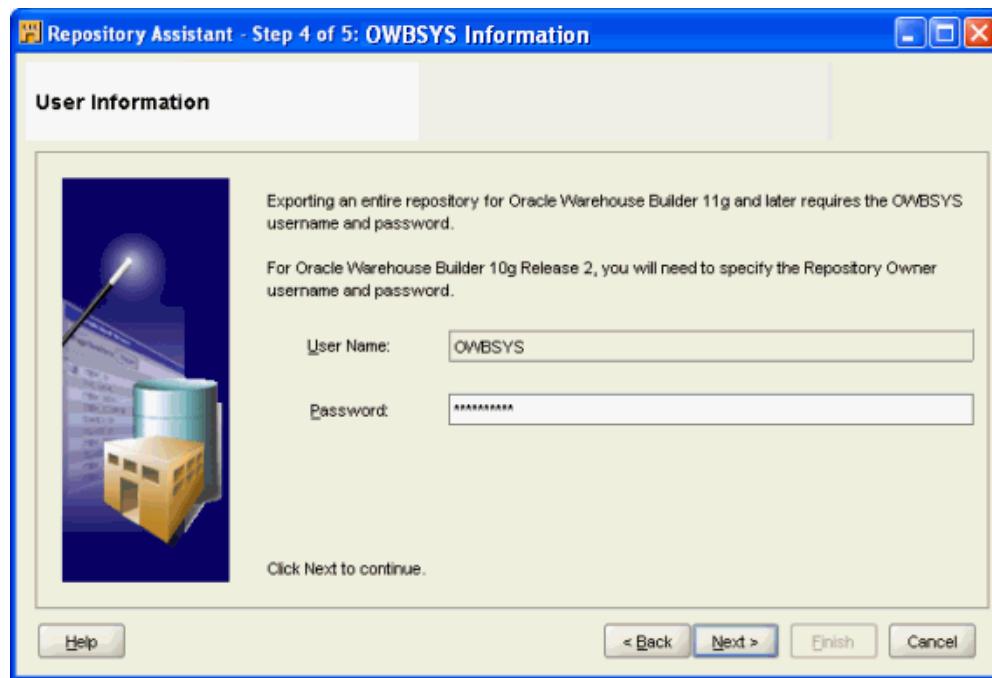
Click **Next**.

5. On the Choose Upgrade Operations page, select **Upgrade repository on the same database instance**.



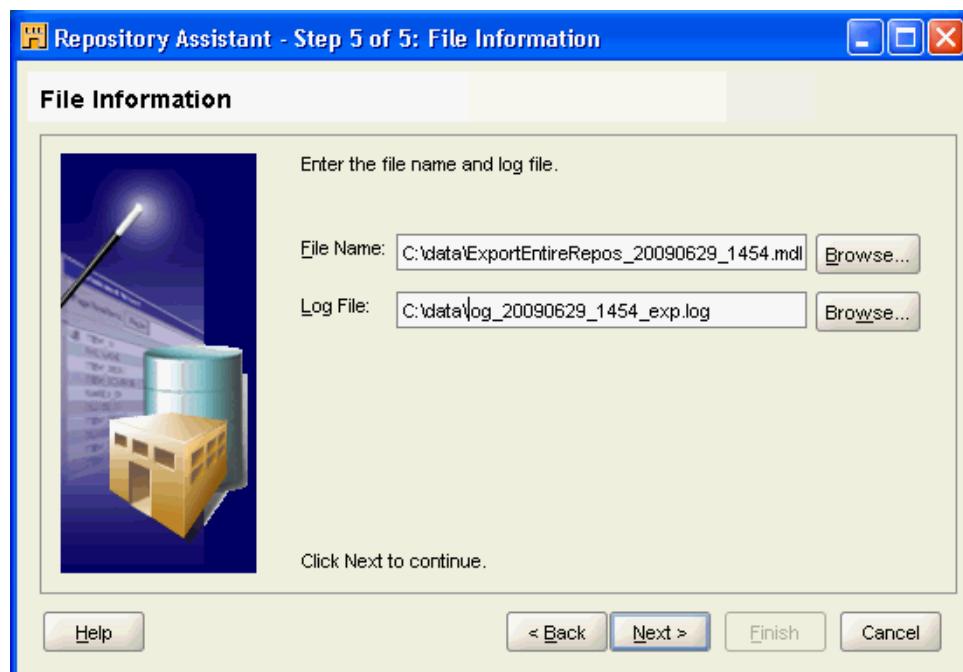
Click **Next**.

6. On the OWBSYS Information page, enter the password for the OWBSYS user.



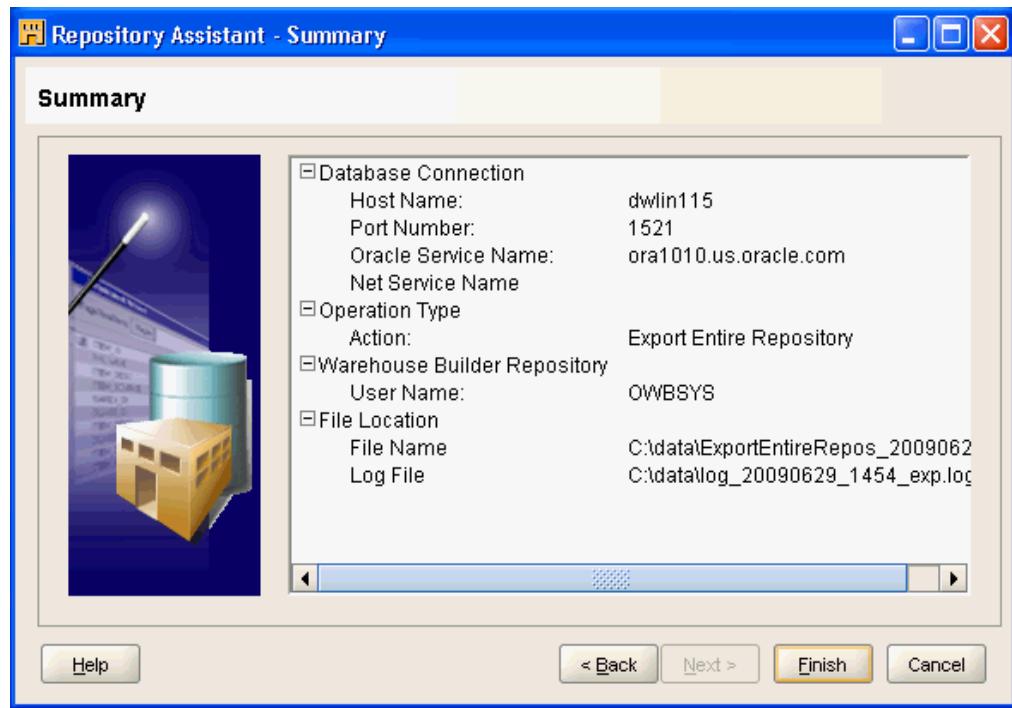
Click **Next**.

7. The File Information page appears. The in-place repository upgrade process creates a repository export MDL file. The default file names and location for this exported MDL file and export log file are displayed here. You can change the location of these files.



Click **Next** to accept the file information.

8. On the Summary page, click **Finish** to initiate the upgrade process.



9. The Installation Progress popup window indicates the progress of the upgrade operation. When complete, click **OK**.

You may wish to review the log for errors, and save the exported MDL file as a backup of your pre-upgrade repository.

Upgrading a Repository to release 11.2 through MDL Files

You can upgrade an OWB release 10.2 or 11.1 repository by exporting its content to a file, and then importing the file into an OWB release 11.2 repository, using the Repository Assistant.

Exporting an OWB Repository to a File

You export a repository with the Repository Assistant as follows.

To export a repository to a file

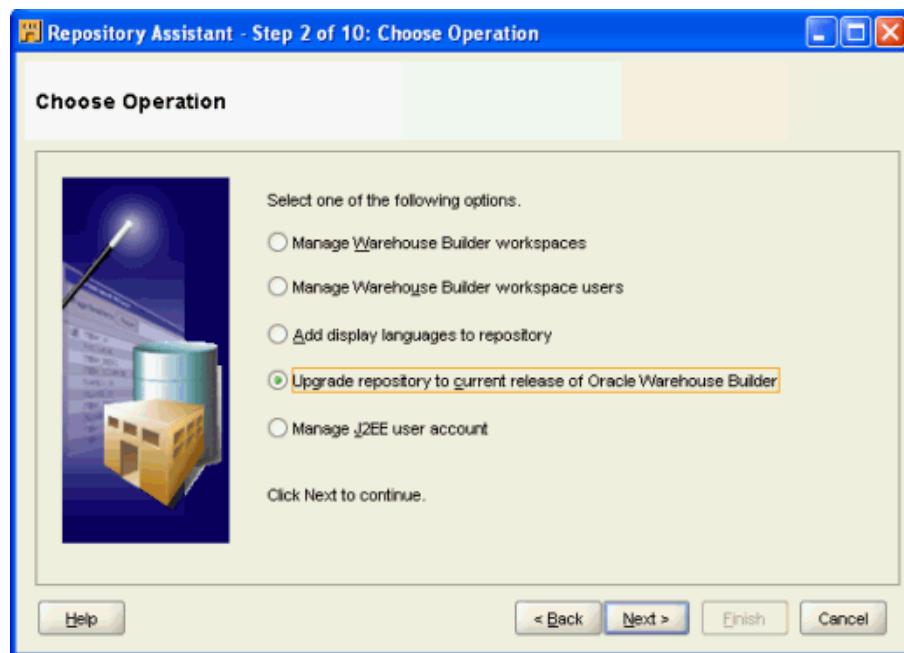
1. Start the Repository Assistant.

In Windows, from the **Start** menu, select **Program Files**, then **Oracle - OWB112**, then **Warehouse Builder**, then **Administration**, then **Repository Assistant**.

In Linux, from the command line, run

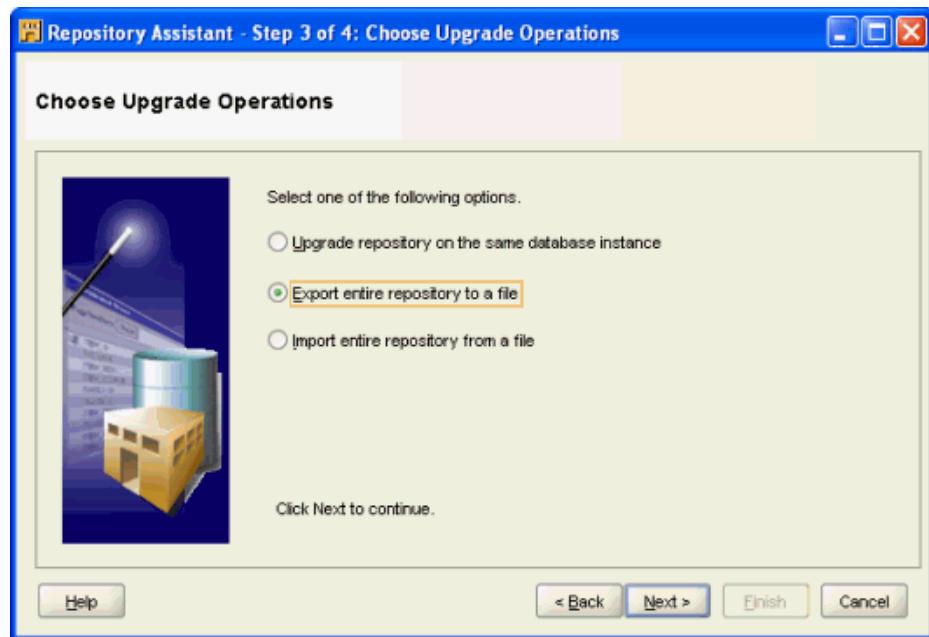
`OWB_HOME/owb/bin/unix/reposinst.sh`.

2. On the Welcome page of the Repository Assistant, click **Next**.
3. On the Database Information page, enter the host name, port number, and the service name where the OWB is installed.
Click **Next**.
4. On the Choose Operation page, select **Upgrade repository to current release of Oracle Warehouse Builder**.



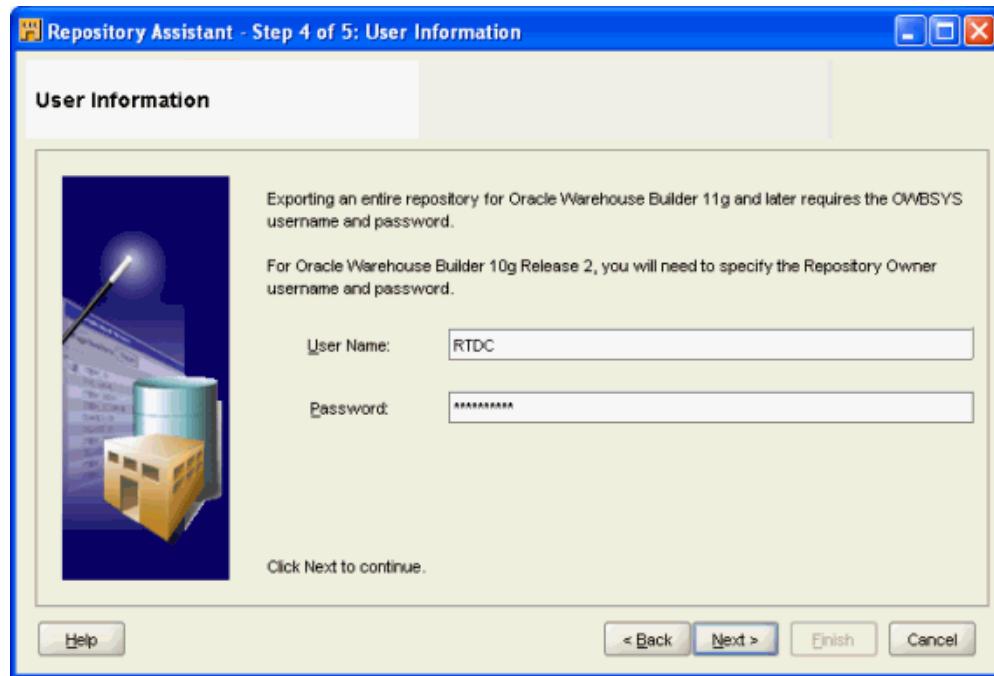
Click **Next**.

5. On the Choose Upgrade Operation page, select **Export entire repository to a file**.



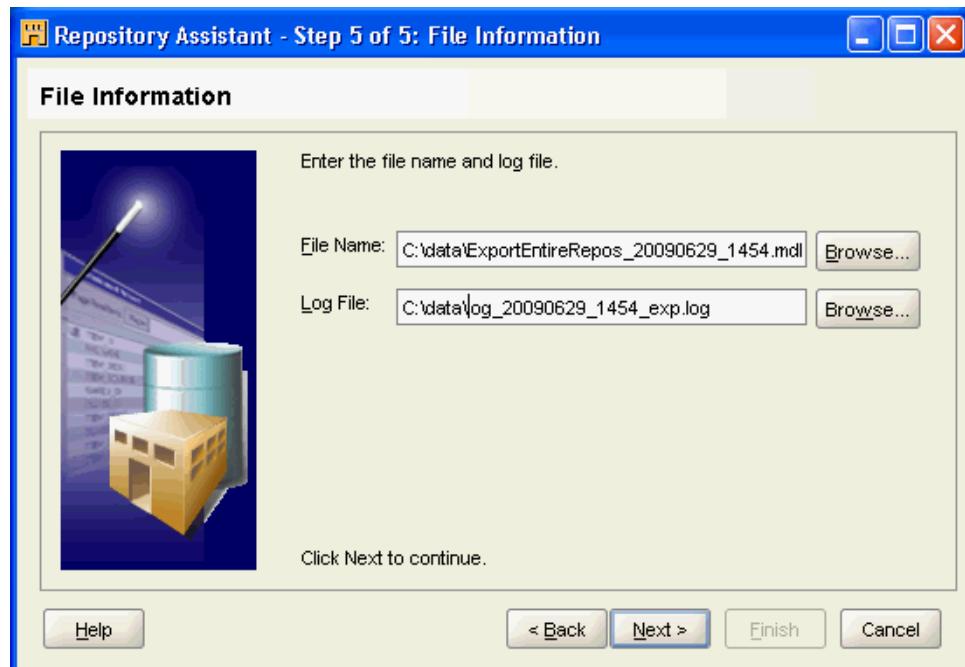
Click **Next**.

6. On the User Information page, for an OWB release 10.2 source, enter the user name and password of the OWB release 10.2 repository owner. For an OWB release 11.1 source, enter the OWBSYS user name and password.



Click **Next**.

7. On the File Information page, enter the following information:
 - File Name:** the name of the MDL file for storing the repository.
 - Log File:** the name of the export operation log file that stores status and statistical information.



Click **Next**.

8. On the Summary page, click **Finish**.

An Export Repository Progress window appears showing progress.

At the end of the export process, a status message appears. The export operation creates the file.

Importing a Repository Export MDL File to OWB Release 11.2

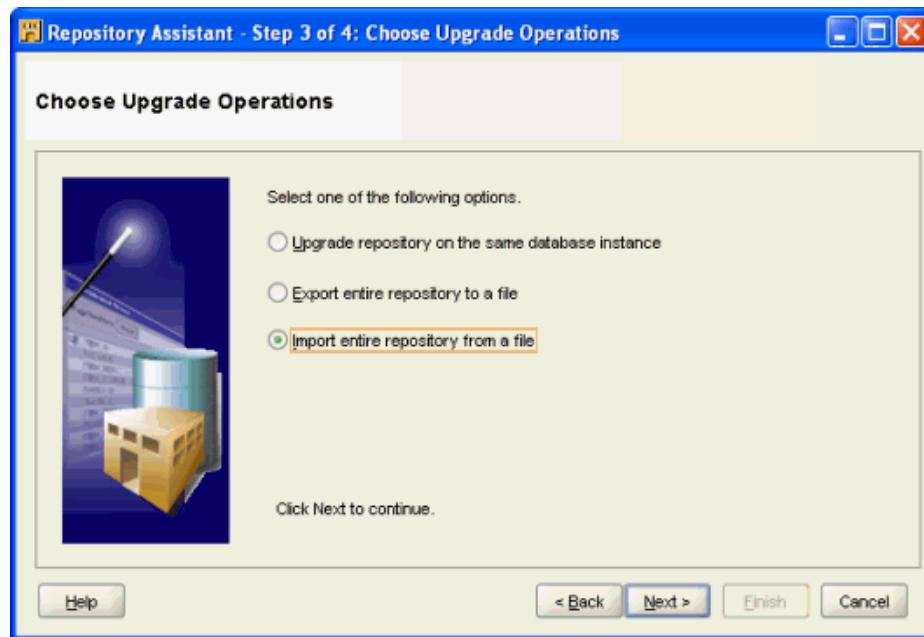
To import a repository export MDL file

1. Start the Repository Assistant.

In Windows, from the **Start** menu, select **Program Files**, then **Oracle - OWB112**, then **Warehouse Builder**, then **Administration**, then **Repository Assistant**.

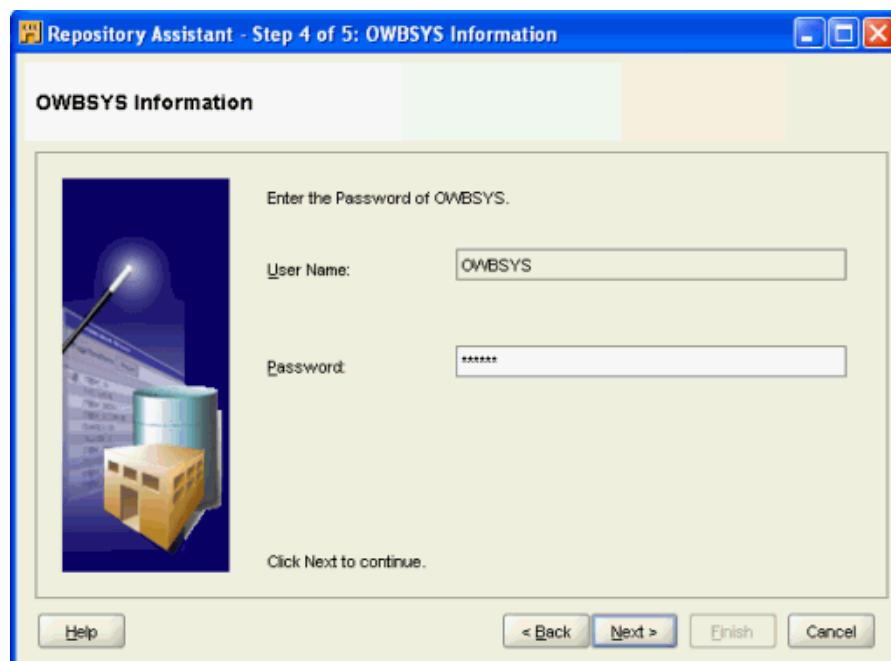
In Linux, from the command line, run
`OWB_HOME/owb/bin/unix/reposinst.sh`.

2. On the Welcome page of the Repository Assistant, click **Next**.
3. On the Database Information page, enter the host name, port number, and the service name where the OWB repository is installed. Click **Next**.
4. On the Choose Upgrade Operation page, select **Import entire repository from file**.



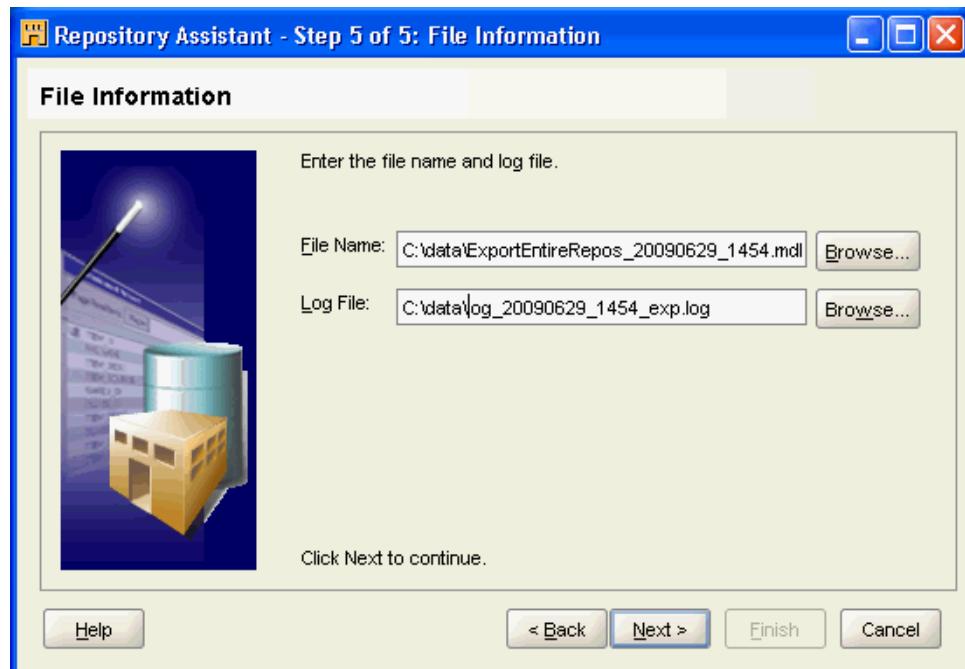
Click **Next**.

5. On the OWBSYS Information page, enter the OWBSYS password.



Click **Next**.

6. On the File Information page, enter or browse to find the name of the files used when exporting the repository.



Click **Next**.

7. On the Summary page, click **Finish**.

Note: When importing an OWB release 10.2 MDL repository export, the import process creates a workspace that has the same name as the OWB release 10.2 repository schema. When importing an OWB release 11.1 repository export, the import process creates one workspace for each workspace being imported from the OWB release 11.1 repository.

When importing an OWB release 10.2 MDL repository export, the import process creates a workspace that has the same name as the OWB release 10.2 repository schema. When importing an OWB release 11.1 repository export, the import process creates one workspace for each workspace being imported from the OWB release 11.1 repository.

Migrating the Repository to a Different Oracle Database Instance

The process of migrating the repository to a different Oracle Database instance involves the export of the existing repository to a file, and subsequent import of information from that file into a new repository on the other Oracle Database instance.

This process is virtually identical to upgrading from an OWB release 10.2 repository to a current version of the repository. The main differences are:

- Before the import, you must create users on the target instance with the same names as the OWB users on the original instance.
- At the time of importing, you must provide the credentials of a remote Oracle Database instance where the export files reside.

Note: Check the latest OWB release notes on <http://www.oracle.com/technetwork/developer-tools/warehouse/overview/index.html> before starting the migration process.

Exporting a Repository to MDL File

The export part of the scenario uses the Repository Assistant to create an MDL zip archive. This MDL zip archive includes information about all the workspaces in the repository, with each workspace stored as a separate file.

See Also: "Exporting an OWB Repository to a File" on page 6-10 for a description of the export operation

Creating OWB Users in the Target Instance

Before importing the repository, in the target Oracle Database instance that contains the migrated repository, create the same users who had access to the original repository. You must do this to ensure that the security information is correctly imported. However, you do not need to register these users as OWB users; the upgrade process handles this automatically.

Importing a Repository from MDL File

The import operation is described in section "Importing a Repository Export MDL File to OWB Release 11.2" on page 6-12. Note that in step 6, you are entering the credential of the source (remote) database, which contains the repository MDL and log files.

Upgrading Locations to OWB Release 11.2

The locations in your OWB repository must also be upgraded for compatibility with OWB 11.2 after the upgrade process. There are two ways to perform this upgrade:

- Use the script provided with OWB release 11.2.0.2 or higher. (This is the simpler procedure and is recommended.)
- Use the Repository Browser Locations Report to perform the upgrades one at a time. (This procedure is as effective as using the script, but requires more effort.)

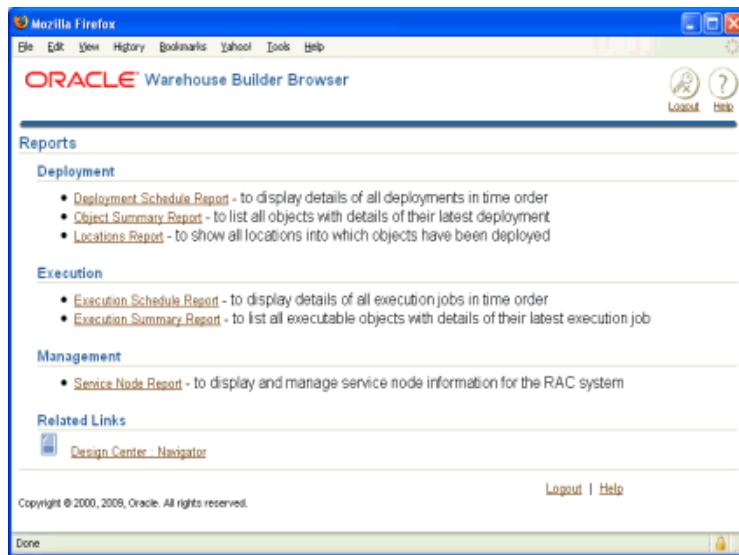
Alternatively, you can also un-register and re-register all locations. However, in this case you must also redeploy all your mappings to affected locations; therefore, this procedure is not recommended.

Upgrading Locations to OWB Release 11.2 Using the Repository Browser

You can also use Locations Reports from the Repository Browser to perform the locations upgrade process as follows.

To upgrade locations using the Repository Browser

1. In Repository Browser, select Control Center. Click **Reports**.
2. In the Reports window, under Deployment, click **Locations Reports**.



3. On the Locations Report, under Logical Details, in the Validation column of the table, click **Unknown**.

Logical Details							
<input type="button" value="Unregister Selected Locations"/> <input type="button" value="Select All"/> <input type="button" value="Select None"/>							
Select	Name	Type	Type Version	Service Description	User	Connect As	Latest Deployment
<input type="checkbox"/>	PlatformSchema	Control Center	0				Unknown
<input type="checkbox"/>	RTDC_USER_LOCATION	Oracle Database	11.2		RTDC_USER	2009-04-01 06:47:36	Unknown
<input type="checkbox"/>	PF_MOD_LOCATION	Oracle Workflow	2.6.4			2009-04-01 07:05:27	Unknown

4. Under Connection details, under Location Status, because the status is Unknown, enter the password and click **Get Status**.

Connection Details	
Property Name	Property Value
Schema	RTDC_USER *****
Password	<input type="password"/>
Host	<input type="text" value="Host Location"/>
Port	<input type="text" value="1521"/>
Service Name	<input type="text" value="Service Name or Location"/>
<input type="button" value="Update Details"/>	
Location Status	
Current Status	Unknown
Password	<input type="password"/>
<input type="button" value="Get Status"/>	

Note that the Current Status is Invalid.

5. Under Location Status, click **Upgrade**.

Connection Details	
Property Name	Property Value
Schema	RTDC_USER *****
Password	<input type="password"/>
Host	<input type="text" value="Host Location"/>
Port	<input type="text" value="1521"/>
Service Name	<input type="text" value="Service Name or Location"/>
<input type="button" value="Update Details"/>	
Location Status	
Current Status	Invalid (3 Error(s))
Password	<input type="password" value="*****"/>
<input type="button" value="Get Status"/>	
<input type="button" value="Upgrade"/>	

Note that the Current Status is now Valid.

6. Repeat steps 3 through 5 for all installations.

- Once you are finished, notice that the Validation column shows Tested locations

Logical Details							
<input type="button" value="Unregister Selected Locations"/> <input type="button" value="Select All"/> <input type="button" value="Select None"/>							
Select Name	Type	Type Version	Service Description	User	Connect As	Latest Deployment	Validation
<input type="checkbox"/> PlatformSchema	Control Center	0				2009-04-01 06:47:36	Unknown
<input type="checkbox"/> RTDC_USER_LOCATION	Oracle Database	11.2		RTDC_USER		2009-04-01 07:05:27	Tested
<input type="checkbox"/> PF_MOD_LOCATION	Oracle Workflow	2.6.4					Tested

Changes to Repository Objects When Moving to OWB Release 11gR2

Design-time objects in your projects are automatically modified during the upgrade process to be compatible with OWB release 11.2. This section lists the changes that are made.

Changes to OWB Release 10.2 and Release 11.1 Objects

When you move a design from OWB release 10.2 or OWB release 11.1 to OWB release 11.2, the following objects are modified during the migration or upgrade:

- Mapping Table Function Operators:** The mapping operators for table functions are upgraded to have only one input group and one output group. Old input groups that were configured as a REF cursor type are upgraded to a SYS_REF_CURSOR type, and a new construct operator is added to the map; its input group values are the same as in the REF cursor type input group, and the output group attribute is set to SYS_RECURSOR type.
- Mapping Key Lookup Operators:** The mapping operators for key lookups are upgraded to the new lookup operator.
- Business Presentation Modules:** Business presentation modules and their contents are discarded.
- BI Beans Locations:** BI beans locations are discarded.
- Data Profile Metadata:** Data profile metadata is not upgraded.

Changes to OWB Release 9.2 and Release 10.1 Objects

When you move a design from OWB release 9.2 or OWB release 10.1 to OWB release 11.2, the following objects are modified during the migration or upgrade:

- Control Centers and Locations:** After an upgrade, there is no association between the locations and the control centers that they reference. You must review the control center details using the Edit Control Center dialog box and select the locations associated with this control center.
- Locations and Control Center Connections:** Locations and Control Center connections are moved out of the projects that own them so that they can be shared across the entire workspace. Thus the statistics in the import log file displays an additional project for these objects.
- Mapping Operator Names:** The MDL Upgrade Utility ensures that the physical names and business names of all mapping operators are unique.
- Dimensions:** An associated dimension table is created with the same name as the dimension. The table contains the columns, constraints, and attribute sets defined in the Dimension Editor Table Properties of the dimension in the earlier release.

- **Cubes:** An associated cube table is created with the same name as the cube. The cube table contains columns, constraints, and attribute sets defined in the Cube Editor Table Properties of the cube in the earlier release.
- **Dimension and Cube Mapping Operators:** The mapping operators for dimensions and cubes are converted to table operators. These table operators use the physical tables created by the MDL Upgrade Utility for dimensions and cubes.
- **Mapping Display Sets for Dimension Hierarchies:** Any mapping sets originally created based on the named attribute set for a dimension hierarchy are removed. This is because display sets for dimension hierarchies are no longer automatically created and maintained.
- **Dimension Attributes:** For each level attribute upgraded, a dimension attribute with the same name is created, if it does not exist in the dimension.
- **Intelligence Objects and Reports:** In the earlier release, intelligence objects and reports were available only using OMB Plus scripting. These objects are not upgraded.
- **Advanced Queues:** An associated queue table is created based on the property AQ queue table name. The queue table created by the MDL File Upgrade Utility contains a column whose data type is the object type for that advanced queue.
- **Advanced Queue Operator in a Mapping:** Mapping Advanced Queue operators are changed to contain only one attribute called PAYLOAD. For Mapping Advanced Queue operators that are used as a source, a new Expand operator is added after the Mapping Advanced Queue operator. For Mapping Advanced Queue operators that are used as a target, a new Construct operator is added before the Mapping Advanced Queue operator.

Configuring Repository Browser, Control Center Agent, and Other Components

This section discusses how to configure the Repository Browser environment within OWB, and Control Center Agent for Oracle Application Server. Procedures for installing optional components are also described.

This section includes the following topics:

- ["Configuring the Repository Browser Environment" on page 7-1](#)
- ["Configuring Control Center Agent for Oracle Application Server" on page 7-2](#)
- ["Integrating with E-Business Suite" on page 7-26](#)
- ["Installing Third-Party Name and Address Data" on page 7-27](#)
- ["Integrating with Oracle Workflow" on page 7-28](#)

Configuring the Repository Browser Environment

The Repository Browser connects to Warehouse Builder repositories and enables you to view metadata, run Web reports, perform lineage and impact analysis on your metadata, and audit run-time executions.

When you install Warehouse Builder from Oracle Universal Installer, the Repository Browser is also installed and runs in all the languages that you selected during installation.

To verify the installation, start the Repository Browser listener and then the Repository Browser. For information on how to use the Repository Browser, refer to *Oracle Warehouse Builder Data Modeling, ETL, and Data Quality Guide*.

Changing the Session Timeout

By default, Repository Browser sessions time out after one hour of inactivity. To change this, edit the `web.xml` file located at
`OWB_HOME\owb\j2ee\applications\owbb\WEB-INF` by updating the `session-config` tag.

By default, the tag displays as follows:

```
<session-config>
<session-timeout>60</session-timeout>
</session-config>
```

Configuring Control Center Agent for Oracle Application Server

The procedures in this section describe how to install and configure Control Center Agent to run in Oracle Application Server. This section includes the following topics:

- [Modifying the OC4J Startup Script](#)
- [Configuring Control Center Agent Server-side Logging](#)
- [Configuring the Audit Data Source](#)
- [Setting Control Center Agent Security Roles](#)
- [Creating JMS Queues](#)
- [Setting the PATH Environment Variable](#)
- [Deploying the Control Center Agent to OC4J](#)
- [Configuring the Control Center Agent Web Service \(jrt-web.war\)](#)
- [Enabling Control Center Agent Audit Table Logging](#)
- [Installing JDBC Drivers for OC4J](#)
- [Overview of Control Center Agent Files](#)
- [Installing Code Templates for Use from Control Center Agent](#)

Modifying the OC4J Startup Script

OWB has several OC4J startup scripts for its components. Some are for starting OC4J for OWB Browser, and others are for starting OC4J for Control Center Agent.

To modify the OC4J startup script:

1. Ensure that you are in the application server configuration directory, *AS_HOME/opmn/config*. *AS_HOME* stands for the root home directory of the application server.
2. Open the file *opmn.xml* in a text editor.
3. Alter the contents of the file as demonstrated in the following example, where bold font indicates the edits you must make.

Note that:

- Setting `MaxPermSize` parameter at `128M` allocates enough `PermGen` space to OC4J to run the Control Center Agent.
- The `Python.path` sets the path for the Python library files used by the Control Center Agent: `jython_lib.zip` and `jython_owblib.jar`. These two files are in the *OWB_HOME/owb/lib/int* directory.
- The `km_security_needed` property determines if restrictions are applied to the types of operating system commands allowed by the OWB Code Template script that runs in the Control Center Agent. Setting it to `true` enforces restrictions.

To review the list of commands allowed by this security mechanism, check the file *AS_HOME/j2ee/home/config/ct_permissions.properties* (it can be specified in the *jrt.xml* configuration file), which is available after Control Center Agent successfully installs in the OC4J in Application Server.

```
<ias-component id="default_group">
  <process-type id="home" module-id="OC4J" status="enabled">
```

```

<module-data>
<category id="start-parameters">
<data id="java-options" value="-server
-XX:MaxPermSize=128M
-ms512M -mx1024M
-XX:AppendRatio=3
-Dpython.path=/home/lib/jython_lib.zip:/home/lib/jython_owbllib.jar
-Dkm_security_needed=true
...
</ias-component>
```

Configuring Control Center Agent Server-side Logging

To set up the Control Center Agent server-side logging:

1. Ensure that you are in the OC4J server configuration directory, *OC4J_HOME/j2ee/home/config/*. *OC4J_HOME* stands for the root home directory of the OC4J.
2. Open the file *j2ee-logging.xml* in a text editor.
3. Alter the contents of the file as demonstrated in the following example, where bold font indicates the edits you must make.

Note that:

- In the log handler section, the *jrt-internal-log-handler* is the handler used by Control Center Agent run-time logger to create log files.
- In the logger section, you are creating the logger for Control Center Agent run-time auditing.

```


<log_handler name="jrt-internal-log-handler"
  class="oracle.core.ojdl.logging.ODLHandlerFactory">
  <property name="path" value="..../log"/>
  <property name="maxFileSize" value="10485760"/>
  <property name="maxLogSize" value="104857600"/>
  <property name="encoding" value="UTF-8"/>
</log_handler>
...

<logger name="oracle.wh.jrt.server.audit" level="ALL"
  useParentHandlers="false">
  <handler name="jrt-internal-log-handler"/>
</logger>
```

Configuring the Audit Data Source

To configure the audit data source:

1. Ensure that you are in the OC4J server configuration directory, *OC4J_HOME/j2ee/home/config/*. *OC4J_HOME* stands for the root home directory of the OC4J.
2. Open the file *data-sources.xml* in a text editor.
3. Define the audit data source as demonstrated in the following example, where bold font indicates the edits you must make.

```

<managed-data-source name="AuditDS" connection-pool-name="OWBSYS Audit
Connection Pool" jndi-name="jdbc/AuditDS"/>
```

```

<connection-pool name="Oracle Audit Connection Pool">
  <connection-factory factory-class="oracle.jdbc.pool.OracleDataSource"
    user="username" password="password"
    url="jdbc:oracle:thin:@//localhost:1521/ORCL"/>
</connection-pool>

```

Setting Control Center Agent Security Roles

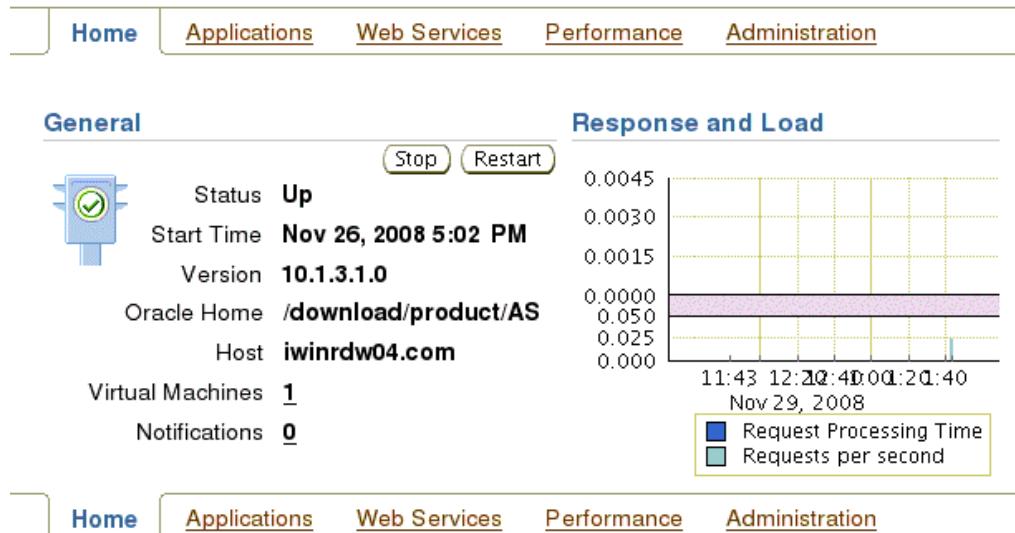
To configure Control Center Agent security roles:

1. In a web browser, navigate to the Enterprise Manager home page.
2. Log in using the oc4jadmin credentials. You may be prompted to activate the oc4jadmin account. If prompted to activate the account:
Create a password for the oc4jadmin account to activate it. Enter the password you want to use. Confirm the password and click **OK** to activate the account.
3. On the Cluster Topology page, locate the link to the OC4J instance installed under the Application Server.

Cluster Topology

Overview						
Hosts	1	Application Servers	1			
OC4J Instances	1	HTTP Server Instances	0			
Members						
View By	Application Servers					
	<input type="button" value="Start"/>	<input type="button" value="Stop"/>	<input type="button" value="Restart"/>			
	Select All	Select None	Expand All	Collapse All		
Select	Name	Status	Type	Host	CPU (%)	Memory (MB)
<input type="checkbox"/>	All Application Servers					
<input type="checkbox"/>	as1....com		Application Server	host04		
<input type="checkbox"/>	home (JVMs: 1)		OC4J		0.39	310.98

4. Click **home** (the OC4J instance).
This takes you to the home page of the OC4J instance.
5. On the OC4J home screen, click the **Administration** tab.

OC4J: home

[Home](#) [Applications](#) [Web Services](#) [Performance](#) [Administration](#)

6. On the Administration Tasks screen, expand **Security**.

7. Click the task icon next to **Security Providers**.

OC4J: home

Home Applications Web Services Performance Administration		
Expand All Collapse All		
Task Name	Go to Task	Description
▼ Administration Tasks		
► Properties		
► Services		
▼ Security		
Security Providers		Configure security providers, create/delete/...
Identity Management		Configure or change the Oracle Internet...
Instance Keystore		Configure the keystore and keys to be ...
Trusted SAML Authorities		Configure trusted SAML assertion issuer ...
► JMX		

[Home](#) [Applications](#) [Web Services](#) [Performance](#) [Administration](#)

8. On Security Providers screen, click **Instance Level Security**.

Security Providers**Instance Level Security**

You can configure the security attributes (realms, users & roles) for all applications deployed to this OC4J instance by clicking on the button below.

[Instance Level Security](#)

9. On Instance Level Security screen, click the **Realms** tab.

Instance Level Security

Security Provider Type **File-Based Security Provider**

Security Provider Attributes: File-Based Security Provider

<input type="button" value="General"/>	<input checked="" type="button" value="Realms"/>
--	--

Security Provider Type **File-Based Security Provider**

XML File Location **system-jazn-data.xml**

Default Realm **jazn.com**

TIP File location path is displayed relative to application deployment directory.

<input type="button" value="General"/>	<input checked="" type="button" value="Realms"/>
--	--

10. Under the **Realms** tab, in the default realm jazn . com, click the number in **Roles** column.

Instance Level Security

Security Provider Type **File-Based Security Provider**

Security Provider Attributes: File-Based Security Provider

<input type="button" value="General"/>	<input checked="" type="button" value="Realms"/>
--	--

Search

Name	<input type="button" value="Go"/>
------	-----------------------------------

Results

<input type="button" value="Create"/>	Realm Name	Roles	Users	Delete
	jazn.com	10	5	<input type="button" value="Delete"/>

<input type="button" value="General"/>	<input checked="" type="button" value="Realms"/>
--	--

11. On the Roles screen, click **Create**.

Roles

Security Provider Type **File-Based Security Provider**

Realm Name **jazn.com**

Search

Name	<input type="button" value="Go"/>
------	-----------------------------------

Results

<input type="button" value="Create"/>	Role Name	Users	Delete

12. On the Add Role screen, enter Name OWB_J2EE_EXECUTOR.

Click OK.

Add Role

Realm Name **jazn.com**

* Name **OWB_J2EE_EXECUTOR**

Grant RMI Login Permission

Grant Administration Permission

Cancel **OK**

13. On the Add Role screen, enter Name OWB_J2EE_OPERATOR.

Click OK.

Add Role

Realm Name **jazn.com**

* Name **OWB_J2EE_OPERATOR**

Grant RMI Login Permission

Grant Administration Permission

Cancel **OK**

14. On the Add Role screen, enter Name OWB_J2EE_OPERATOR.

Under Assign Roles, in the list of Available Roles, select OWB_J2EE_EXECUTOR and oc4j-administrators, and move to the list of Selected Roles.

Click OK.

Add Role

Realm Name **jazn.com**

* Name **OWB_J2EE_OPERATOR**

Grant RMI Login Permission

Grant Administration Permission

Assign Roles

A role may inherit from other roles. Select the roles you would like this role to inherit.

Available Roles	Selected Roles
ascontrol_admin ascontrol_appadmin ascontrol_monitor oc4j-app-administrators users	OWB_J2EE_EXECUTOR oc4j-administrators

Available Roles

ascontrol_admin
ascontrol_appadmin
ascontrol_monitor
oc4j-app-administrators
users

Selected Roles

OWB_J2EE_EXECUTOR
oc4j-administrators

Move
Move All
Remove
Remove All

15. On the Add Role screen, enter **Name** OWB_J2EE_ADMINISTRATOR.

Under Assign Roles, in the list of **Available Roles**, select OWB_J2EE_OPERATOR and OWB_J2EE_EXECUTOR, and move to the list of **Selected Roles**.

Click **OK**.

Add Role

Realm Name **jazn.com**

* Name **OWB_J2EE_ADMINISTRATOR**

Grant RMI Login Permission

Grant Administration Permission

Assign Roles

A role may inherit from other roles. Select the roles you would like this role to inherit.

Available Roles	Selected Roles
ascontrol_admin	OWB_J2EE_OPERATOR
ascontrol_appadmin	OWB_J2EE_EXECUTOR
ascontrol_monitor	
oc4j-app-administrators	
users	
oc4j-administrators	

Move **Move All** **Remove** **Remove All**

16. On the bread crumb, click **Instance Level Security**.

17. Under the **Realms** tab, in the default realm **jazn.com**, click the number in **Users** column.

18. On the Users screen, click user **oc4jadmin**.

Users

Security Provider Type **File-Based Security Provider**

Realm Name **jazn.com**

Search

Name **Go**

Results

User Name	Assigned Roles	Delete
oc4jadmin	OWB_J2EE_OPERATOR, oc4j-administrators*, OWB_J2EE_ADMINISTRATOR*, OWB_J2EE_EXECUTOR, oc4j-app-administrators*, ascontrol_admin*	

19. Under Assign Roles, in the list of **Available Roles**, select OWB_J2EE_ADMINISTRATOR and oc4j_app_administrators, and move to the list of **Selected Roles**.

- Click **Apply**.
20. On the breadcrumb, click **Instance Level Security**.
 21. Under the **Realms** tab, in the default realm jazn.com, click the number in **Roles** column.
 22. Click **Create**.
 23. On the Add Role screen, enter Name OWB_INTERNAL_USERS.
- Click **OK**.

Creating JMS Queues

You must create two JMS queues, for JRT:owbQueue, and for JRT:owb_abortQueue.

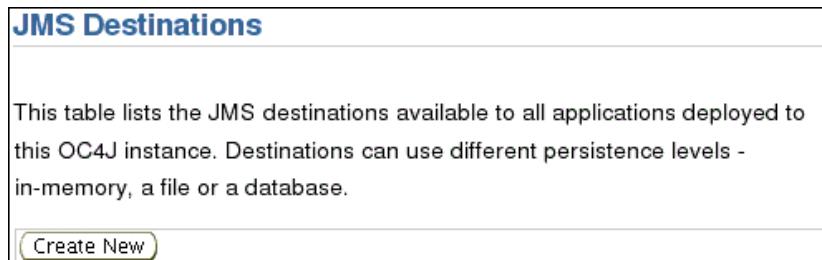
To create JMS Queues:

1. On the breadcrumb, click **OC4J:home**.
2. On the OC4J:home screen, click the **Administration** tab.
3. Expand **Services**, and then expand **Enterprise Messaging Service**.

OC4J: home					
Home	Applications	Web Services	Performance	Administration	
Expand All Collapse All					
Task Name	Go to Task		Description		
▼ Administration Tasks					
▶ Properties					
▼ Services					
JDBC Resources			Create/delete/view data sources and connection pools.		
▼ Enterprise Messaging Service					
JMS Destinations			Create/delete/edit JMS destinations.		
JMS Connection Factories			Configure JMS connection factories.		
In-Memory and File Based Persistence			Configure settings for in-memory and file based persistence.		
Database Persistence			Configure settings for database persistence.		
OracleAS JMS Router			Configure the JMS Router.		
JNDI Browser			Browse the JNDI bindings of this OC4J instance.		
Transaction Manager (JTA)			Configure and monitor transaction management capabilities.		
▶ Security					
▶ JMX					

4. Click the task icon next to **JMS Destinations**.

5. On JMS Destinations screen, click **Create New**.



6. On Add Destination screen, choose **Destination Type Queue**, enter **Destination Name** owbQueue, select **Persistence** In Memory Persistence Only, and enter **JNDI Location** jms/owbQueue.

Click **OK**.

The screenshot shows the "Add Destination" dialog. It includes fields for Destination Type (set to Queue), Destination Name (owbQueue), and Description. Below these, the "Persistence" section is expanded, showing the "In Memory Persistence Only" option selected, with the JNDI Location set to jms/owbQueue. Other persistence options like File Based Persistence and Database Based Persistence are shown but not selected. A table at the bottom lists resource providers, showing one entry: "No database persistence" under "Resource Provider Name" and "JMS providers found" under "Datasource JNDI Location".

Select	Resource Provider Name	Datasource JNDI Location
	No database persistence	JMS providers found.

7. On Add Destination screen, choose **Destination Type Queue**, enter **Destination Name** owb_abortQueue, select **Persistence** In Memory Persistence Only, and enter **JNDI Location** jms/owb_abortQueue.

Click **OK**.

Setting the PATH Environment Variable

To ensure that system commands such as `linux ls`, `sh`, and so on, can run normally in scripts within the Control Center Agent, the environment variable `PATH` must be held the directory that contains the system commands.

To set the PATH environment variable:

1. On the breadcrumb, click OC4J:home.
2. On the OC4J:home screen, click the Administration tab.
3. Expand Administration Tasks, then expand Properties.

OC4J: home			
Home Applications Web Services Performance Administration			
Expand All Collapse All			
Task Name	Go to Task	Description	
▼ Administration Tasks			
▼ Properties			
EJB Compiler Settings		Configure the EJB Compiler.	
J2EE Websites		Manage the J2EE websites in this OC4J instance.	
JSP Properties		Set JSP container properties.	
Logger Configuration		Set log levels for all Loggers.	
Thread Pool Configuration		Configure the thread pools of this OC4J instance.	
Shared Libraries		Manage the shared libraries of this OC4J instance.	
Server Properties		Configure server properties for this OC4J instance.	
► Services			
► Security			
► JMX			
Home Applications Web Services Performance Administration			

4. Click the task icon next to **Server Properties**.
5. On the Server Properties screen, scroll downwards to Environment Variables section.
6. Under Environment Variables, click **Add Another Row**.
7. Enter **PATH** in **Name**, and fill **Value** with directories that contain the system commands.

Click **Apply**.



Deploying the Control Center Agent to OC4J

To deploy the Control Center Agent to OC4J:

1. On the breadcrumb, click OC4J:home.
2. On the OC4J:home screen, click the Applications tab.
3. Click Deploy to begin deploying Control Center Agent.



4. On the Deploy: Select Archive screen, under Archive, select Archive is present on local host. Upload the archive to the server where Application Server Control is running.
Click Browse and locate the jrt.ear file in the *OWB_HOME/owb/jrt/applications* directory.

Deploy: Select Archive

Cancel Step 1 of 3 Next

Archive

The following types of archives can be deployed: J2EE application (EAR files), Web Modules (WAR files), EJB Modules (EJB JAR files) and Resource Adapter Modules (RAR files).

Archive is present on local host. Upload the archive to the server where Application Server Control is running.

Archive Location [Browse...](#)

Archive is already present on the server where Application Server Control is running.

Location on Server
The location on server must be the absolute path or the relative path from j2ee/home

Deployment Plan

The deployment plan is an XML file that contains the deployment settings for an application. If you do not have a deployment plan, one will be created automatically during the deployment process. Later in the deployment process, you can optionally edit the deployment plan and save it for a future deployment of this application.

Automatically create a new deployment plan.
The deployment plan settings will be based on OC4J defaults and information contained in the archive

Deployment plan is present on local host. Upload the deployment plan to the server where Application Server Control is running.

Plan Location [Browse...](#)

Deployment plan is already present on server where Application Server Control is running.

Location on Server
The location on server must be the absolute path or the relative path from j2ee/home

5. Under Deployment Plan, select **Automatically create a new deployment plan**.

Click **Next**.

While the upload is in progress, you see the Deploy: Uploading screen.

Deploy: Uploading

The following archive and deployment plan (if specified) are being uploaded:

Archive **/home/swu/jrt.ear**
Deployment Plan **Creating new plan**

 Uploading is in progress...

 **TIP** Wait while uploading is in progress.

6. On the Deploy: Application Attributes screen, enter **Application Name** **jrt**, and **Context Root** **jrt**. Leave the other attributes at their default values.

Click **Next**.

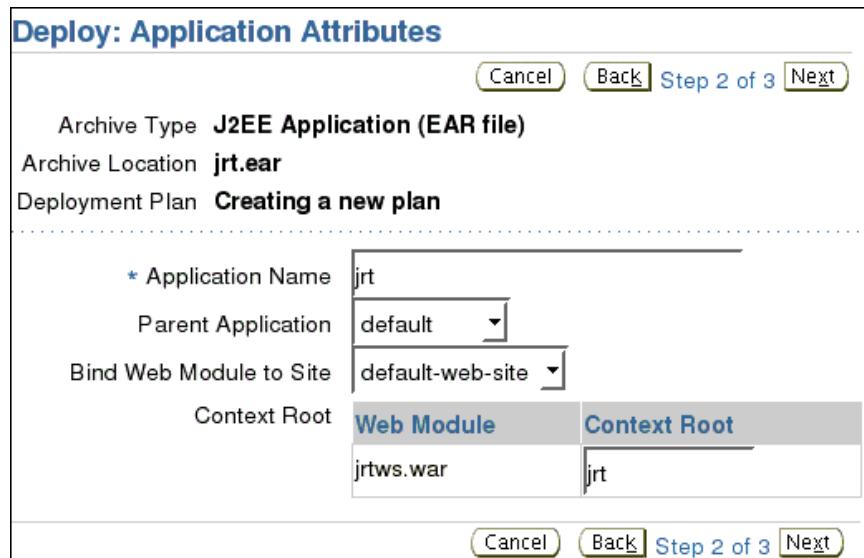
Deploy: Application Attributes

Archive Type **J2EE Application (EAR file)**
Archive Location **jrt.ear**
Deployment Plan **Creating a new plan**

* Application Name **jrt**
Parent Application **default**
Bind Web Module to Site **default-web-site**
Context Root

Web Module	Context Root
jrtws.war	jrt

Cancel **Back** **Step 2 of 3** **Next**



7. On Deploy: Deployment Settings screen, leave all attributes at their default values, and click **Deploy**.

Deploy: Deployment Settings

(Cancel) (Back) Step 3 of 3 (Deploy)

Archive Type J2EE Application (EAR file)	Application Name jrt
Archive Location jrt.ear	Parent Application default
Deployment Plan Creating a new plan	Bind Web Module to Site default-web-site
	Context Root jrt

Deployment Tasks

The table below provides a set of common deployment tasks you might want to perform for this application. Only those tasks that apply to the current application are enabled.

Task Name	Go To Task	Description
Map Environment References		Map any environment references in your application (for example, data sources) to physical entities currently present on the operational environment.
Select Security Provider		A security provider acts as the source for available users and groups when mapping security roles.
Map Security Roles		Map any security roles exposed by your application to existing users and groups. The list of users and groups is obtained from the security provider you selected for this application.
Configure EJBs		Configure the Enterprise JavaBeans in your application.
Configure Clustering		Configure clustering of your application.
Configure Class Loading		Manipulate the classpath of your application.

Advanced Deployment Plan Editing

Click Edit Deployment Plan to set more advanced deployment options. (Edit Deployment Plan)

Save Deployment Plan

After you make changes, you can save the deployment plan to your local disk. You can then use the saved deployment plan to redeploy this application later. (Save Deployment Plan)

- Once the application is successfully deployed, a confirmation message displays.

Confirmation

The Application "jrt" has been successfully deployed. (Return)

Progress Messages

```
[Nov 30, 2008 10:04:22 PM] Application Deployer for jrt STARTS.  

[Nov 30, 2008 10:04:22 PM] Copy the archive to /scratch/...  

[Nov 30, 2008 10:04:23 PM] Initialize /scratch/download/...  

[Nov 30, 2008 10:04:23 PM] Unpacking jrt.ear  

[Nov 30, 2008 10:04:24 PM] Done unpacking jrt.ear  

[Nov 30, 2008 10:04:24 PM] Unpacking jrtws.war
```

(Return)

9. On the Confirmation screen, click **Return**.

Under the Applications tab, expand **All Applications**, then expand **default**.

Note that the new application CCA is up.

Configuring the Control Center Agent Web Service (jrt-web.war)

The Control Center Agent Web Service is deployed automatically with Control Center Agent. No additional setup is required.

To check that the Web Service is running:

1. On the breadcrumb, click **OC4J:home**.
2. On the OC4J:home screen, click the **Web Services** tab.

Note that **JRTWebService** is listed.

OC4J: home

Select	Port Name	Web Service	Application	Status	Port Enabled	Start Time
<input checked="" type="radio"/>	HttpSoap11	JRTWebService	jrt	Up	✓	5:02:41 PM

To test the Web Service:

1. Under the Web Services tab, select **JRTWebService** and click **Test Service**.
2. On the Test Web Service: HttpSoap11 screen, under Discovered Websites, select the http site and click **Test Web Service**.

Test Web Service: HttpSoap11

Discovered Websites

This table shows websites that may be used to test this Web Service.

URL	Test Web Service
http://.../jrt/webservice	<input type="button" value="Test Web Service"/>

Select Listener

Select	Protocol	Host	Port
<input checked="" type="radio"/> J2EE Website	http		

TIP The URL field may be modified to test the web service using a custom URL.

3. The JRTWebService endpoint screen displays that the Web Service is running.

You can access the operations available from the Operation drop down box, fill in the required parameters, and then click **Invoke** to execute the operation.

JRTWebService endpoint

For a formal definition, please review the [Service Description](#).

Download the JavaScript Stub (*BETA*) for [HttpSoap11](#) and see its [documentation](#).

HttpSoap11

Operation : validateJobID HTML Form XML Source

Reliable Messaging Include In Header

WS-Security Include In Header

jobID	<input type="text"/>	xsd:string
operation	<input type="text"/>	xsd:string

Note: XML source view contents will not be reflected in the HTML form view

Show Transport Info
 Perform stress test Enable

Invoke

- [Optional] Alternatively, go to `http://AS_hostname/jrt/webservice`.

To use the Web Service with a secure Web site (available only to admin users):

- Create or modify the `secure-web-site.xml` file in the `config` directory.
- A sample `secure-web-site.xml` file is listed below.

Note that the relevant code is in bold typeface.

Particularly, ensure that `protocol` is set to `https`, `secure` is set to `true`, and a secure http port is identified. Additionally, you must create a keystore file and add the `ssl-config` entry (you must use an absolute path for the key store file).

```
<?xml version="1.0" standalone="yes"?>

<web-site
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:noNamespaceSchemaLocation=
    "http://xmlns.oracle.com/oracleas/schema/web-site-10_0.xsd"
  display-name="OC4J 10g (10.1.3) Secure Web Service"
  schema-major-version="10"
  schema-minor-version="0"
  protocol="https"
  port="4443"
  secure="true"
>
<ssl-config keystore="directory_path/owb/jrtrun/config/serverkeystore.jks"
  keystore-password="password" />

<default_web_app application="default" name="defaultWebApp" />
<web-app application="system" name="dms0" root="/dmsoc4j" access-log="false"
  load-on-startup="true" />
<web-app application="system" name="dms0" root="/dms0" access-log="false"
  load-on-startup="true" />
<web-app application="system" name="JMXSoapAdapter-web"
  root="/JMXSoapAdapter" />
<web-app application="default" name="jmsrouter_web" root="/jmsrouter"
  load-on-startup="true" />
```

```

<web-app application="javasso" name="javasso-web" root="/jsso"
  load-on-startup="false" />
<web-app application="ascontrol" name="ascontrol" root="/em"
  load-on-startup="true" ohs-routing="false"/>
<web-app application="jrt" name="jrtws" load-on-startup="true" root="/jrt" />

<!-- Access Log, where requests are logged to -->
<access-log path="../log/secure-web-access.log"
  split="day" />

<!-- Uncomment this if you want to use ODL logging capabilities
<odl-access-log path="../log/secure-web-access" max-file-size="1000"
  max-directory-size="10000" />
-->

</web-site>

```

3. After you create the `secure-web-site.xml` file, deploy it to `jrtws.war`, you may access the JRTWebService securely through `https://hostname:4443/jrt/webservice`.

Enabling Control Center Agent Audit Table Logging

To enable Audit Table logging, you must set up a connection pool, and then a managed data source prior to Control Center Agent deployment.

Configuring OWBSYS Audit Connection Pool

To configure the OWBSYS Audit Connection Pool:

1. On the breadcrumb, click **OC4J:home**.
2. On the OC4J:home screen, click the **Administration** tab.
3. Expand **Services**.

OC4J: home					
Home	Applications	Web Services	Performance	Administration	
Expand All Collapse All					
Task Name	Go to Task	Description			
▼ Administration Tasks					
▶ Properties					
▼ Services					
JDBC Resources		Create/delete/view data sources and connection pools.			
▶ Enterprise Messaging Service					
JNDI Browser		Browse the JNDI bindings of this OC4J instance.			
Transaction Manager (JTA)		Configure and monitor transaction management capabilities.			
▶ Security					
▶ JMX					

4. Click the task icon next to **JDBC Resources**.
5. On the JDBC Resources screen, under Connection Pools, click **Create**.

The screenshot shows the 'JDBC Resources' page. At the top, there is a dropdown menu labeled 'Application' with 'All' selected. Below it, under 'Data Sources', there is a 'Create' button. Under 'Connection Pools', there is also a 'Create' button.

6. On the Create Connection Pool - Application screen, under Application, select **default**.

Under Connection Pool Type, select **New Connection Pool**.

Click **Continue**.

The screenshot shows the 'Create Connection Pool - Application' screen. It has two tabs: 'Application' (selected) and 'Connection Pool Type'. The 'Application' tab shows a dropdown menu with 'default' selected. The 'Connection Pool Type' tab shows two radio buttons: 'New Connection Pool' (selected) and 'New Connection Pool from Existing Connection Pool'. Below the radio buttons is a description: 'Create a new connection pool that is configured like an existing connection pool.' A dropdown menu for 'Existing Connection Pool' contains the entry '"Example Connection Pool"'.

7. On the Create Connection Pool screen, enter the **Name** OWBSYS Audit Connection Pool. Under URL, select **JDBC URL**, and enter the URL (in this example, `jdbc:oracle:thin:@localhost:1521:ora111` is used for a local installation).

Under the Credential section, enter the User Name (in this example, `owbsys_audit`), and enter the **Use Cleartext Password**.

Create Connection Pool

[Cancel](#) [Back](#) [Finish](#)

[Home](#) [Attributes](#) [Proxy Interfaces](#)

* Name: OWBSYS Audit Connection Pool

* Connection Factory Class: oracle.jdbc.pool.OracleDataSource
Class must be available to the application's class loader.

URL

You can either specify a URL directly or have it generated from connection information. When you test a connection, the connection factory class and credentials specified on this page will be used to perform the test.

JDBC URL: jdbc:oracle:thin:@//localhost:1521/mydb.com [Test Connection](#)

Generate URL from Connection Information [Test Connection](#)

• • •

Credentials

TIP For OracleDataSources, credentials must be entered if not already specified in the URL.

Username: owbsys_audit

Use Cleartext Password
Password: *****

Use Indirect Password [?](#)
Indirect Password:
example: Scott, customers/Scott

8. [Optional] On the Confirmation screen, test the availability of the connection. Click **Test Connection**.

The screenshot shows the JDBC Resources screen with two sections: Data Sources and Connection Pools.

Data Sources:

Attributes							
Name	Application	JNDI Location	Connection Pool	Managed by OC4J	Test Connection	Delete	
"AuditDS"	default	jdbc/AuditDS	"OWBSYS Audit Connection Pool"	<input checked="" type="checkbox"/>			
"OracleDS"	default	jdbc/OracleDS	"Example Connection Pool"	<input checked="" type="checkbox"/>			

Connection Pools:

Name	Application	Connection Factory Class	Monitor Performance	Test Connection	Refresh Pool	Delete	
"Example Connection Pool"	default	oracle.jdbc.pool...					
"OWBSYS Audit Connection Pool"	default	oracle.jdbc.pool...					

9. On the Test Connection: "OWBSYS Audit Connection Pool" screen, click Test.

The dialog box is titled "Test Connection: 'OWBSYS Audit Connection Pool'". It contains fields for Username (owbsys_audit), Password (*****), URL (jdbc:oracle:thin:@//localhost:1521/mydb.com), and SQL Statement (select * from dual). It includes "Test" and "OK" buttons.

A confirmation screen appears.

Note: Application Server 10g is not case-sensitive when verifying Oracle Database credentials; it converts all letters to lower case before attempting to connect to the database. Because Oracle Database Release 11g R1 and subsequent releases are case sensitive, create user credentials in lowercase when using for Audit Table logging to prevent the Invalid username/password error message.

Configuring an Audit Table Data Source

To deploy Control Center Agent to OC4J, you must also configure the data source for the audit table. This data source is required by Control Center Agent; if not configured, the Control Center Agent deployment fails.

To configure the audit table data source:

1. On the breadcrumb, click OC4J:home.
2. On the OC4J:home screen, click the Administration tab.
3. Expand Services.

OC4J: home

Task Name	Go to Task	Description
Administration Tasks		
Properties		
Services		
JDBC Resources		Create/delete/view...
Enterprise Messaging Service		
JNDI Browser		Browse the JNDI bindings...
Transaction Manager (JTA)		Configure and monitor...
Security		
JMX		

4. Click the task icon next to **JDBC Resources**.
5. On the JDBC Resources screen, under Data Sources, click **Create**.

JDBC Resources

Application All

Data Sources
<input type="button" value="Create"/>
...
Connection Pools
<input type="button" value="Create"/>

6. On the Create Data Source - Application & Type screen:
 - Under Application, select **Application default**.
 - Under Data Source Type, select **Managed Data Source**.
 - Click **Continue**.

Create Data Source - Application & Type

Application

Select the application to which this new data source is to be added.

Application

Data Source Type

Managed Data Source
A managed data source is one where OC4J provides critical system infrastructure such as global transaction management, connection pooling, statement caching and error handling.

Native Data Source
A native data source is one that implements the java.sql.DataSource interface and does not make use of OC4J's connection pooling or statement caching capabilities. A native data source can only participate in local transactions.

New Data Source from Existing Data Source
Create a new data source that is configured like an existing data source.

Existing Data Source

7. On the Create Data Source - Managed Data Source screen,
 - Enter **Name** AuditDS.
 - Enter **JNDI Location** jdbc/AuditDS.
 - Click **Finish**.

If you do not require audit table logging, leave other parameters at their default values.

Create Data Source - Managed Data Source

Application **default**

* Name	AuditDS
* JNDI Location	jdbc/AuditDS
Transaction Level	Global & Local Transactions
Connection Pool	OWBSYS Audit Connection Pool
* Login Timeout (seconds)	0

Installing JDBC Drivers for OC4J

The following instructions demonstrate how to install JDBC drivers for OC4J.

To install the JDBC drivers for OC4J:

1. Obtain the *.jar file that contains the JDBC driver for your desired platform.
2. Stop the OC4J service.
3. Copy the JDBC driver file into the *AS_HOME/j2ee/home/applib* directory, where *AS_HOME* is the Application Server home.
4. Restart the OC4J service.

Overview of Control Center Agent Files

This section describes the files generated at run time. Note that if your application name is other than jrt, it appears in these directory paths as *app_name*.

The Control Center Agent files include:

- "ODL Job Log Files" on page 7-24
- "Control Center Agent Run-Time Log File" on page 7-24
- "JobID File" on page 7-24
- "Control Center Agent Settings" on page 7-25
- "JRTAdministratorMBean" on page 7-25

ODL Job Log Files

By default, ODL Job log files are located in the *AS_HOME/j2ee/home/log/app_name* directory.

Each job log resides in its own directory. For example, log file for job 10 is in *AS_HOME/j2ee/home/log/app_name/job10/log.xml*, and the metadata file for job 10 in *AS_HOME/j2ee/home/log/app_name/job10/meta_data.txt*.

You may choose to configure Control Center Agent to store the log files in other directories; ensure that Control Center Agent can access these log directories.

Control Center Agent Run-Time Log File

The Control Center Agent run-time log file *app_name.log* employs the ODL log file rotation mechanism. The current run-time log is always *app_name.log*, while older log files are *app_name.log.N*, where *N* is the round of log rotation). By default, this file is:

AS_HOME/j2ee/home/log/app_name.log

JobID File

Control Center Agent uses a file to keep track of the current job ID in the system; be careful not to remove this file. This file can be in one of two possible locations:

- When the Control Center Agent is installed as part of the larger OWB installation, the file is at:
OWB_HOME/owb/jrt/application-deployments/jrt/counter/counter.info
- When the Control Center Agent is installed on the Application Server, the file is at:
AS_HOME/j2ee/home/applications/application_name/counter/counter.info

Control Center Agent Settings

Control Center Agent settings are persisted in the file `jrt.xml`. You may modify the settings in this file directly, and then restart Control Center Agent. Alternatively, you can modify these settings by changing the attributes of the [JRTAdministratorMBean](#) in Enterprise Manager. This file is:

`AS_HOME/j2ee/home/applications/app_name/config/jrt.xml`

JRTAdministratorMBean

The [JRTAdministratorMBean](#) enables you to dynamically change Control Center Agent settings without restarting the Control Center Agent.

To change Control Center Agent settings with the JRTAdministratorMBean

1. On the breadcrumb, click [OC4J:home](#).
2. On the OC4J:home screen, click the **Applications** tab.
3. Expand **All Applications**, then expand **default**.
4. In the **jrt** row, click the task icon in column **Application Defined MBeans**.



The screenshot shows the Oracle Application Server Control Center interface. At the top, there are buttons for Start, Stop, Restart, Undeploy, Redeploy, and Deploy. Below these are links for Select All, Select None, Expand All, and Collapse All. The main area is a table titled "Select Name" with columns: Status, Start Time, Active Requests, Request Processing Time (seconds), Active EJB Methods, and Application Defined MBeans. The table lists several applications: "All Applications" (Status: Up, Start Time: Nov 26, 2008 5:02:47 PM PDT, Active Requests: 0, Processing Time: 0.00, EJB Methods: 0, MBeans: 0), "ascontrol" (Status: Up, Start Time: Nov 26, 2008 5:02:47 PM PDT, Active Requests: 0, Processing Time: 0.00, EJB Methods: 0, MBeans: 0), "default" (Status: Up, Start Time: Nov 26, 2008 5:02:47 PM PDT, Active Requests: 0, Processing Time: 0.00, EJB Methods: 0, MBeans: 0), "jrt" (Status: Up, Start Time: Nov 30, 2008 8:17:23 PM PDT, Active Requests: 0, Processing Time: 0.00, EJB Methods: 0, MBeans: 0), and "Middleware Services" (Status: Up, Start Time: N/A, Active Requests: 0, Processing Time: 0.00, EJB Methods: 0, MBeans: 0). Each row has a task icon in the last column.

Select	Name	Status	Start Time	Active Requests	Request Processing Time (seconds)	Active EJB Methods	Application Defined MBeans
<input type="checkbox"/>	All Applications						
<input type="checkbox"/>	ascontrol		Nov 26, 2008 5:02:47 PM PDT	0	0.00	0	
<input type="checkbox"/>	default		Nov 26, 2008 5:02:47 PM PDT	0	0.00	0	
<input type="checkbox"/>	jrt		Nov 30, 2008 8:17:23 PM PDT	0	0.00	0	
<input type="checkbox"/>	Middleware Services						

5. On the Application MBeans screen, in the left-hand side navigation menu, select [JRTAdministrator](#).

The right-hand side shows all the Attributes and Operations of JRTAdministrator MBean, under the corresponding tabs.

The screenshot shows the Oracle Warehouse Builder interface for managing Application MBeans. On the left, a tree view lists 'Application:jrt' (expanded), 'JRT' (expanded), 'JRTAdministrator' (selected and highlighted in blue), 'MBeanServer:jrt' (collapsed), and 'MBeanServer:system' (collapsed). A search bar at the top has 'MBean Name' selected and a 'Find' button. On the right, the 'MBean: JRT:JRTAdministrator' details page is displayed. It shows the name as 'jrt:type=JRT,name=JRTAdministrator,vendor=OWB,J2EEApplication=jrt' and a description: 'Provide access to the properties and operations of JRTAdministratorMBean via Enterprise Manager'. Below this are tabs for 'Attributes (18)' and 'Operations (6)'. The 'Attributes (18)' tab is active, showing a table with two rows:

Name	Description	Access	Value
CacheJobStatusSize	Attribute CacheJobStatusSize controls the maximum number of job status to be stored in the in-memory job log cache in Control Center Agent. Obviously the bigger this number is the more memory Control Center Agent will take up. The attribute value is between 1 and 10000. Default value: 3000.	RW	3000
CacheLogNo	Attribute CacheLogNo controls the maximum number of job logs to be stored in the in-memory job log cache in Control Center Agent. Obviously the bigger this number is the more memory Control Center Agent will take up. The attribute value is between 1 and 100. Default value: 10.	RW	10

6. Change the attributes to desired values, then click **Apply**.

Installing Code Templates for Use from Control Center Agent

If you want to use a code template, you must install any additional JDBC driver files that it requires to access heterogeneous systems on all the client computers and on the computer that hosts the Control Center Agent.

To install a Code Template:

1. Load the required JDBC drivers to the *OWB_HOME/owb/lib/ext* directory.
2. Import the code templates into any projects that uses them, in the design client.

You only import the code template once for the project. After it is imported, all users working on the project can use the code template.

Integrating with E-Business Suite

When integrating with Oracle E-Business Suite (EBS), you must first import its metadata, and then design mappings for moving and transforming data. Specifically:

- During the design phase, Oracle Warehouse Builder users must have access to metadata in the APPS schema.
- During the execution phase, Oracle Warehouse Builder users must access the data in the APPS schema.

Because direct access to the APPS production schema may be limited and restricted, you may define a user on the EBS database for Oracle Warehouse Builder users to access only the relevant metadata and data.

To enable access to EBS data and metadata:

1. Create a user on the database that hosts the EBS. This user needs, at minimum, CONNECT and RESOURCE privileges.
2. Grant access to the relevant metadata by running the `OWB_ORACLE_HOME\owb\cmi\ebs\owbebs.sql` script.

This script grants access to the following tables in the APPS schema, which contain metadata for EBS tables, views, sequences, and keys:

- `FND_APPLICATION`
- `FND_APPLICATION_VL`
- `FND_TABLES`
- `FND_VIEWS`
- `FND_SEQUENCES`
- `FND_COLUMNS`
- `FND_PRIMARY_KEYS`
- `FND_FOREIGN_KEYS`
- `FND_PRIMARY_KEY_COLUMNS`
- `FND_FOREIGN_KEY_COLUMNS`

The script also creates a synonym in the user schema for each of the preceding objects.

3. Enable a user to extract data from the EBS database.

You can create a new user, or grant the appropriate privileges to the same user you created earlier. For each object from which you want to enable data extraction, grant this user at least SELECT access.

Warehouse Builder users can now import the E-Business Suite metadata as described in the importing section of *Oracle Warehouse Builder Data Modeling, ETL, and Data Quality Guide*.

Installing Third-Party Name and Address Data

Warehouse Builder gives you the option to perform name and address cleansing on your data with the Name and Address operator. The Name and Address operator identifies and corrects errors and inconsistencies in name and address source data. The operator identifies inconsistencies by comparing input data to data libraries supplied by the third-party name and address cleansing software vendors. Purchase the data libraries directly from these vendors.

To install data libraries, refer to the installation instructions of the name and address cleansing software vendor of your choice. For the list of certified name and address cleansing software providers, refer to Oracle Technology Network at <http://www.oracle.com/technetwork/developer-tools/warehouse/overview/index.html>.

To integrate with third-party name and address data libraries:

1. Install Warehouse Builder as instructed in this guide.
2. Purchase data libraries from one of the certified vendors listed on My Oracle Support at <https://support.oracle.com>.

3. Install and access the certified vendor's data libraries and Name and Address adapter following the vendor's instructions.

If you are installing in a Real Application Cluster environment, then you may be able to install the name and address adapter on many nodes to benefit from the parallelism and failover enabled by the RAC architecture. Check with your vendor to see if your purchase license allows a multiple-node installation.

You do not need to install the data libraries on multiple nodes. However, if you install all data libraries on one node, then performance may suffer due to file access time latency. Follow the recommendations of your name and address cleansing software vendor.

4. Design a mapping using the Name and Address operator to cleanse name or address data. Refer to *Oracle Warehouse Builder Data Modeling, ETL, and Data Quality Guide* for information on designing mappings using the Name and Address operator.

Integrating with Oracle Workflow

If you plan to use Warehouse Builder process flows, then use Oracle Workflow to enable deployment as described in this section.

See Also: *Oracle Warehouse Builder Data Modeling, ETL, and Data Quality Guide* for more information about schedules

To enable integration with Oracle Workflow:

1. Stop all running OC4J instances.
2. Locate Oracle Workflow installation program, at *OWB_HOME\owb\wf\install\wfinstall.**.
If you installed Oracle Warehouse Builder 11g on Oracle Database 10g Release 2, separately download the software for Oracle Workflow 2.6.4.
3. Start Oracle Workflow installation program.

For Windows, enter the following at the command prompt:

```
C:\> cd OWB112\owb\wf\install  
C:\OWB112\owb\wf\install> wfinstall.bat
```

For Linux, enter the following in the shell:

```
% cd OWB112/owb/wf/install  
% wfinstall.csh
```

Note that if you use multiple Oracle databases, you must unset the environment variables related to Oracle, such as ORACLE_HOME, ORACLE_SID, ORACLE_PORT, and LD_LIBRARY_PATH.

4. Complete the Workflow Configuration Assistant using following values:
 - **Install Option:** Server Only
 - **Workflow Account:** *owf_mgr*
 - **Workflow Password:** *wf_password*
 - **SYS Password:** *sys_password*; use the SYS password for the database where installing Oracle Workflow.

- **TNS Connect Descriptor:** *hostname:port:service_name*, where the values correspond to your database.
Because the assistant does not reference the `tnsnames.ora` file, do not use a net service name.
 - **LDAP Parameters:** this is optional; see Oracle Workflow documentation for details.
 - **Mailer Parameters:** this is optional; see Oracle Workflow documentation for details.
 - **Tablespace:** this is optional; you may change the tablespace.
- 5.** Click **Submit** to start the Workflow configuration process.
- The configuration process can take several minutes. Check `OWB112/owb/wf/install/wf.log` for messages to follow the progress of the configuration process.
- When the process is complete, the Workflow Configuration Assistant displays a message of completion.
- 6.** Install the Workflow Client [Optional]
- The installation of Oracle Workflow client is optional because the Process Flow Editor in Warehouse Builder replaces its functionality. You may want to install Oracle Workflow client if you want to view the deployed Warehouse Builder processes in Oracle Workflow.
- On the computer that hosts the Warehouse Builder client, install Oracle Workflow client from Oracle Workflow client CD.
- 7.** Create a Workflow Proxy User.
- When the Workflow instance is not on the same computer that hosts the Warehouse Builder repository, you need to create a proxy user.
- Within the database hosting the repository, use SQL Plus to create a user and grant it the `OWB_USER` role as a default. This enables the remote Oracle Workflow instance to connect to the services provided by the Control Center.

Considerations for the OWB_USER Role

If you encounter a problem when trying to grant the `OWB_USER` role, then it is possible that `OWB_USER` already exists in an earlier OWB implementation. In this case, you must first rename the existing `OWB_USER` to something else before you can grant a user the `OWB_USER` role in the current release. In order to retain the properties for the renamed role, you must register it in the new installation of release 11.2.

Part II

Administering Oracle Warehouse Builder

This part contains the following sections:

- [Chapter 8, "Managing Configurations"](#)
- [Chapter 9, "Managing the Control Center and Repository"](#)
- [Chapter 10, "Managing the Control Center Agent"](#)
- [Chapter 11, "Using Snapshots to Manage Metadata"](#)
- [Chapter 12, "Using the Metadata Loader"](#)
- [Chapter 13, "Managing Security"](#)

Managing Configurations

This section discusses the design issues associated with multiple configuration environments.

This section includes the following topics:

- [About Configuration Management](#)
- [Creating Configurations Templates](#)
- [Editing Configuration Templates](#)
- [Activating Configurations](#)
- [Configuring Objects](#)

About Configuration Management

When you install Warehouse Builder and create a new repository, you have a default project with a single automatic default configuration, DEFAULT_CONFIGURATION. See [Figure 8–1](#) and note that each new project has an automatic DEFAULT_CONFIGURATION. You can observe this by expanding the Configurations node in Project Navigator. Note also that the active configuration appears on the toolbar as DEFAULT_CONFIGURATION.

This default configuration contains the default physical properties of all the data objects in the project; these properties are necessary for deployment. As a single entity, the DEFAULT_CONFIGURATION enables deployment to the default control center, DEFAULT_CONTROL_CENTER, and default agent, DEFAULT_AGENT, which are necessary to the internal workings of Warehouse Builder.

When you define a new configuration, OWB adds it to the list of configurations. With multiple configurations, you can specify how the same logical object may be defined across various environments: you can define multiple physical properties and deployment details for a single table, view, or other such object. In this manner, configurations enable you to connect design objects to their realizations.

If you link a configuration to a template, the configuration picks up the default values of the configuration properties from this template; enabling you to override the existing set of configuration values with your own values.

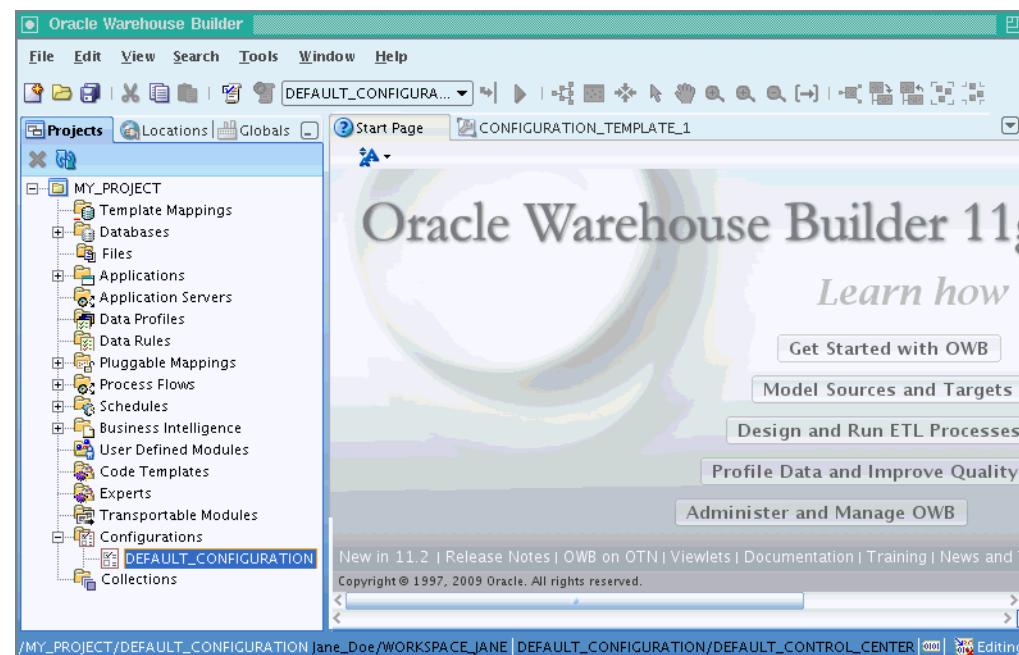
Configurations have several roles in the OWB environment:

- The key role of a configuration is to determine which Control Center Service and Control Center Agent manage the deployment and the deployment location. A configuration is always associated with only one Control Center and CCA, and specifies which of the services controls the deployment of any objects when this configuration is active. The configuration settings of the design objects control the

Location property. For example, each Oracle Database module has a configuration property Location; the value of the Location property must be a reference to an Oracle Database location, and it determines where any objects inside the module are deployed.

- The other role of configurations is to capture information specific to the creation and use of an object for a specific location. For example, you can define the tablespace where a new table should be created.

Figure 8–1 Oracle Warehouse Builder Configuration Management



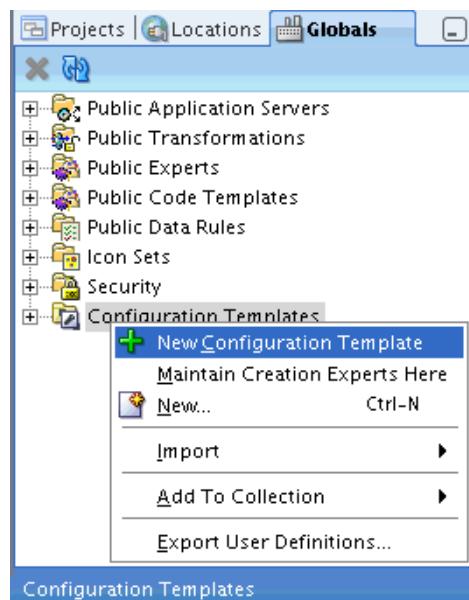
Configuration details are grouped by type, such as performance parameters, storage space for tables, and so on. Because each object type can have a number of configuration properties that are not used or have general defaults, OWB uses Configuration Templates to simplify how much the user needs to do.

Configuration Templates may be used to override automatic default values for configuration properties and to flag the properties for either the general set or the advanced set. For example, one may move the less frequently used configuration properties to the advanced set.

Creating Configurations Templates

To create a configuration template:

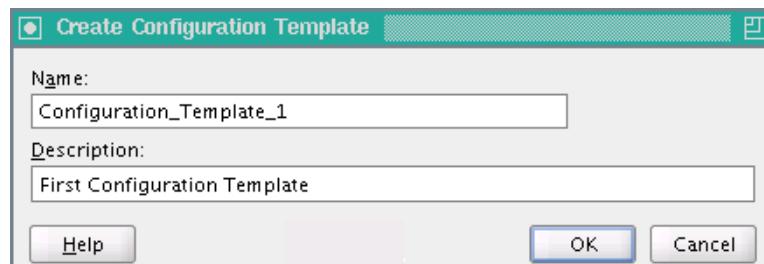
1. In the Globals Navigator, select Configuration Templates.
2. Right-click Configuration Templates, and select New Configuration Template.



The Create Configuration Template wizard opens. Note that on every screen of the wizard, you have the option of clicking **Help** to get more information.

3. On the Name and Description screen, enter the **Name** of the new configuration template, and optionally, its **Description**.

Click **OK**.



4. The new Configuration Template appears in the right pane of the OWB console.

Note that when you create a new configuration template, its default setting of advanced options is initially set to be identical to previously created configuration templates.

Property	Advanced	Seeded Default	CONFIGURATION_TEMPLATE_1
+ Databases			
+ Applications			
+ Application Server			
+ Data Profile			
+ Process Flow Module			
+ Business Intelligence			

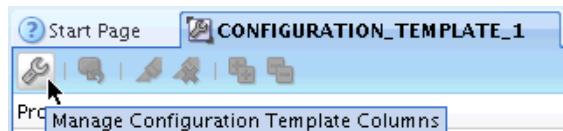
Editing Configuration Templates

To edit the configuration values in a template, use the Configuration Template Editor, which allows you to set the configuration properties of objects to Advanced, and to override the existing configuration defaults.

Note that the configuration template editor can show the different templates side by side, to make it easier to compare and edit multiple templates.

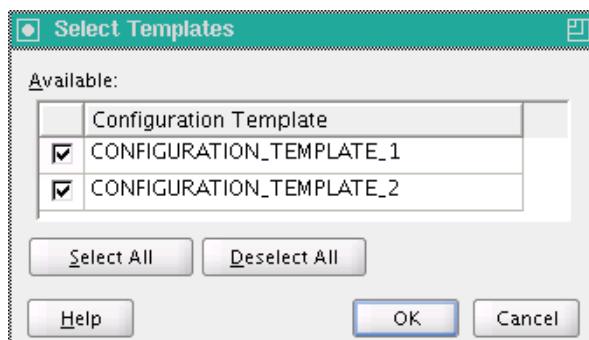
To edit a configuration template:

1. In the Globals Navigator, under Configuration Templates, click the **Manage Configuration Template Columns** icon.



2. In the Select Templates window, select the templates you want to edit.

Click **OK**.



3. Click the **Expand All** icon.



4. Make changes to the side-by-side templates.

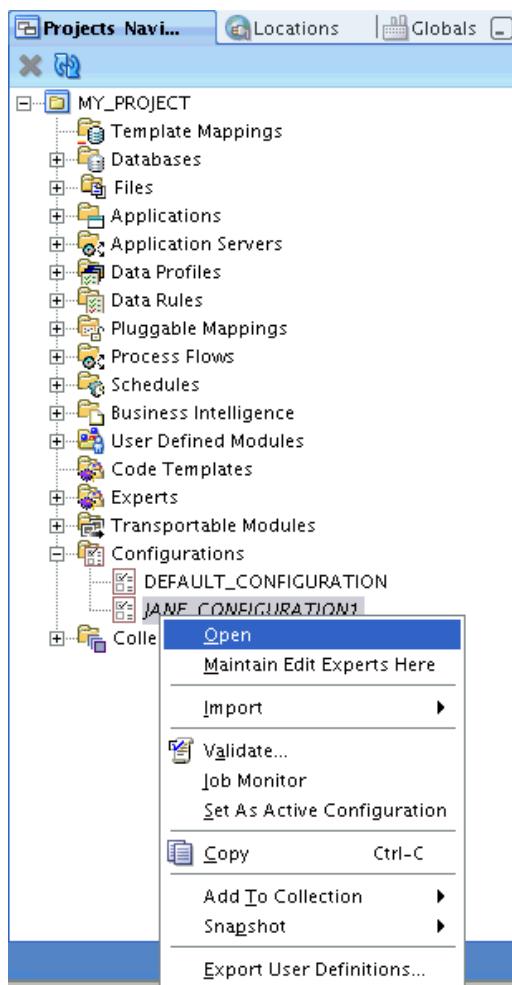
The screenshot shows the 'CONFIGURATION_TEMPLATE_1' editor window. The left pane displays a navigation tree with nodes like 'Databases', 'DB2', 'DB2 Module', 'GENERATION_TARGET_DIRECTORIES', 'Sequence', 'Table', 'Change Data Capture', 'Transformations', 'Function', 'Generation Comments', and 'Identification'. The right pane is a grid table with columns: 'Property', 'Advanced', 'Seeded Default', 'CONFIGURATION_TEMPLATE_2', and 'CONFIGURATION_TEMPLATE_1'. The 'CONFIGURATION_TEMPLATE_1' column is highlighted in green. The 'Change Data Capture' section is expanded, showing rows for 'Enable' (checked), 'Table Position' (value 0), and 'Function'.

Property	Advanced	Seeded Default	CONFIGURATION_TEMPLATE_2	CONFIGURATION_TEMPLATE_1
Databases				
DB2				
DB2 Module				
GENERATION_TARGET_DIRECTORIES				
LOADER Directory	<input type="checkbox"/>	ctl\	ctl\	ctl\
LOADER Extension	<input type="checkbox"/>	.ctl		
LOADER Run Parameter File	<input type="checkbox"/>	_run.ini		
Sequence				
Table				
Change Data Capture				
Enable	<input checked="" type="checkbox"/>	false	true	true
Table Position	<input type="checkbox"/>	0		
Transformations				
Function				
Generation Comments	<input type="checkbox"/>			
Identification				

Activating Configurations

To activate a configuration:

1. In the Project Navigator, select a project and expand its navigation tree.
2. Expand the Configurations folder.
3. Select a configuration.
4. Right-click the selected configuration; from the menu, select **Set As Active Configuration**.



All changes that you make to the configuration parameters of objects are saved in this configuration. If you switch to the previous configuration, these parameters show the settings of that configuration.

The Control Center service and Control Center Agent associated with this configuration are now set; they manage all future deployments or executions.

Configuring Objects

When you choose the Configure option for any object, it shows the set of configuration properties with the default values that are defined in the Configuration Template. You can expand the Property tree to see all properties.

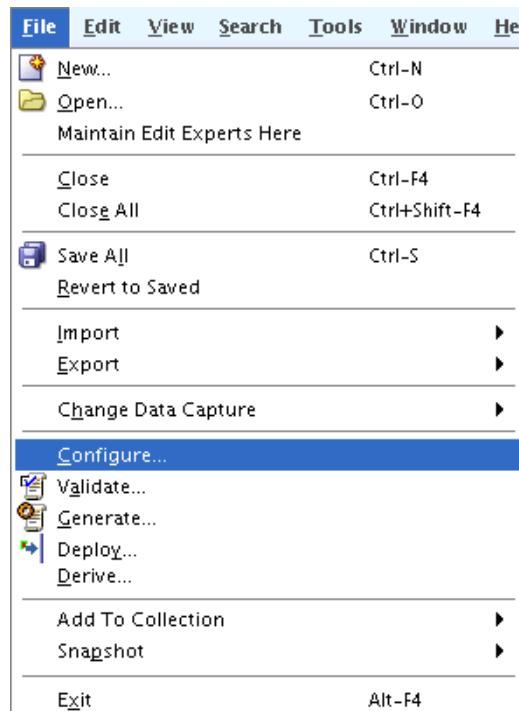
Configuration defines the physical characteristics of data objects. These may include defining a tablespace and setting performance parameters in the configuration of a table.

Note that you may change the configuration of an object at any time before deployment, and that it is possible to define multiple configurations for the same set of objects. This is particularly useful when deploying to multiple environments, such as test and production.

All objects have a parameter `deployable`, which is selected by default. To prevent an object from being deployed, deselect this parameter.

To configure an object:

1. In the Project Navigator, select the object to configure.
2. In the File menu, click **Configure**.



The Configuration editor is displayed.

3. Select a parameter to display and edit its value in the right-side panel.

The screenshot shows the Oracle Configuration editor interface. The title bar says 'Configuration of ORACLE_1'. The left pane shows a tree view with nodes: ORACLE_1, Deployment System Type, Generation Preferences, Generation Target Directories, and Identification. The 'Identification' node is expanded, showing parameters: Application Short Name (WB), Deployable (true), Location (ORACLE_1_LOCATION1(Default)), Main Application Short Name (ora), Streams Administrator Location (ORACLE_1_LOCATION1(Default)), and Top Directory (..\\..\\codegen\\). The 'Location' parameter is currently selected.

Property	Value
Application Short Name	WB
Deployable	true
Location	ORACLE_1_LOCATION1(Default)
Main Application Short Name	ora
Streams Administrator Location	ORACLE_1_LOCATION1(Default)
Top Directory	..\\..\\codegen\\
Run Time Directories	
Tablespace Defaults	

4. Click **Help** for additional information.

Managing the Control Center and Repository

Users often need to manage OWB Control Center service and repository. To do this, you can use the utilities in the *OWB_HOME/owb/rtp/sql* directory by accessing them at the command prompt. This section describes the available resources and tools, and the necessary management tasks.

This section contains the following topics:

- [Overview of OWB Control Center and Repository Management](#)
- [OWB Administrative Utilities](#)

Overview of OWB Control Center and Repository Management

The OWB Control Center, Control Center service, and OWB repository occasionally require administration tasks that go beyond the OWB Control Center Manager GUI.

These tasks can be divided into the following groups:

- [Starting and Stopping the Control Center Service](#)
- [Displaying and Managing Control Center Configuration](#)
- [Managing Control Center Deployment and Execution Jobs](#)
- [OWB Repository and Location Administration Utilities](#)

Details about each group of tasks and the associated commands are provided in the following sections.

See also:

- An architectural overview including each of the major OWB architectural components in *Oracle Warehouse Builder Concepts*
- Process of submitting deployment and execution requests to the Control Center Service in *Oracle Warehouse Builder Concepts*
- Information on deployment and execution jobs in *Oracle Warehouse Builder Concepts*
- Commands used for managing the Control Center Agent, described in [Chapter 10, "Managing the Control Center Agent"](#)

Starting and Stopping the Control Center Service

A group of commands starts and stops the control center service, displays its current availability, and troubleshoots a range of issues.

This group includes the following commands:

- ["service_doctor"](#) on page 9-8 displays a variety of OWB repository and Control Center service diagnostics.
- ["show_service"](#) on page 9-9 displays the status of the Control Center service as either Available or Not Available.
- ["start_service"](#) on page 9-12 starts a Control Center service.
- ["stop_service"](#) on page 9-12 stops a Control Center service.

Displaying and Managing Control Center Configuration

Another group of commands displays and modifies the properties of the Control Center, thereby modifying its behavior.

This group includes the following commands:

- ["display_platform_property"](#) on page 9-5 displays the value of a platform property
- ["set_ccs_home"](#) on page 9-8 sets the properties associated with a Control Center Service
- ["set_platform_property"](#) on page 9-8 sets the value of a platform property
- ["set_platform_property_from_file"](#) on page 9-9 sets the value of a platform property from a specified file location

Managing Control Center Deployment and Execution Jobs

The third group of commands overrides default handling of an OWB deployment or execution requests by the Control Center, deactivates or aborts a job, or expedites handling of a specific request.

This group includes the following commands:

- ["abort_exec_request"](#) on page 9-3 terminates a run that is currently busy.
- ["abort_unit_request"](#) on page 9-3 terminates a deployment job at the unit level.
- ["deactivate_all"](#) on page 9-4 deactivates all incomplete deployment and execution jobs.
- ["deactivate_deployment"](#) on page 9-4 deactivates a deployment job.
- ["deactivate_execution"](#) on page 9-4 deactivates an execution job.
- ["expedite_exec_request"](#) on page 9-5 expedites a running job
- ["list_requests"](#) on page 9-6 lists the details of any active deployment or execution requests
- ["print_exec_details"](#) on page 9-6 prints the audit execution hierarchy and details about executions to a text file
- ["print_running_exec_details"](#) on page 9-7 prints the audit execution hierarchy and details about the incomplete executions to a text file
- ["sqlplus_exec_background_template"](#) on page 9-9 runs the specified object in the background, as a non-blocking call.

- "sqlplus_exec_template" on page 9-11 runs the specified object as a blocking call

OWB Repository and Location Administration Utilities

The OWB repository and locations sometimes require direct manipulation. Also, run-time audit data accumulates in the OWB repository, and some users may want to purge historical run-time audit data, to improve performance.

This fourth group includes the following commands:

- "[delete_warehouse_object](#)" on page 9-5 deletes a warehouse object from a location in the repository
- "[grant_error_table_privileges](#)" on page 9-6 enables the Repository Browser to view and purge error tables
- "[grant_upgrade_privileges](#)" on page 9-6 grants the required roles and privileges for upgrade deployment actions to a specific user
- "[UpdateControlCenter](#)" on page 9-13 sets the host, port, and service parameters for the Control Center in the OWB Client Repository.
- "[UpdateLocation](#)" on page 9-13 sets the host, port, and service parameters for the location in the OWB Client Repository.
- "[purge_audit_template](#)" on page 9-7 purges either deployment or execution audit data
- "[reset_repository](#)" on page 9-8 resets the registration details of an OWB repository
- "[rrepos_report](#)" on page 9-8 displays the details of the Control Center and its registered locations
- "[set_repository_password](#)" on page 9-9 sets the password for the OWB repository

OWB Administrative Utilities

The following information is an alphabetical list of OWB administrative utilities.

abort_exec_request

The `abort_exec_request.sql` script terminates a run that is currently busy.

Connect as `workspaceOwner` or `workspaceUser` with administrative privileges. Run the script:

```
SQL> @drive:OWB_HOME/owb/rtp/sql/abort_exec_request.sql auditId workspace;
```

- For an `auditId` that represents a running job, this script aborts the job cleanly. The `auditId` may be obtained by running the `list_requests.sql` script; see "[list_requests](#)" on page 9-6.
- The `workspace` is the workspace where the job is running, and should be declared as `workspaceOwner.workspaceName`. If using only the `workspaceName`, `workspaceOwner` defaults to current user.

abort_unit_request

The `abort_unit_request.sql` script terminates a deployment job at the unit level. A deployment unit is a collection of objects that are being deployed to the same location with the same deployment action.

Connect as *workspaceOwner* or *workspaceUser* with administrative privileges.
Run the script:

```
SQL> @drive:OWB_HOME/owb/rtp/sql/abort_unit_request.sql auditId workspace;
```

- For an *auditId* that represents a deployment unit job, this script aborts the job cleanly. The *auditId* may be obtained by running the *list_requests.sql* script; see "["list_requests"](#)" on page 9-6.
- The *workspace* is the workspace where the job is running, and should be declared as *workspaceOwner.workspaceName*. If using only the *workspaceName*, *workspaceOwner* defaults to current user.

deactivate_all

The *deactivate_all.sql* script deactivates all incomplete deployment and execution jobs.

Connect as *workspaceOwner* or *workspaceUser* with administrative privileges.
Run the script:

```
SQL> @drive:OWB_HOME/owb/rtp/sql/deactivate_all.sql workspace;
```

- The *workspace* is the workspace where the job is running, and should be declared as *workspaceOwner.workspaceName*. If using only the *workspaceName*, *workspaceOwner* defaults to current user.

deactivate_deployment

The *deactivate_deployment.sql* script deactivates a deployment job.

Connect as *workspaceOwner* or *workspaceUser* with administrative privileges.
Run the script:

```
SQL> @drive:OWB_HOME/owb/rtp/sql/deactivate_deployment.sql auditId workspace;
```

- For an *auditId* that represents a deployment job, this script deactivates the job and all its unit jobs cleanly. The *auditId* may be obtained by running the *list_requests.sql* script; see "["list_requests"](#)" on page 9-6.
- The *workspace* is the workspace where the job is running, and should be declared as *workspaceOwner.workspaceName*. If using only the *workspaceName*, *workspaceOwner* defaults to current user.

deactivate_execution

The *deactivate_execution.sql* script deactivates an execution job.

Connect as *workspaceOwner* or *workspaceUser* with administrative privileges.
Run the script:

```
SQL> @drive:OWB_HOME/owb/rtp/sql/deactivate_execution.sql auditId workspace;
```

- For an *auditId* that represents an execution job, this script deactivates the job cleanly. The *auditId* may be obtained by running the *list_requests.sql* script; see "["list_requests"](#)" on page 9-6.
- The *workspace* is the workspace where the job is running, and should be declared as *workspaceOwner.workspaceName*. If using only the *workspaceName*, *workspaceOwner* defaults to current user.

delete_warehouse_object

The `delete_warehouse_object.sql` script deletes a warehouse object.

Connect as `workspaceOwner` or `workspaceUser` with administrative privileges.
Run the script:

```
SQL> @drive:OWB_HOME/owb/rtp/sql/delete_warehouse_object.sql object location workspace;
```

- The named object is deleted.
- The location is where the object may be found.
- The workspace is the workspace where the job is running, and should be declared as `workspaceOwner.workspaceName`. If using only the `workspaceName`, `workspaceOwner` defaults to current user.

display_platform_property

The `display_platform_property.sql` script displays the value of a platform property. These properties control the behavior of the control center service.

Connect as `workspaceOwner` or `workspaceUser` with administrative privileges.
Run the script:

```
SQL> @drive:OWB_HOME/owb/rtp/sql/display_platform_property.sql property;
```

- The script assumes that at most one `property` with the given name exists. To set the property, see "["set_platform_property"](#) on page 9-8.

expedite_exec_request

The `expedite_exec_request.sql` script expedites a running job by moving it to the top of the list of pending jobs.

Connect as `workspaceOwner` or `workspaceUser` with execute, deploy, and administrative privileges. Run the script:

```
SQL> @drive:OWB_HOME/owb/rtp/sql/expedite_exec_request.sql authId returnNumber returnCode workspace;
```

- For an `auditId` that represents a deployment job, this script expedites the job cleanly. The `auditId` may be obtained by running the `list_requests.sql` script; see "["list_requests"](#) on page 9-6.
- The `returnNumber` and `returnCode` are the result values assigned to the job, and have the following values:
 - 0 = OK
 - 1 = OK_WITH_WARNINGS
 - 3 = FAILURE
- The workspace is the workspace where the job is running, and should be declared as `workspaceOwner.workspaceName`. If using only the `workspaceName`, `workspaceOwner` defaults to current user.

grant_error_table_privileges

The `grant_error_table_privileges.sql` script enables the Repository Browser to view and purge error tables. It grants SELECT and DELETE privileges on the specified table to the OWBSYS user.

Connect as the user who owns the error table. Run the script:

```
SQL> @drive:OWB_HOME/owb/rtp/sql/grant_error_table_privileges.sql tableName;
```

- In this script, `tableName` is the name of the error table, such as `TABLE1_ERR`.

grant_upgrade_privileges

The `grant_upgrade_privileges.sql` script grants the required roles and privileges for upgrading deployment actions to a specific user.

Connect as `workspaceOwner` or `workspaceUser` with execute, deploy, and administrative privileges, such as `SYS`. Run the script:

```
SQL> @drive:OWB_HOME/owb/rtp/sql/grant_upgrade_privileges.sql userName;
```

- In this script, `userName` is the name of the schema or user who has the new upgrade privileges.

list_requests

The `list_requests.sql` script lists the details of any active deployment or execution requests. A deployment is a collection of deployment units, which are collections of objects deployed by the same deployment action to the same location. Each deployment or execution is uniquely identified by an audit ID that may be used as a parameter to other SQL scripts, such as "["deactivate_deployment"](#) on page 9-4, ["deactivate_execution"](#) on page 9-4, and so on.

Connect as `workspaceOwner` or `workspaceUser` with execute, deploy, and administrative privileges. Run the script:

```
SQL> @drive:OWB_HOME/owb/rtp/sql/list_requests.sql workspace;
```

- The `workspace` is the workspace where the job is running, and should be declared as `workspaceOwner.workspaceName`. If using only the `workspaceName`, `workspaceOwner` defaults to current user.

print_exec_details

The `print_exec_details.sql` script prints the audit execution hierarchy and details about the various executions associated with an `auditId` to a file called `exec_details.txt`.

Connect as `workspaceOwner` or `workspaceUser` with execute, deploy, and administrative privileges. Run the script:

```
SQL> @drive:OWB_HOME/owb/rtp/sql/print_exec_details.sql auditId workspace;
```

- The `auditId` represents an execution job about which this script print the job details. The `auditId` may be obtained by running the script `list_requests.sql`; see "["list_requests"](#) on page 9-6.
- The `workspace` is the workspace where the job is running, and should be declared as `workspaceOwner.workspaceName`. If using only the `workspaceName`, `workspaceOwner` defaults to current user.

print_running_exec_details

The `print_running_exec_details.sql` script prints the audit execution hierarchy and details about the various incomplete executions that were started after the specified date associated with an `auditId` to a file called `exec_details.txt`.

Connect as `workspaceOwner` or `workspaceUser` with execute, deploy, and administrative privileges. Run the script:

```
SQL> @drive:OWB_HOME/owb/rtp/sql/print_running_exec_details.sql auditId workspace;
```

- The `auditId` represents an execution job about which this script prints the job details. The `auditId` may be obtained by running the script `list_requests.sql`; see "list_requests" on page 9-6.
- The `workspace` is the workspace where the job is running, and should be declared as `workspaceOwner.workspaceName`. If using only the `workspaceName`, `workspaceOwner` defaults to current user.

purge_audit_template

The `purge_audit_template.sql` script purges either deployment audit data or execution audit data.

Connect as `workspaceOwner` or `workspaceUser` with administrative privileges. The script is of the form:

```
SQL> @drive:OWB_HOME/owb/rtp/sql/purge_audit_template.sql
      workspace
      {DEPLOYMENT | EXECUTION}
      {sql_predicate | ALL | DATE_RANGE} [start_date end_date];
```

- The `workspace` is the workspace where the job is running, and should be declared as `workspaceOwner.workspaceName`. If using only the `workspaceName`, `workspaceOwner` defaults to current user.
- The `audit_type` is one of:
 - `DEPLOYMENT` for deployment audit data; see columns in `ALL_RT_AUDIT_DEPLOYMENTS` view
 - `EXECUTION` for execution audit data; see columns in `ALL_RT_AUDIT_EXECUTIONS` view
- The condition can be one of:
 - `sql_predicate`, such as "`number_script_run_errors > 0`"; this must comply with SQL syntax rules
 - `ALL`, for purging all deployment or execution data
 - `DATE_RANGE`, for purging data within a specified data range
- If `DATE_RANGE` is used, `start_date` is the start of date range (such as `TO_DATE('01-Jan-08')`) and `end_date` is the end of the date range (such as `SYSDATE`).

There are several ways to invoke this script, as demonstrated in these examples:

```
SQL> @drive:OWB_HOME/owb/rtp/sqlpurge_audit_template.sql
      workspace DEPLOYMENT ALL null null
```

```
SQL> @drive:OWB_HOME/owb/rtp/sqlpurge_audit_template.sql
      workspace EXECUTION "number_task_errors > 0" null null
```

```
SQL> @drive:OWB_HOME/owb/rtp/sqlpurge_audit_template.sql
      workspace EXECUTION_DATE_RANGE TO_DATE('01-Jan-2008') SYSDATE
```

reset_repository

The `reset_repository.sql` script resets the registration details for an OWB repository. It also stores the specified password as the password for the repository; see "["set_repository_password"](#)" on page 9-9. Note that after the registration details are reset, you must register the passwords for all previously registered locations.

Connect as repository owner, OWBSYS. Run the script:

```
SQL> @drive:OWB_HOME/owb/rtp/sql/reset_repository.sql password
```

rtrepos_report

The `rtrepos_report.sql` script displays the details of the Control Center and its registered locations. It provides information before using the `location_util.bat` script to alter some of the stored credentials.

Connect as repository owner, OWBSYS. Run the script:

```
SQL> @drive:OWB_HOME/owb/rtp/sql/rtrepos_report.sql
```

service_doctor

The `service_doctor.sql` script displays diagnostics about the OWB repository and its Control Center Service.

Connect as repository owner, OWBSYS. Run the script:

```
SQL> @drive:OWB_HOME/owb/rtp/sql/service_doctor.sql
```

set_ccs_home

The `set_ccs_home.sql` script sets the properties associated with a Control Center Service to recognize the location of the where to pick up the Instance Number, Version and Server Side Home. These parameters are not included in an OWB Installation when the "Allow Local Control Center Service" option is set.

Connect as repository owner, OWBSYS. Run the script:

```
SQL> @drive:OWB_HOME/owb/rtp/sql/set_ccs_home.sql instance version server_home
```

- The `instance` is the service node instance that you are updating
- The `version` is the OWB version that you are using
- The `server_home` is the OWB Home on the server computer, where the OWB install components are located.

set_platform_property

The `set_platform_property.sql` script sets the value of a platform property. These properties controls the behavior of the Control Center Service.

Note that the script assumes that the specified property must exist and its value is mutable.

Connect as repository owner, OWBSYS. Run the script:

```
SQL> @drive:OWB_HOME/owb/rtp/sql/set_platform_property.sql
```

property *property_value*

- The *property* is the name of the property that is changed
- The *property_value* is the name of the new value of the property
- To return a list of the properties, execute the following query:

```
select property_path from wb_rt_platform_properties where is Mutable <> 0
```

set_platform_property_from_file

The *set_platform_property_from_file.sql* script sets the value of a platform property to the contents of the *file_name* found in the *directory*. These properties controls the behavior of the Control Center Service.

Note that the script assumes that the specified property must exist and its value is mutable.

Connect as Control Center owner. Run the script:

```
SQL> @drive:OWB_HOME/owb/rtp/sql/set_platform_property_from_file.sql
      property directory file_name
```

- The *property* is the name of the property that is changed.
- The *directory* is the path where the new value for the property can be found.
- The *file_name* is the name of the file that contains the new value of the property.

set_repository_password

The *set_repository_password.sql* script sets the password for the OWB repository. This password is used by the Control Center Service at startup time.

Connect as repository owner, OWBSYS. Run the script:

```
SQL> @drive:OWB_HOME/owb/rtp/sql/set_repository_password.sql password
```

- The *password* is the new password value.

show_service

The *show_service.sql* script displays the status of the Control Center Service, which provides access to the deployment and execution features of an OWB repository. The valid return values for the status of the Control Service Center are Available or Not Available. To change the status of the Control Center Service, use the SQL scripts [start_service](#) on page 9-12 and [stop_service](#) on page 9-12 as appropriate.

Connect as repository owner, OWBSYS. Run the script:

```
SQL> @drive:OWB_HOME/owb/rtp/sql/show_service.sql
```

sqlplus_exec_background_template

The *sqlplus_exec_background_template.sql* script runs the specified object in the background, not blocking the caller. To get the task status and return result, check public views, of the type *ALL_RT_**. Use the script "[sqlplus_exec_template](#)" on page 9-11 to run the object in the foreground, which blocks the caller until the completion of the task. In its unchanged form the script takes the three keys required to identify the executable task. The task runs in the background with the default

parameters configured prior to deployment. The `custom_params` (defined on the object in the OWB Designer) and `system_params` values override the default input parameters of the task. The script returns the following values: 1 if task reports SUCCESS, 2 if WARNING, 3 if ERROR.

Connect as `workspaceOwner` or `workspaceUser` with execute privilege. The script is of the form:

```
SQL> @drive:OWB_HOME/owb/rtp/sql/sqlplus_exec_background_template.sql
      workspace
      location_name
      {PLSQLMAP | SQLLOADERCONTROLFILE | PROCESSFLOW | ABAPFILE |
       DATAAUDITOR | SCHEDULEDJOB}
      [parent/]task_name
      system_params
      custom_params
```

- The `workspace` is nominated in the parameters, and should be declared as `workspaceOwner.workspaceName`; if using only `workspaceName`, `workspaceOwner` defaults to user.
- The `location_name` is the physical name of the location to which the task is deployed.
- The `task_type` has the following options:
 - `PLSQLMAP` is used for OWB PL/SQL mapping.
 - `SQLLOADERCONTROLFILE` is used for OWB SQL*Loader mapping.
 - `PROCESSFLOW` is used for OWB ProcessFlow.
 - `ABAPFILE` is used for OWB SAP mapping.
 - `DATAAUDITOR` is used for OWB DataAuditor mapping.
 - `SCHEDULEDJOB` is used for an OWB scheduled job.
 - `CTMAPPING` is used for OWB Code Template mapping.
- The `task_name` is the physical name of the deployed object; it can be optionally qualified by the name of the deployed parent, like the ProcessFlow package name. A module name cannot be used because it is not a deployable object.
- The `custom_params` (defined on the object in the OWB Designer) and `system_params` values override the default input parameters of the task

If you encounter problems accessing public views, then connect to SQL*Plus using your repository user or owner credentials, and issue:

```
Set role OWB_USER
```

Run `@start_access_workspace.sql workspaceName workspaceOwner`. You can then access the public views.

There are several ways to invoke this script; note that `system_params` and `custom_params` are not used in these examples:

```
SQL> @drive:OWB_HOME/owb/rtp/sql/sqlplus_exec_background_template.sql
      workspace warehouse PLSQL mapping , , , ,
```

```
SQL> @drive:OWB_HOME/owb/rtp/sql/sqlplus_exec_background_template.sql
      workspace platform_schema SQL_LOADER load , , , ,
```

```

SQL> @drive:OWB_HOME/owb/rtp/sql/sqlplus_exec_background_template.sql
      workspace workflow PROCESS process_1 "," "," ,

SQL> @drive:OWB_HOME/owb/rtp/sql/sqlplus_exec_background_template.sql
      workspace platform_schema SAP sap_1 "," "," ,

SQL> @drive:OWB_HOME/owb/rtp/sql/sqlplus_exec_background_template.sql
      workspace warehouse DATA_AUDITOR data_auditor_1 "," "," ,

```

Note: The comma character can be escaped using the backslash character (\,); likewise the backslash character can be escaped by using two backslash characters (\\\). A single-quote character must be replaced by a sequence of four single-quote characters ('''').

sqlplus_exec_template

The `sqlplus_exec_template.sql` script runs the specified object and blocks the caller until the task is completed. To get the task status and return result, check public views of the type `ALL_RT_*`. Use the script "["sqlplus_exec_background_template"](#)" on page 9-9 to run the object in the background, without blocking the caller throughout the completion of the task. The script takes the three keys required to identify the executable task. The task runs with the default parameters configured prior to deployment. The `custom_params` (defined on the object in the OWB Designer) and `system_params` values override the default input parameters of the task. The script returns the following values: 1 if task reports SUCCESS, 2 if WARNING, 3 if ERROR.

Connect as `workspaceOwner` or `workspaceUser` with execute privilege. The script is of the form:

```

SQL> @drive:OWB_HOME/owb/rtp/sql/sqlplus_exec_template.sql
      workspace
      location_name
      {PLSQLMAP | SQLLOADERCONTROLFILE | PROCESSFLOW | ABAPFILE |
       DATAAUDITOR | SCHEDULEDJOB}
      [parent/] task_name
      system_params
      custom_params

```

- The `workspace` is nominated in the parameters, and should be declared as `workspaceOwner.workspaceName`; if using only `workspaceName`, `workspaceOwner` defaults to user.
- The `location_name` is the physical name of the location to which the task is deployed.
- The `task_type` has the following options:
 - `PLSQLMAP` is used for OWB PL/SQL mapping.
 - `SQLLOADERCONTROLFILE` is used for OWB SQL*Loader mapping.
 - `PROCESSFLOW` is used for OWB ProcessFlow.
 - `ABAPFILE` is used for OWB SAP mapping.
 - `DATAAUDITOR` is used for OWB DataAuditor mapping.
 - `SCHEDULEDJOB` is used for an OWB scheduled job.
 - `CTMAPPING` is used for OWB Code Template mapping.

- The `task_name` is the physical name of the deployed object; it can be optionally qualified by the name of the deployed parent, like the ProcessFlow package name. A module name cannot be used because it is not a deployable object.
- The `custom_params` (defined on the object in the OWB Designer) and `system_params` values override the default input parameters of the task

There are several ways to invoke this script; note that `system_params` and `custom_params` are not used in these examples:

```
SQL> @drive:OWB_HOME/owb/rtp/sql/sqlplus_exec_template.sql  
      workspace warehouse PLSQL mapping "," "," ,"  
  
SQL> @drive:OWB_HOME/owb/rtp/sql/sqlplus_exec_template.sql  
      workspace platform_schema SQL_LOADER load "," "," ,"  
  
SQL> @drive:OWB_HOME/owb/rtp/sql/sqlplus_exec_template.sql  
      workspace workflow PROCESS process_1 "," "," ,"  
  
SQL> @drive:OWB_HOME/owb/rtp/sql/sqlplus_exec_template.sql  
      workspace platform_schema SAP sap_1 "," "," ,"  
  
SQL> @drive:OWB_HOME/owb/rtp/sql/sqlplus_exec_template.sql  
      workspace warehouse DATA_AUDITOR data_auditor_1 "," "," ,"
```

Note: The comma character can be escaped using the backslash character (\,); likewise the backslash character can be escaped by using two backslash characters (\\\). A single-quote character must be replaced by a sequence of four single-quote characters (' ''').

start_service

The script `start_service.sql` starts a Control Center Service that is managed by the database. A Control Center Service, when available, provides access to the deployment and execution features of an OWB repository. To determine if the Control Center Service is available, use the [show_service](#) script on page 9-9; the return values are either Available or Not Available. To stop the service, use the [stop_service](#) script on page 9-12. When the service is invoked using this script, a monitoring job is created; it restarts the service if it fails, or if the database is brought back online. Alternatively, a Control Center Service can be started in a different `OWB_HOME`, on another host; the service is available only when that host is available and provided the service has been started and not stopped.

Connect as repository owner, OWBSYS. Run the script:

```
SQL> @drive:OWB_HOME/owb/rtp/sql/start_service.sql
```

stop_service

The script `stop_service.sql` stops a Control Center Service that is managed by the database. A Control Center Service, when available, provides access to the deployment and execution features of an OWB repository. To determine if the Control Center Service is available, use the [show_service](#) script on page 9-9; the return values are either Available or Not Available. To start the service, use the [start_service](#) script on page 9-12. When the service is stopped using this script, the monitoring job is removed; the service remains disabled until it is re-started by the [start_service](#) script.

Connect as repository owner, OWBSYS. Run the script:

```
SQL> @drive:OWB_HOME/owb/rtp/sql/stop_service.sql
```

UpdateControlCenter

This SQL script sets the host:port:service parameters for a control center in the OWB Client Repository. The connection type must be host:port:service.

UpdateLocation

This SQL script sets the host:port:service parameters for a location in the OWB Client Repository. The connection type must be host:port:service. The location may be registered or unregistered.

Managing the Control Center Agent

This section describes how to start and stop the Control Center Agent (CCA), and how to administer it using the `cca_admin` utility. Note that the CCA manages deployment and execution of the mappings that are based on Code Templates. CCA is not part of the Control Center service hosted within the Oracle Database.

This section contains the following topics:

- [Starting and Stopping the Control Center Agent](#)
- [About the `cca_admin` Utility](#)
- [Starting the `cca_admin` Utility](#)
- [`cca_admin` Commands](#)

See Also:

- [Chapter 7, "Configuring Repository Browser, Control Center Agent, and Other Components"](#), and specifically section ["Configuring Control Center Agent Server-side Logging"](#) on page 7-3 if you have your own Oracle Application Server and want to install the CCA there
- [Chapter 7, "Configuring Repository Browser, Control Center Agent, and Other Components"](#) for procedures that modify the installation defaults

Starting and Stopping the Control Center Agent

This section describes how to start and stop the CCA.

Note that when you invoke the `ccastart` for the first time, it prompts you to enter a password for the OC4J administrator account. The default username for this account is `oc4jadmin`.

To start the CCA:

- *In Linux*, in the directory `OWB_HOME/owb/bin/unix/`, run the script `ccastart`.
- *In Windows*, in the directory `OWB_HOME\owb\bin\win32\`, run the script `ccastart.bat`.

To stop the CCA:

- *In Linux*, in the directory `OWB_HOME/owb/bin/unix/`, run the script `ccashut`. This script accepts an optional argument `password`.

- In Windows, in the directory *OWB_HOME\owb\bin\win32*, run the script `ccashut.bat`. This script accepts an optional argument *password*.

For a representation of architectural structures of this installation, see [Figure 1–5, "Control Center Agent Topology" on page 1-6](#).

For procedures that modify installation defaults, see [Chapter 7, "Configuring Repository Browser, Control Center Agent, and Other Components"](#).

About the cca_admin Utility

The `cca_admin` utility changes Control Center Agent settings dynamically, without the need to shut down and subsequently restart the run-time environment. `cca_admin` commands retrieve the values of various environment attributes, set these to new values, and invoke the operations of `JRTAdministratorMBean`, introduced in section ["JRTAdministratorMBean" on page 7-25](#).

The typical `cca_admin` session includes the following steps:

- Running the `cca_admin` script, as described in ["Starting the cca_admin Utility" on page 10-2](#)
- Connecting to an MBeanServer using the `connect` command described in ["connect" on page 10-4](#)
- Issuing one or more commands to view or update run-time environment attributes or invoke `cca_admin` commands described in ["cca_admin Commands" on page 10-2](#)
- Disconnecting from the server using the command `"disconnect"` on page 10-4 or the command `"exit"` on page 10-4.

Starting the cca_admin Utility

The `cca_admin` tool is delivered as a shell script on Linux or a batch file on Windows.

- In Linux, use the following command:
`OWB_HOME /owb/bin/unix/cca_admin`
- In Windows, use the following command:
`OWB_HOME\owb\bin\win32\cca_admin.bat`.

Before activating the `JRTAdministratorMBean`, you can use only the basic commands: `help`, `connect`, `disconnect`, and `exit`. The full range of commands is available once you connect to an MBean server and start using the `JRTAdministratorMBean`.

cca_admin Commands

This section provides a complete list of the `cca_admin` commands as follows:

- ["cleanUpJobLogFilesForGivenTimeInterval" on page 10-4](#)
- ["cleanUpNLeastRecentJobLogFiles" on page 10-4](#)
- ["connect" on page 10-4](#)
- ["disconnect" on page 10-4](#)
- ["exit" on page 10-4](#)

- ["getCacheJobStatusSize" on page 10-5](#)
- ["getCacheLogNo" on page 10-5](#)
- ["getKmPermissionsFile" on page 10-5](#)
- ["getLogRoot" on page 10-5](#)
- ["getMaxAbortTimeout" on page 10-5](#)
- ["getMaxRuntimeFileSizeMB" on page 10-6](#)
- ["getMaxRuntimeLogSizeMB" on page 10-6](#)
- ["getRuntimeLogEncoding" on page 10-6](#)
- ["getRuntimeLogFileLocation" on page 10-6](#)
- ["getRuntimeLogLevel" on page 10-6](#)
- ["help" on page 10-6](#)
- ["isKmSecurityNeeded" on page 10-7](#)
- ["isLogToAuditTable" on page 10-7](#)
- ["isLogToOdl" on page 10-7](#)
- ["isShowRuntimeLoggerLogHostAddr" on page 10-7](#)
- ["isShowRuntimeLoggerLogHostId" on page 10-7](#)
- ["isShowRuntimeLoggerLogMsgClassPath" on page 10-8](#)
- ["isShowRuntimeLoggerLogMsgLogLevel" on page 10-8](#)
- ["isShowRuntimeLoggerLogUserId" on page 10-8](#)
- ["readSettings" on page 10-8](#)
- ["reset" on page 10-8](#)
- ["setCacheJobStatusSize" on page 10-8](#)
- ["setCacheLogNo" on page 10-9](#)
- ["setKmPermissionsFile" on page 10-9](#)
- ["setKmSecurityNeeded" on page 10-9](#)
- ["setLogRoot" on page 10-9](#)
- ["setLogToAuditTable" on page 10-10](#)
- ["setLogToOdl" on page 10-10](#)
- ["setRuntimeLogEncoding" on page 10-10](#)
- ["setMaxAbortTimeout" on page 10-10](#)
- ["setMaxRuntimeFileSizeMB" on page 10-10](#)
- ["setMaxRuntimeLogSizeMB" on page 10-10](#)
- ["setRuntimeLogLevel" on page 10-11](#)
- ["setShowRuntimeLoggerLogHostAddr" on page 10-11](#)
- ["setShowRuntimeLoggerLogHostId" on page 10-11](#)
- ["setShowRuntimeLoggerLogMsgClassPath" on page 10-11](#)
- ["setShowRuntimeLoggerLogMsgLogLevel" on page 10-11](#)

- "setShowRuntimeLoggerLogUserId" on page 10-12
- "settingsToString" on page 10-12
- "showAllJobLogFile" on page 10-12

cleanUpJobLogFilesForGivenTimeInterval

This command removes all log files that were modified during the specified time range, between `startDate` and `endDate`, and returns the total number of files removed. Both parameters are expressed in the format `DD-MM-YYYY HH:MM:SS`. If the specified `startDate` is `null`, no records are deleted. If `endDate` is `null`, all log files created after the `startDate` are purged.

Example 10-1 Using cleanUpJobLogFilesForGivenTimeInterval Command

```
cleanUpJobLogFilesForGivenTimeInterval 26-12-2008 13:38:56 12-01-2009 22:15:58
```

cleanUpNLeastRecentJobLogFiles

This command removes the specified number, `n`, of least recent log files from the system, and returns the number of job log files that are actually removed.

Example 10-2 Using cleanUpNLeastRecentJobLogFiles command

```
cleanUpNLeastRecentJobLogFiles 10
```

connect

This command connects the user to MBean server.

The command is of the following form, where `host` is the computer on which the CCA is located, `port` is the port number for accessing the CCA, `user` is the user name for accessing the CCA, `password` is that user's password, `connection_type` is one of RMI, OPMN, RMIS, OPMN_RMIS, OPMN_CLUSTER, or OPMN_CLUSTER_RMIS, `application_name` is the name of the application deployed to OC4J, and `instance` is the OC4J process that contains the CCA application:

```
connect host port user password connection_type application_name instance
```

Example 10-3 Using connect Command

```
connect localhost 23791 oc4jadmin welcome rmi jrt host
```

disconnect

This command disconnects the user from the MBean server.

Example 10-4 Using disconnect Command

```
disconnect
```

exit

This command exits the `cca_admin` shell when it is running in the interactive mode. If the connection to the MBean server is not closed, it first disconnects and then exits `cca_admin` shell.

Example 10–5 Using exit Command

```
exit
```

getCacheJobStatusSize

This command returns a value for the numerical (integer) MBean attribute CacheJobStatusSize, which controls the maximum number of the job status logs that can be stored by the in-memory job log cache in CCA. The attribute's value is in the range of 1 and 10000, with the default value of 3000.

Example 10–6 Using getCacheJobStatusSize Command

```
getCacheJobStatusSize 3000
```

getCacheLogNo

This command returns the value of MBean attribute CacheLogNo, which controls the maximum number of job logs that can be stored by the in-memory job log cache in CCA. Obviously the bigger this number is the more memory CCA takes up. The attribute's value is in the range of 1 and 100, with the default value of 10.

Example 10–7 Using getCacheLogNo Command

```
getCacheLogNo
```

getKmPermissionsFile

This command returns the value of MBean attribute KmPermissionsFile, which is the file system path to the ct_permissions.properties file. If this parameter is set to true, the CCA checks ct_permissions.properties to determine what system commands may be executed. This path may be either absolute or relative. The relative value refers to the \$OC4J_HOME/j2ee/home directory. The default value is config/ct_permissions.properties.

Example 10–8 Using getKmPermissionsFile Command

```
getKmPermissionsFile
```

getLogRoot

This command returns the value of MBean attribute LogRoot, which is the directory that stores the CCA job log file. The attribute value may be either absolute or relative. The relative value refers to \$OC4J_HOME/j2ee/home directory; the default value is \$OC4J_HOME/j2ee/home/log.

Example 10–9 Using getLogRoot Command

```
getLogRoot
```

getMaxAbortTimeout

Returns the value of MBean attribute MaxAbortTimeout, the maximum amount of wait time in seconds before the execution is aborted. The attribute's value is in the range of 1 to 60; and the default value is 60.

Example 10–10 Using getMaxAbortTimeout Command

```
getMaxAbortTimeout
```

getMaxRuntimeFileSizeMB

This command returns the value of MBean attribute MaxRuntimeFileSizeMB, the maximum size of any log file, in MB. When a log file exceeds this limit, it is closed and a new log file is opened. The attribute's value is in the range of 1 MB to 15 MB, and the default value is 3 MB.

Example 10–11 Using getMaxRuntimeFileSizeMB Command

```
getMaxRuntimeFileSizeMB
```

getMaxRuntimeLogSizeMB

This command returns the value of MBean attribute MaxRuntimeLogSizeMB, the maximum size of the log file directory, in MB. When the directory exceeds this limit, log files are purged, starting with the oldest. The attribute's value is in the range of 1 MB to 20 MB, and the default value is 10 MB.

Example 10–12 Using getMaxRuntimeLogSizeMB Command

```
getMaxRuntimeLogSizeMB
```

getRuntimeLogEncoding

This command returns the value of MBean attribute RuntimeLogEncoding, which specifies the character encoding of the log file. The default value is UTF-8.

Example 10–13 Using getRuntimeLogEncoding Command

```
getRuntimeLogEncoding
```

getRuntimeLogFileLocation

This command returns the value of MBean attribute RuntimeLogFileLocation, the name of CCA job log file, a read-only attribute. This value is \$OC4J_HOME/j2ee/home/log/appName.log, where appName is the name of the CCA application.

Example 10–14 Using getRuntimeLogFileLocation Command

```
getRuntimeLogFileLocation
```

getRuntimeLogLevel

This application returns the value of MBean attribute RuntimeLogLevel, the log level for run-time logger. The legal values for this attribute are: SEVERE, WARNING, INFO, CONFIG, FINE, FINER, FINEST, ALL (default).

Example 10–15 Using getRuntimeLogLevel Command

```
getRuntimeLogLevel
```

help

This command returns help information for a specified valid command; if a command is not specified, it returns the list of valid commands.

Example 10–16 Using help Command

```
help
help isLogToOdl
```

isKmSecurityNeeded

This command returns the value of MBean attribute KmSecurityNeeded, determining whether security is enforced on system commands that are invoked by CCA script. If the value is `true` (which is the default), CCA checks the `ct_permissions.properties` file for allowable system commands, and only execute the commands defined in `ALLOWED_CMD`. If the value is `false`, CCA executes all system commands.

Example 10–17 Using isKmSecurityNeeded Command

```
isKmSecurityNeeded
```

isLogToAuditTable

This command returns the value of MBean attribute LogToAuditTable, which turns audit table logging on and off. Audit table logging is only effective for execution job logs. The attribute value is either `true` or `false`.

Example 10–18 Using isLogToAuditTable Command

```
isLogToAuditTable
```

isLogToOdl

This command returns the value of MBean attribute LogToOdl, which turns ODL logging on and off. Only effective for execution job logs. Deployment and web service logs are always created in the ODL log format, regardless of the value of this setting. The attribute value is either `true` or `false`.

Example 10–19 Using isLogToOdl Command

```
isLogToOdl
```

isShowRuntimeLoggerLogHostAddr

This command returns the value of MBean attribute ShowRuntimeLoggerLogHostAddr, which determines whether the IP address of the host that is running CCA is included in the run-time log messages. The attribute value is either `true` or `false` (default).

Example 10–20 Using isShowRuntimeLoggerLogHostAddr Command

```
isShowRuntimeLoggerLogHostAddr
```

isShowRuntimeLoggerLogHostId

This command returns the value of MBean attribute ShowRuntimeLoggerLogHostId, which determines whether the ID of the host that is running CCA is included in the run-time log messages. The attribute value is either `true` or `false` (default).

Example 10–21 Using isShowRuntimeLoggerLogHostId Command

```
isShowRuntimeLoggerLogHostId
```

isShowRuntimeLoggerLogMsgClassPath

This command returns the value of MBean attribute

ShowRuntimeLoggerLogMsgClassPath, which determines whether the path of the class where the message is logged is included in the run-time log messages. The attribute value is either true or false (default).

Example 10–22 Using isShowRuntimeLoggerLogMsgClassPath Command

```
isShowRuntimeLoggerLogMsgClassPath
```

isShowRuntimeLoggerLogMsgLogLevel

This command returns the value of MBean attribute

ShowRuntimeLoggerLogMsgLogLevel, which determines whether the log level of the log message is included in the run-time log messages. The attribute value is either true or false (default).

Example 10–23 Using isShowRuntimeLoggerLogMsgLogLevel Command

```
isShowRuntimeLoggerLogMsgLogLevel
```

isShowRuntimeLoggerLogUserId

This command returns the value of MBean attribute

ShowRuntimeLoggerLogUserId, which determines whether the ID of the user is included in the run-time log messages. The attribute value is either true or false (default).

Example 10–24 Using isShowRuntimeLoggerLogUserId Command

```
isShowRuntimeLoggerLogUserId
```

readSettings

This command reads CCA settings from the CCA configuration file.

Example 10–25 Using readSettings Command

```
readSettings
```

reset

This command resets attributes to their default values, if the attributes are defined to have defaults.

Example 10–26 Using reset Command

```
reset
```

setCacheJobStatusSize

This command sets the value for MBean attribute CacheJobStatusSize, which controls the maximum number of the job status that can be stored in the in-memory

job log cache in CCACCA. The attribute's value is an integer in the range of 1 to 10000; the default value is 3000.

Example 10–27 Using setCacheJobStatusSize Command

```
setCacheJobStatusSize 3000
```

setCacheLogNo

This command sets a value for MBean attribute CacheLogNo, which controls the maximum number of job logs that can be stored in the in-memory job log cache in CCA. The attribute's value is an integer in the range of 1 and 100; the default value is 10.

Example 10–28 Using setCacheLogNo Command

```
setCacheLogNo 10
```

setKmPermissionsFile

This command sets the value for MBean attribute KmPermissionsFile, which is the file system path to the ct_permissions.properties file. If the value is set to true, the CCA checks the file to determine what system commands may run. The path can be either absolute or relative. The relative refers to \$OC4J_HOME/j2ee/home directory; the default value is config/ct_permissions.properties.

Example 10–29 Using setKmPermissionsFile Command

```
setKmPermissionsFile $OC4J_HOME/j2ee/home/config/ct_permissions.properties
```

setKmSecurityNeeded

This command sets a value for MBean attribute KmSecurityNeeded, which determines whether security is enforced on system commands invoked from CCA scripts. When the value is true (default), CCA checks the ct_permissions.properties file for system commands that may run, and only execute the ones defined in ALLOWED_CMD. When the value is false, CCA executes all system commands.

Example 10–30 Using setKmSecurityNeeded Command

```
setKmSecurityNeeded true  
setKmSecurityNeeded false
```

setLogRoot

This command sets the value of MBean attribute LogRoot, the directory of the CCA job log file. The attribute's value may be either absolute or relative. The relative value refers to \$OC4J_HOME/j2ee/home directory; the default value is \$OC4J_HOME/j2ee/home/log.

Example 10–31 Using setLogRoot Command

```
setLogRoot $OC4J_HOME/j2ee/home/log
```

setLogToAuditTable

This command sets the value for MBean attribute LogToAuditTable, which turns audit table logging on and off. Audit table logging is only effective for execution job logs. The attribute's value is true or false.

Example 10–32 Using setLogToAuditTable Command

```
setLogToAuditTable true  
setLogToAuditTable false
```

setLogToOdl

This command sets the value for MBean attribute LogToOdl, which turns ODL logging on and off. This is only effective for execution job logs; deployment and web service logs are created in ODL log format regardless of this setting. The attribute's value is true or false.

Example 10–33 Using setLogToOdl Command

```
setLogToOdl true  
setLogToOdl false
```

setMaxAbortTimeout

This command sets the value for MBean attribute MaxAbortTimeout, the maximum wait time (in seconds) before aborting the execution. The attribute's value is an integer in the range of 1 to 60 (default).

Example 10–34 Using setMaxAbortTimeout command

```
setMaxAbortTimeout 60
```

setMaxRuntimeFileSizeMB

This command sets the value of MBean attribute MaxRuntimeFileSizeMB, the maximum size, in MB, a log file in the directory. When a log file exceeds this limit, it is closed and a new log file is opened. The attribute's value is an integer in the range of 1 MB and 15 MB; the default value is 3 MB.

Example 10–35 Using setMaxRuntimeFileSizeMB Command

```
setMaxRuntimeFileSizeMB 5
```

setMaxRuntimeLogSizeMB

This command sets the value of MBean attribute MaxRuntimeLogSizeMB, the maximum size of the log file directory, in MB. When the directory exceeds this limit, log files are purged, starting with the oldest. The attribute's value is in the range of 1 MB to 20 MB, and the default value is 10 MB.

Example 10–36 Using setMaxRuntimeLogSizeMB Command

```
setMaxRuntimeLogSizeMB 10
```

setRuntimeLogEncoding

This command sets the value of MBean attribute RuntimeLogEncoding, which specifies the character set encoding of the log file. The default value is UTF-8.

Example 10–37 Using setRuntimeLogEncoding Command

```
setRuntimeLogEncoding UTF-8
```

setRuntimeLogLevel

This command sets the value for MBean attribute RuntimeLogLevel, the log level for run-time logger. The attribute's value may be SEVERE,WARNING, INFO, CONFIG, FINE, FINER, FINEST, ALL (default).

Example 10–38 Using setRuntimeLogLevel Command

```
setRuntimeLogLevel ALL
```

setShowRuntimeLoggerLogHostAddr

This command sets the value for MBean attribute ShowRuntimeLoggerLogHostAddr, which determines whether the IP address of the CCA host is included in the run-time log messages. The attribute value is either true or false (default).

Example 10–39 Using setShowRuntimeLoggerLogHostAddr Command

```
setShowRuntimeLoggerLogHostAddr true  
setShowRuntimeLoggerLogHostAddr false
```

setShowRuntimeLoggerLogHostId

This command sets the value for MBean attribute ShowRuntimeLoggerLogHostId, which determines whether the ID of the CCA host is included in the run-time log messages. The attribute value is either true or false (default).

Example 10–40 Using setShowRuntimeLoggerLogHostId Command

```
setShowRuntimeLoggerLogHostId true  
setShowRuntimeLoggerLogHostId false
```

setShowRuntimeLoggerLogMsgClassPath

This command sets the value for MBean attribute ShowRuntimeLoggerLogMsgClassPath, which determines whether the path of the class that logged the message is included in the run-time log messages. The attribute value is either true or false (default).

Example 10–41 Using setShowRuntimeLoggerLogMsgClassPath Command

```
setShowRuntimeLoggerLogMsgClassPath true  
setShowRuntimeLoggerLogMsgClassPath false
```

setShowRuntimeLoggerLogMsgLogLevel

This command sets the value for MBean attribute ShowRuntimeLoggerLogMsgLogLevel, which determines whether the log level of the log message is included in the run-time log messages. The attribute value is either true or false (default).

Example 10–42 Using setShowRuntimeLoggerLogMsgLogLevel Command

```
setShowRuntimeLoggerLogMsgLogLevel true
```

```
setShowRuntimeLoggerLogMsgLogLevel false
```

setShowRuntimeLoggerLogUserId

This command sets the value for MBean attribute ShowRuntimeLoggerLogUserId, which determines whether the ID of the user is included in the run-time log messages. The attribute value is either true or false (default).

Example 10–43 Using setShowRuntimeLoggerLogUserId Command

```
setShowRuntimeLoggerLogUserId true  
setShowRuntimeLoggerLogUserId false
```

settingsToString

This command converts CCA setting information to String, and shows all CCA settings.

Example 10–44 Using settingsToString Command

```
settingsToString
```

showAllJobLogFiles

This command shows all job log files names.

Example 10–45 Using showAllJobLogFiles Command

```
showAllJobLogFiles
```

Using Snapshots to Manage Metadata

This section describes how to use snapshots to manage metadata. Snapshots in Oracle Warehouse Builder enable you to capture point-in-time versions of the metadata definitions of the design objects you create in the Project Navigator. You can capture all design objects in a project, or you can selectively choose objects to include in a snapshot.

This section includes the following topics:

- [About Snapshots and Metadata Loader](#)
- [Using Snapshots](#)

About Snapshots and Metadata Loader

Because snapshots and the Metadata Loader have overlapping capabilities, it may not be readily apparent when to use one and not the other. In general, snapshots address the needs of a development team for tracking changes to an evolving design. The Metadata Loader facilitates the more traditional administrative tasks of backing up, restoring, and migrating repositories.

A key factor to understand is that snapshots and the Metadata Loader differ in the scope of objects they handle. Snapshots are limited to handling only the design objects users create in the Project Navigator. The Metadata Loader, however, handles everything in the repository. This includes all design objects and information related to location and security.

Refer to [Table 11–1](#) for a list of important differences between the two tools.

Table 11–1 Comparison of Snapshots and Metadata Loader

Feature	Snapshots	Metadata Loader
Scope	Captures metadata related to objects in the Project Navigator only. Captures metadata for mappings, process flows, and other design objects. Does not capture information about locations, security, and user defined objects.	Captures all metadata related to a repository including all design objects and information about locations, security, and user defined objects.

Table 11–1 (Cont.) Comparison of Snapshots and Metadata Loader

Feature	Snapshots	Metadata Loader
Output	You can create a full snapshot from which to restore objects. Or you can create a signature snapshot which requires less space and enables you to track changes but not restore objects. Both types of snapshots are stored in Oracle Database.	When exporting, the Metadata Loader creates a ZIP file with the extension .mdl. You can use the output to restore a repository or populate a new repository on a different Oracle Database.
Integration with Version Control Tools	You can save the output as a file and then export the file into a version control tool.	You can export MDL files into a version control tool. When using the GUI, first extract the MDL file before putting the objects into the version control system. When using the OMB Plus scripting language, set ZIPFILEFORMAT=N as described in "About Metadata Loader Control Files" on page 12-19.
Compatibility	Snapshots are not compatible between releases of Warehouse Builder.	MDL files are upwards compatible with newer releases of Warehouse Builder. When you import an MDL file from a previous release, the utility automatically upgrades the metadata to the new release.
Object Locking	To restore a snapshot, the Change Manager must obtain a lock on all primary objects. Locks held within that session are first freed by prompting the user to save or revert any pending changes, then automatically closing all editor windows. If a lock is held by another session, the restore fails and Change Manager rolls back any changes.	To import, the Metadata Loader must obtain a lock on all the primary objects. If a necessary lock is not available, the import fails and the Metadata Loader rolls back any changes.
Compare	You can compare snapshots by launching a command in the user interface.	You can compare MDL files by using a third-party diff tool.

Using Snapshots

When using snapshots, you can backup and restore your metadata, maintain a history of metadata changes, and compare different versions of the metadata.

A snapshot captures all the metadata information about the selected objects and their relationships at a particular instance of time. While an object can only have one current definition in a workspace, it can have multiple snapshots that describe its changing states throughout its history. Warehouse Builder supports the following types of snapshots:

- Full snapshots provide backup and restore functionality.
- Signature snapshots provide historical records for comparison.

Snapshots are stored in Oracle Database. Contrast this with Metadata Loader exports, which are stored as separate disk files. Note that you can export snapshots to disk files.

Snapshots are useful for both warehouse managers and designers. Managers can use full snapshots to perform large-scale actions, such as deploying a warehouse or restoring a warehouse to an earlier state. Designers can create full snapshots of a

particular component under development so that, if necessary, they can restore that component to its previous state. Designers can also use signature snapshots to track the changes made to a particular component.

When used with other features, such as MDL metadata imports and impact analyses, snapshots help you manage the metadata. For example, you can use snapshots to determine the impact an MDL metadata import has on the current metadata workspace. With the knowledge gained by the comparison, you may choose to import the metadata at a lower level of detail to avoid overwriting the definition of related metadata objects.

Snapshots are also used to support the recycle bin, providing the information necessary to restore a deleted metadata object.

The rest of this section describes the metadata change management feature using the graphical user interface. For information about creating and managing snapshots using scripts, see *Oracle Warehouse Builder API and Scripting Reference*.

Creating Snapshots

When you take a snapshot, you capture the metadata of all or specific objects in your workspace at a given point in time. You can use a snapshot to detect and report changes in your metadata.

You can create snapshots of any first class objects, except for automatic default objects, and User or Role objects. Use Create Snapshot wizard to create snapshots.

Note that a snapshot of a collection is a snapshot of both the shortcuts in the collection and the actual objects. The same applies to a snapshot of a user folder, because it includes references (which are similar to shortcuts) to both contained and actual objects.

To create a snapshot:

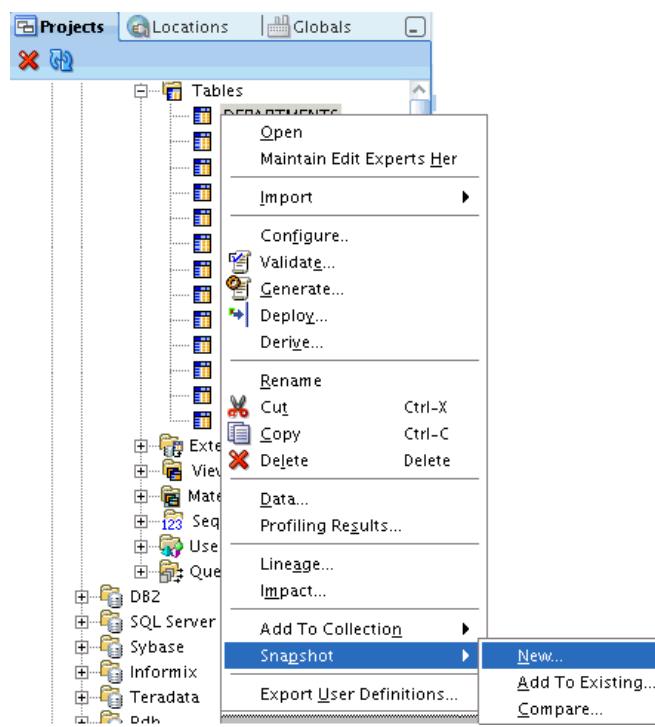
1. In the Project Navigator, select all the objects that you want to include in the snapshot.

You do not need to select parent or child objects, because the wizard does that for you.

For example, if you select a collection that contains two cubes, then the snapshot includes both cubes and any gives you the option to include any dependent objects such as dimensions and source tables.

Note that you must select actual objects or user folders, not the standard folders that contain them.

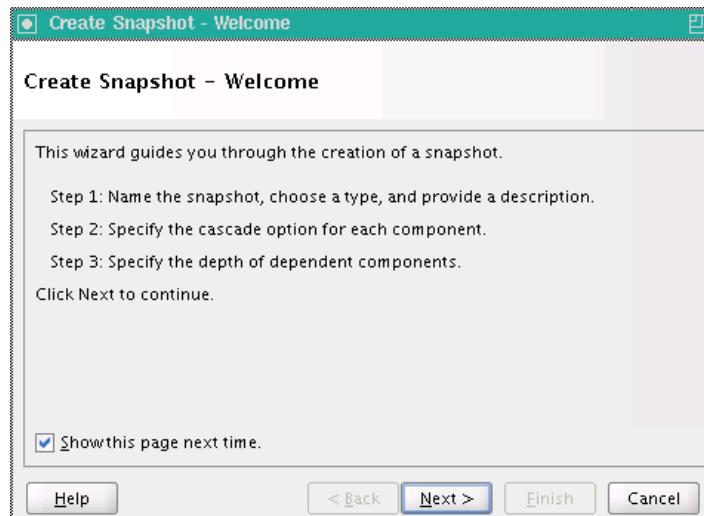
2. Right-click the selected object, then select **Snapshot**, then select **New**.



Alternatively, from the File menu, select **Snapshot**, then select **New**.

3. On the Welcome screen, click **Next**.

Note that you can get supporting information at any time by clicking **Help**.



4. On the Name page, specify the **Name** and the **Type** (Full or Signature) of the snapshot.

Click **Next**.

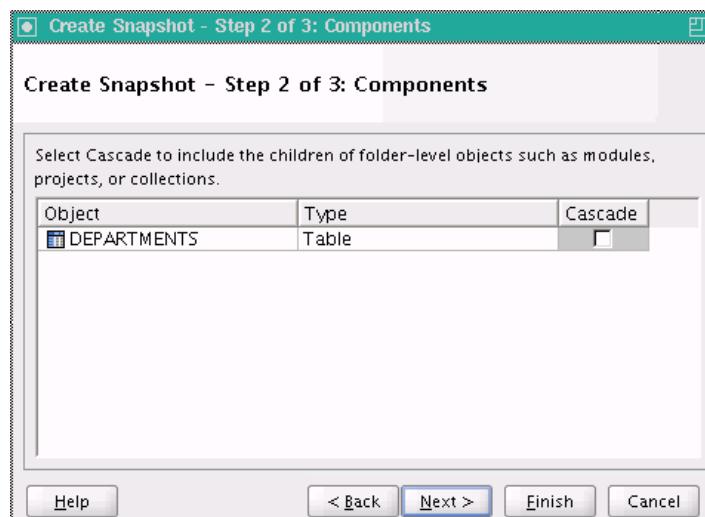
[Optional] Provide a **Description**.



5. On the Components screen, choose whether to select the **Cascade** option.

Click **Next**.

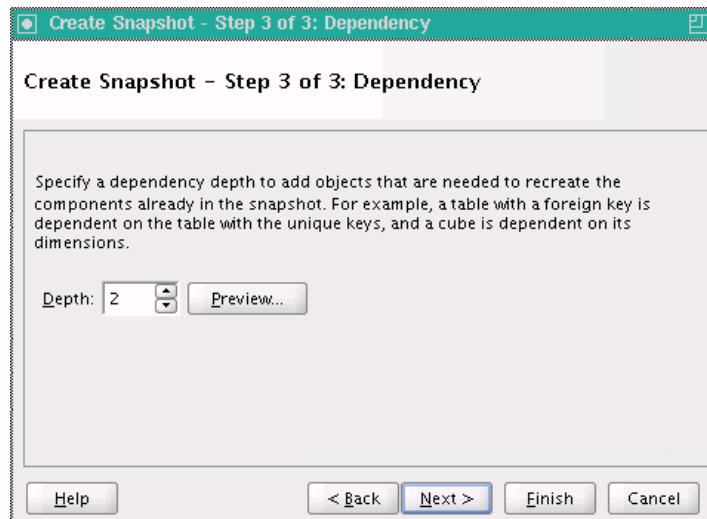
This screen displays the components that are captured in the snapshot. If a component is a node-level object, such as a module, then select the Cascade option to include the subcomponents.



6. On the Dependency screen, specify the **Depth** of dependency to include dependent objects in the snapshot.

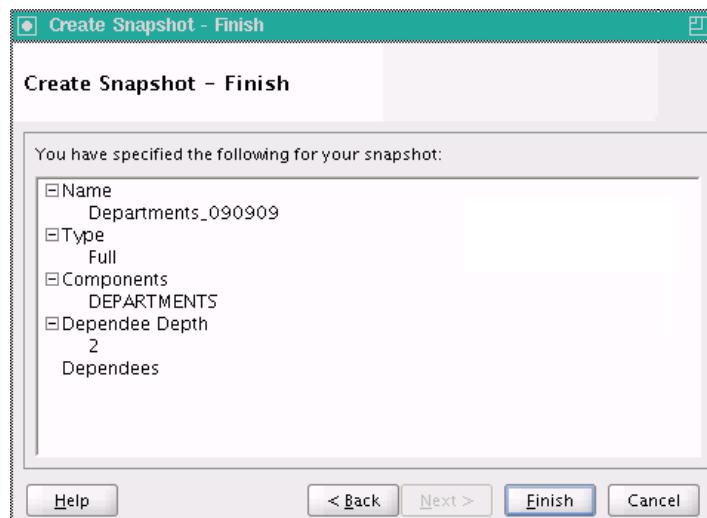
Click **Next**.

Because it impacts performance, avoid specifying unnecessarily large values for depth. For example, supplying a depth value greater than 0 for a Project snapshot searches Globals for dependencies.

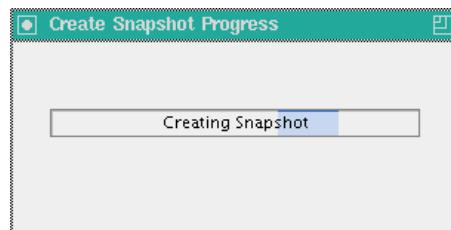


7. On the Finish screen, review the characteristics of the snapshot.

Click **Finish** to create the snapshot.

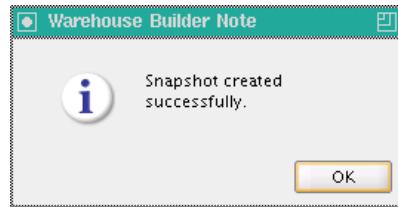


8. The system starts creating the snapshot; notice the Create Snapshot Progress pop-up.



When the snapshot is complete, a Warehouse Builder Note pop-up screen appears, stating "Snapshot created successfully."

Click **OK**.



Adding Components to a Snapshot

After you create a snapshot, you can add more components. Keep in mind, however, that when you add components, you are changing the historical record provided by the snapshot, so its creation date changes.

To update a snapshot, use the Add to Snapshot wizard.

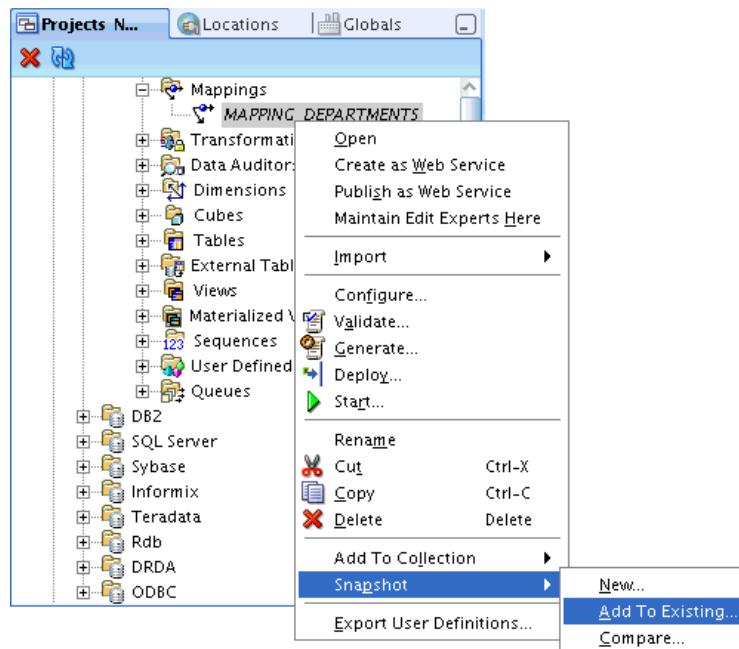
To add new components to the snapshot:

1. In the Design Center Project Navigator, select all the components you want to add to the snapshot.

For example, you can select tables, mappings, and dimensions from an Oracle module.

2. Right-click and select **Snapshot**, then select **Add to Existing**.

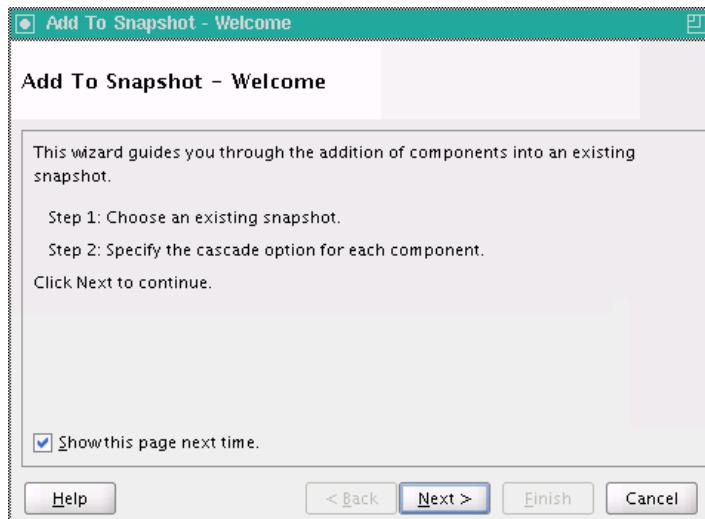
Alternatively, from the File menu, select **Snapshot**, then select **Add to Existing**.



The Add to Snapshot wizard appears.

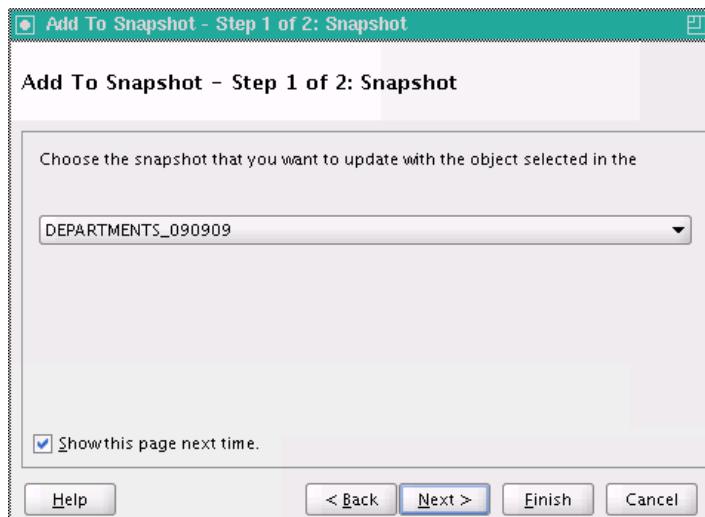
3. On the Welcome screen, click **Next**.

Note that you can get supporting information at any time by clicking **Help**.



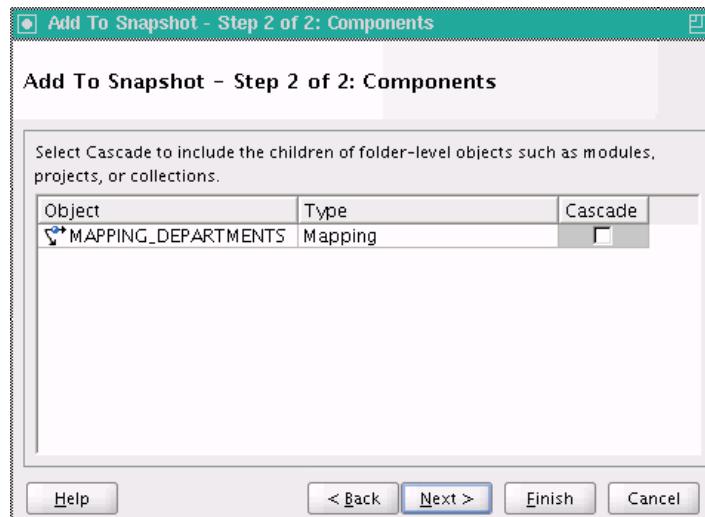
4. On the Snapshot screen, from the list of snapshots, select the snapshot where to add the components.

Click **Next**.

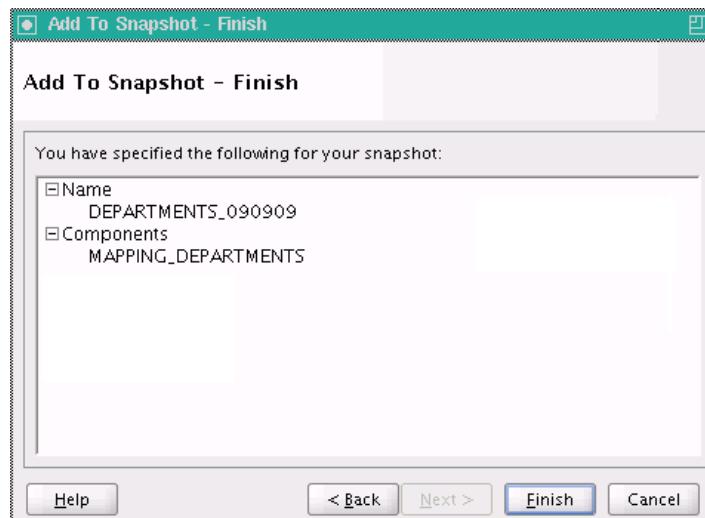


5. On the Components screen, note all the components that are added to the snapshot. If a component is a folder level object, such as a module, then select the Cascade option to include its subcomponents.

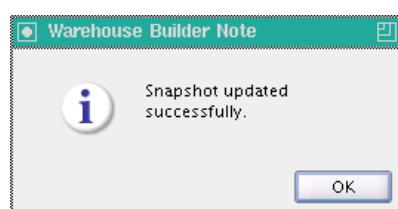
Click **Next**.



6. On the Finish screen, review the characteristics of the snapshot.
Click **Finish** to update the snapshot.



7. The system starts updating the snapshot; notice the Update Snapshot Progress pop-up.
When the snapshot is complete, a Warehouse Builder Note pop-up screen appears, stating "Snapshot updated successfully."
Click **OK**.



Managing Snapshots

You can manage your snapshots from the Metadata Change Management window in the Design Center. To open this window, select **Change Manager** from the Tools menu.



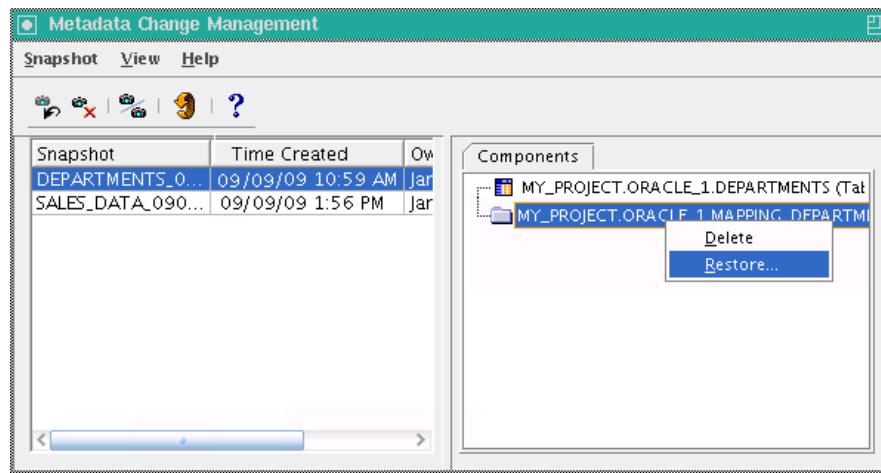
The Metadata Change Management window contains a menu bar and a toolbar. You can start most tasks in several different ways, either by using the menus, clicking the tools, or right-clicking a snapshot or a component.

Metadata Change Management window supports the following activities:

- [Managing Snapshot Access Privileges](#)
- [Comparing Snapshots](#)
- [Restoring Workspace Objects From Snapshots](#)
- [Exporting and Importing Snapshots](#)
- [Deleting Snapshots](#)

Figure 11–1 shows the Metadata Change Management window.

Figure 11–1 Metadata Change Management Window

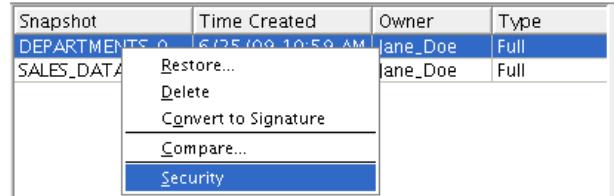


Managing Snapshot Access Privileges

You can control access to snapshots just like any other object. By default, everyone has full access rights to the snapshots.

To change snapshot access privileges:

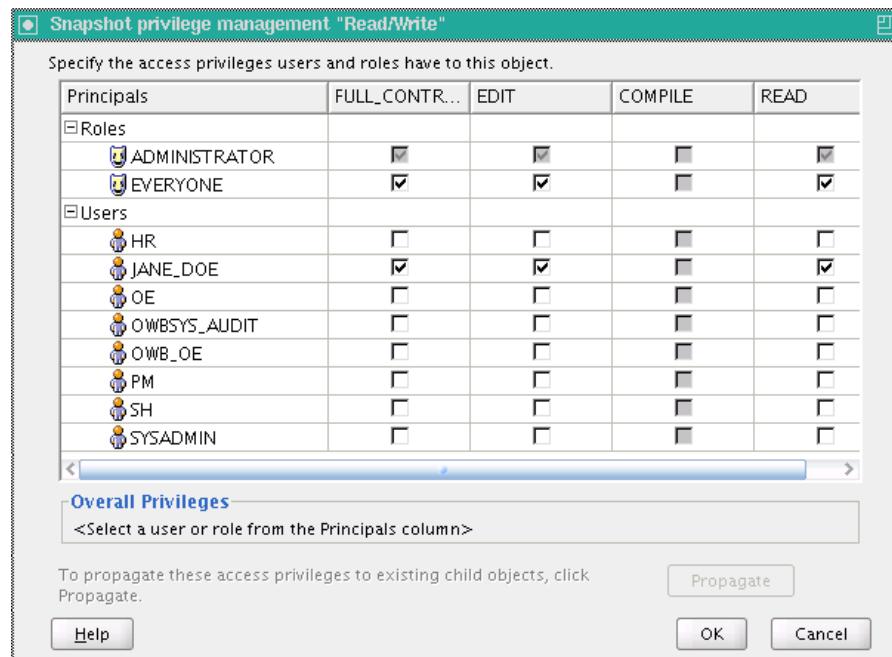
1. In the left section of the Metadata Change Management window, right-click the name of the snapshot.
2. From the menu, select **Security**.



The Snapshot Privilege Management dialog box is displayed.

3. On the Snapshot Privilege Management screen, for each user or role, select the privileges to grant and clear the privileges to deny.

Click **Help** for additional information.



Comparing Snapshots

You can compare two snapshots, or a snapshot with a current workspace object. The results of a comparison list all objects as one of:

- Identical objects appearing in both snapshots
- An object appearing in only one snapshot
- Similar objects appearing in both snapshots, with different property values

If you take a snapshot of an object that has a child object, and that child object subsequently changes, a comparison shows that the parent object has changed. For example, if you create a Cascade snapshot of a project, all the modules in that project are its children. If any of the modules change, a comparison of the snapshot to the workspace shows that the project has changed.

The Metadata Change Manager uses Universal Object Identifiers (UOIDs) as the basis for all comparisons. When you delete an object and re-create it with the same metadata, the object has a different UUID although it may be identical to the deleted object in all other aspects. When you compare the re-created object to its original version, the results show that all the metadata has changed. Likewise, objects do not match if they have been deleted and re-created during an import or Intelligence Object derivation.

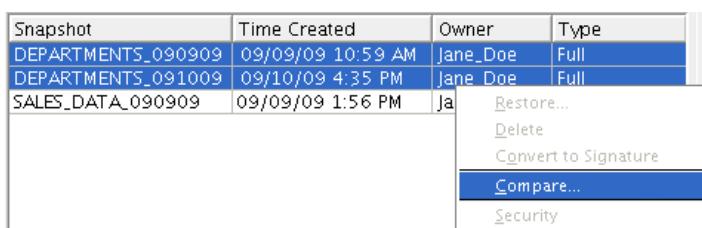
Note that a temporary signature snapshot of the current workspace object is taken automatically; this increases the comparison time. If you are planning to make multiple comparisons to the same current workspace object, Oracle recommends that you take a signature snapshot and use it directly in snapshot-to-snapshot comparisons.

Two compare two snapshots:

1. On the Metadata Change Management screen, select two snapshots for comparison.

Right-click the selected snapshots, and select **Compare**.

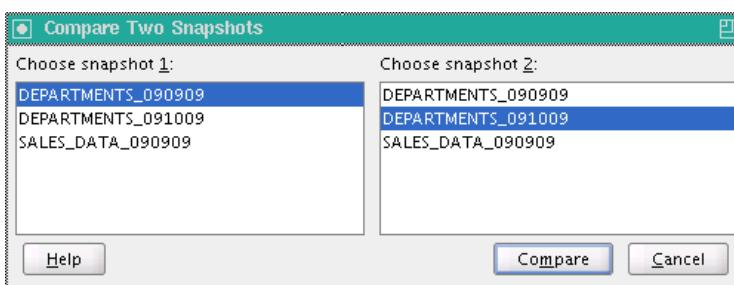
Alternatively, from the Snapshot menu, select **Compare**.



The Compare Two Snapshots screen appears.

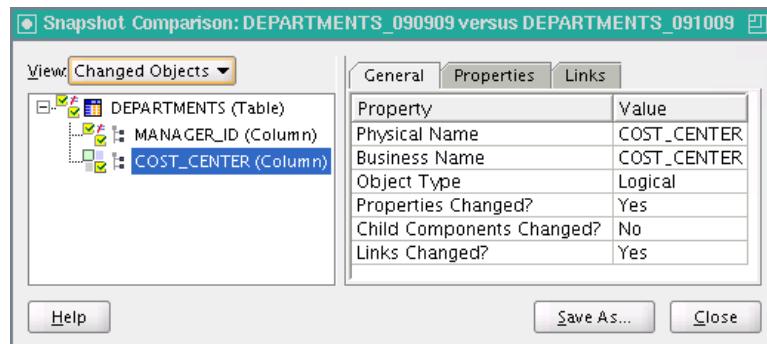
2. On the Compare Two Snapshots screen, ensure that the correct snapshots are selected, and click **Compare**.

[Optional] Click **Help** for additional information.



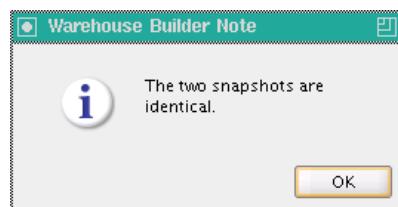
The Snapshot Comparison screen appears.

3. The Snapshot Comparison screen displays the differences between the two snapshots. By selecting one of the changed objects you can see the details of the changes.



If there are no differences between snapshots, a Warehouse Builder Note pop-up screen appears, stating "The two snapshots are identical."

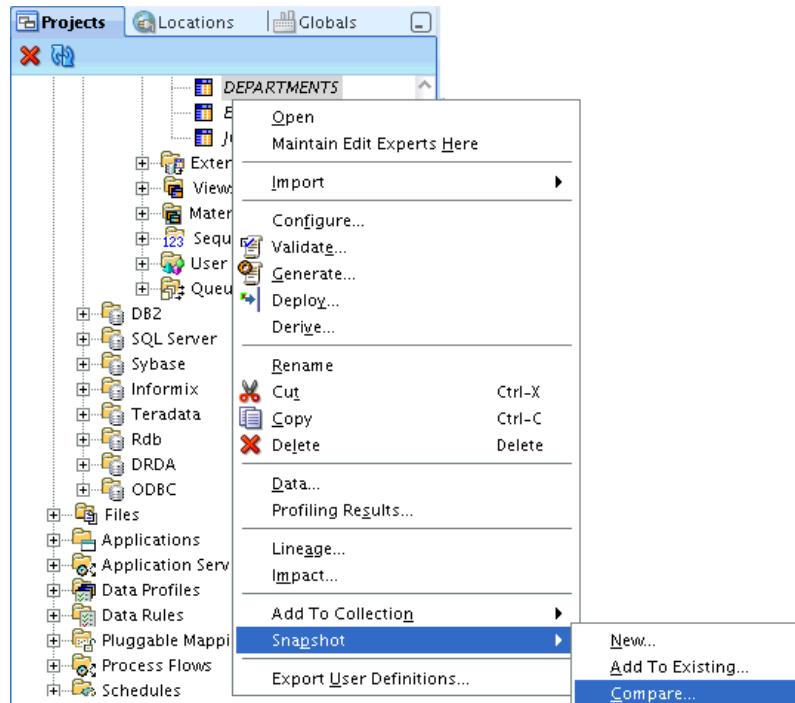
Click **OK**.



4. [Optional] Click **Save As** to save the results of the comparison to an *.xml file.
5. Click **Close**.

To compare the current version of an object with a snapshot version:

1. From the Project Navigator, select the object for comparison.
2. Right-click the selected object, select **Snapshot**, and then select **Compare**.

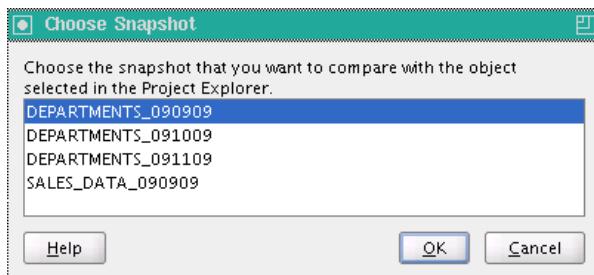


Alternatively, from the File menu, select **Snapshot**, and then select **Compare**.

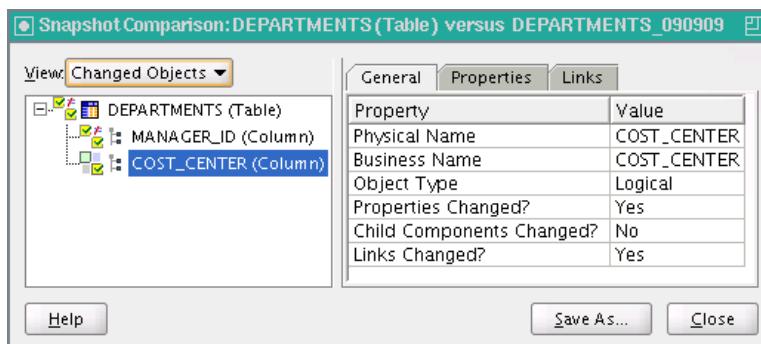
The Choose Snapshot screen appears. Note that it only lists the snapshots that contain the object of interest.

3. On the Choose Snapshot screen, select the snapshot for comparison.

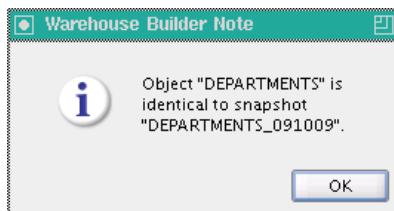
Click **OK**.



4. The Snapshot Comparison window displays the differences between the two objects. If there are none, then a message informs you that the objects are the same.
5. The Snapshot Comparison screen displays the differences between the object and the snapshot. By selecting one of the changed objects you can see the details of the changes.



If there are no differences between snapshots, a Warehouse Builder Note pop-up screen appears, stating "The *object* is identical to the snapshot *snapshot*".



Click **OK**.

6. [Optional] Click **Save As** to save the results of the comparison to an *.xml file.
7. Click **Close**.

Converting a Full Snapshot to a Signature Snapshot

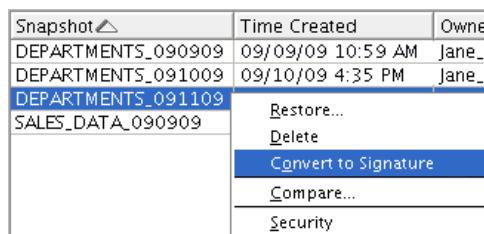
You can convert a full snapshot to a signature snapshot if it is no longer necessary for backup. This conversion preserves the snapshot history while saving a significant amount of space in the workspace.

Note that a signature snapshot cannot be used to restore metadata objects.

To convert a full snapshot:

1. Open the Metadata Change Management window, as described in "[Managing Snapshots](#)" on page 11-10.
2. Select the snapshot that you want to convert.
3. Right-click the selected snapshot, and select **Convert to Signature**.

Alternatively, from the Snapshot menu, select **Convert to Signature**.



4. On the Warehouse Builder Warning dialog box, click **Yes**.



5. Notice that the Type of the snapshot has changed to Signature.

Snapshot	Time Created	Owner	Type
DEPARTMENTS_090909	09/09/09 10:59 AM	Jane_Doe	Full
DEPARTMENTS_091009	09/10/09 4:35 PM	Jane_Doe	Full
DEPARTMENTS_091109	09/11/09 5:08 PM	Jane_Doe	Signature
SALES_DATA_090909	09/09/09 1:56 PM	Jane_Doe	Full

Restoring Workspace Objects From Snapshots

You can replace the current definition of an object in the workspace with the snapshot image of that object.

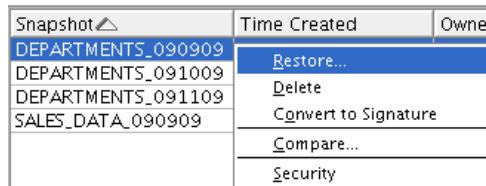
Note that:

- You can only use full snapshots; you cannot restore objects from signature snapshots.
- You can restore all components or only selected components of the snapshot.
- When you restore a collection from a snapshot, you restore both the collection and the actual objects.
- When you restore a container, all its child objects are also restored.
- To restore only the collection or selected objects within a container, use the Components tab.

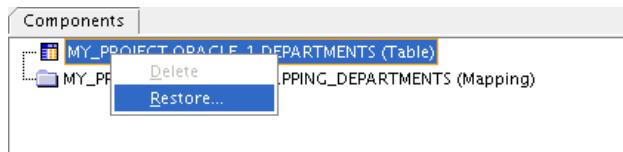
To restore objects from a snapshot:

1. Save any work that you have done in the current session.

2. Open the Metadata Change Management window, as described in "Managing Snapshots" on page 11-10.
3. Select the snapshot that contains the version of the objects you want to restore.
4. [Optional] To restore all objects, right-click the snapshot, and select **Restore**.

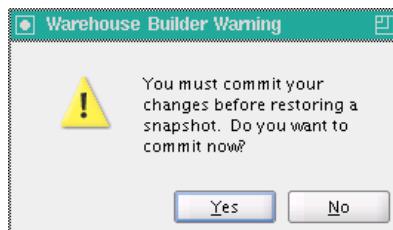


5. [Optional] to restore only some components, select these components in the Components tab, right-click the component, and select **Restore**.



6. [Optional] If the most recent changes to the object have not been committed, a Warehouse Builder Warning note appears, stating "You must commit your changes before restoring a snapshot. Do you want to commit now?"

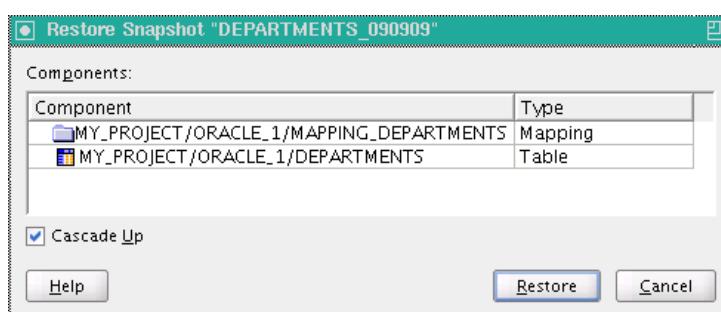
Click **Yes**.



7. On the Restore Snapshot screen, review the objects in the list and verify that the correct objects are restored. If you

Select **Cascade Up**. This ensures that any components whose parents no longer exist in the workspace are restored with new parent objects.

Click **Restore**.



8. A Warehouse Builder Warning note appears, stating "All currently open editors will be closed after a successful restore. Do you want to continue?"

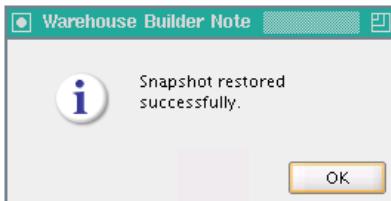
Click **Yes**.



- The system starts restoring the snapshot; notice the Restore Snapshot Progress pop-up.

When the snapshot is restored, a Warehouse Builder Note pop-up screen appears, stating "Snapshot restored successfully."

Click **OK**.

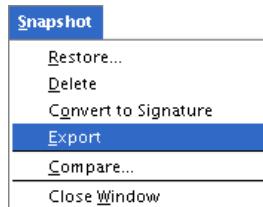


Exporting and Importing Snapshots

You can export a full snapshot to an MDL file and use this file to re-create metadata objects either in the same database or in a different one. Note that you cannot export signature snapshots or individual components of a snapshot.

To export a snapshot:

- Open the Metadata Change Management window.
- Select the full snapshot you want to export.
- From the Snapshot menu, select **Export**.



- [Optional] If the most recent changes to the object have not been committed, a Warehouse Builder Warning note appears, stating "You must either save or revert your changes before exporting."

Click **Save**.

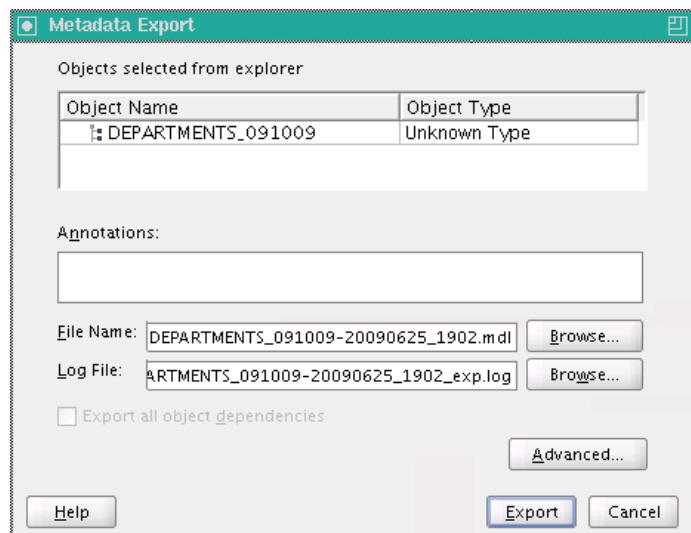


5. On the Metadata Export screen, do the following:

- Click **Browse** to select the location and name of the export file (File Name).
- Click **Browse** to select the location of the Log File.
- Click **Export**.

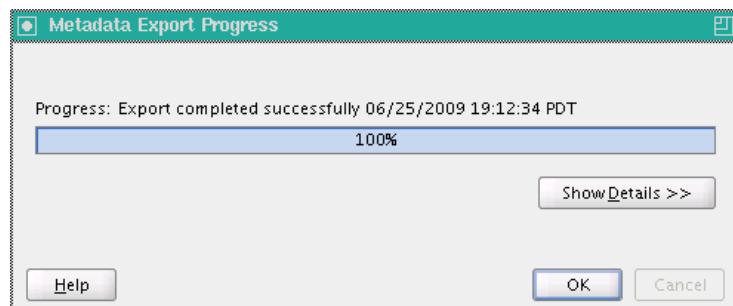
Note that the export file has an .mdl extension, and the log file has a .log extension.

[Optional] Click **Help** for additional information.



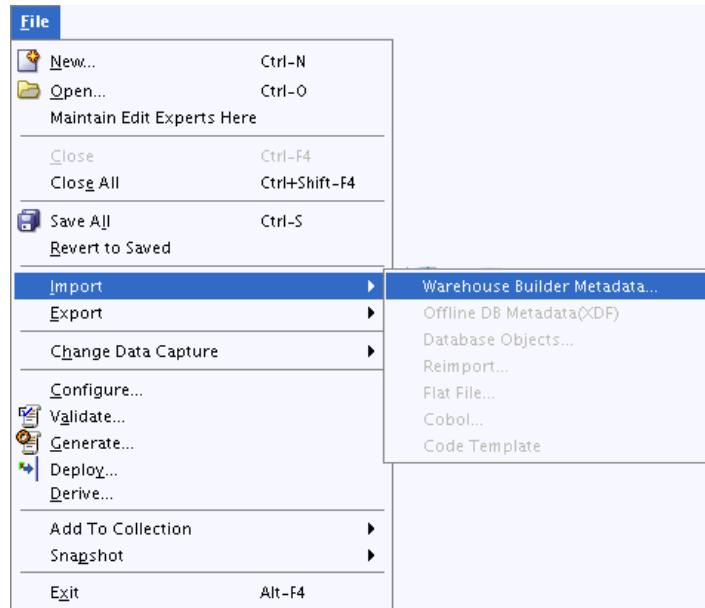
6. Note that the Metadata Export Progress screen appears.

When the progress bar shows 100%, click **OK**.

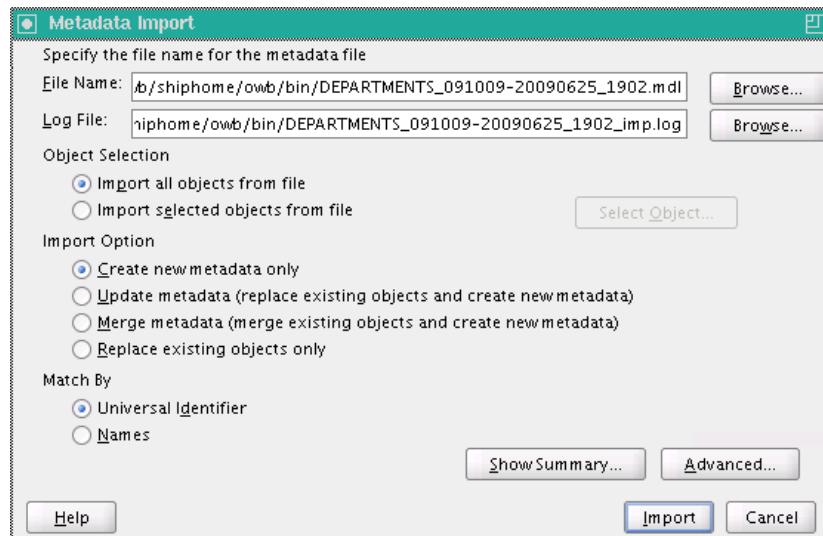


To import a snapshot:

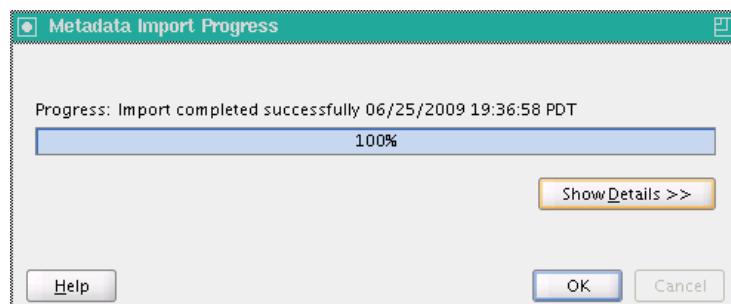
1. From the File menu, select **Import**, then select **Warehouse Builder Metadata**.



2. On the Metadata Import screen, specify the **File Name** of the file that is imported. Click **Import**.



3. Note that the Metadata Import Progress screen appears. When the progress bar shows 100%, click **OK**.



Deleting Snapshots

You can delete snapshots or components within a snapshot from the Metadata Change Management window.

Note that you cannot delete individual components of a signature snapshot.

To delete a snapshot:

1. On the Metadata Change Management screen, select the snapshot to delete.
2. Right-click the snapshot and select **Delete**.

Alternatively, from the Snapshot menu, select **Delete**.

Snapshot	Time Created	Owner	Type
DEPARTMENTS_090909	09/09/09 10:59 AM	Jane_Doe	Full
DEPARTMENTS_091009	09/10/09 4:35 PM	Jane_Doe	Full
DEPARTMENTS_091109		oe	Signature
SALES_DATA_090909		oe	Full

This figure shows the screen for this step.

- *****
3. The Warehouse Builder Warning note appears, stating "You must either save or revert your changes before exporting."

Click **Yes**.



To delete components from a snapshot:

1. In the Metadata Change Management, select the snapshot from where to delete components.
2. On the Components tab, select the component to delete, right-click the component, then select **Delete**.

Alternatively, from the Snapshot menu, select **Delete**.

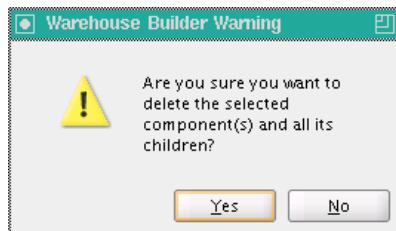
Components
MY_PROJECT.ORACLE_1.DEPARTMENTS (Table)
MY_PROJECT.ORACLE_1.MAPPING_DEPARTMENTS (Mapping)

This figure shows the screen for this step.

Warehouse Builder displays a delete confirmation dialog box.

3. The Warehouse Builder Warning note appears, stating "Are you sure you want to delete the selected component(s) and all its children?"

Click Yes.



Snapshot Performance

Performance of snapshot operations depends on how accurate the SQL optimizer statistics are for a small subset of tables and indexes. Because most of these in a newly created repository contain little data, it is necessary to seed and lock the statistics initially. This ensures reasonable performance for moderately sized snapshots in a lightly populated repository.

As the number and size of snapshots increases, unless accurate statistics are refreshed on a periodic basis, the performance of snapshots degrades. Although a refresh may be performed in several different ways, OMBOPTIMIZE (in OMB*Plus) or Optimize Repository (in Tools menu) should be run at least once. Otherwise, certain tables remain locked and continue to use old, seeded statistics.

12

Using the Metadata Loader

This section discusses how to use the Metadata Loader for change and version management. The Metadata Loader copies and moves all types of metadata objects in a repository. With this utility, you can move metadata between Oracle Warehouse Builder repositories that reside on platforms with different operating systems. You can use the Design Center to run the Metadata Loader utilities. The Design Center provides a graphical interface that guides you through the process of exporting and importing metadata.

This section contains the following topics:

- [Exporting Metadata from the Design Center](#)
- [Importing Metadata Using the Design Center](#)
- [Upgrading Metadata from Earlier Releases](#)
- [Metadata Loader Utilities](#)
- [Multiple Session Concurrency and MDL](#)
- [Metadata Loader Log File](#)
- [About Metadata Loader Results](#)
- [Managing Versions and History with the Metadata Loader](#)

Exporting Metadata from the Design Center

You can use Design Center to export objects from a workspace into an MDL file. This includes objects that are part of the Project Navigator, Locations Navigator, and Globals Navigator. The information related to the exported objects, such as table columns and their constraints, data loading configuration parameters, and named attribute sets, are also exported.

Before you attempt to export metadata, ensure you have READ privileges on any object that you want to export. You also need to have READ privileges on folder objects. If you do not have READ privileges on a folder object, such as projects or modules, the folder object and all objects that it contains are not exported. During an export, the Metadata Export Utility skips objects for which you do not have READ privileges. It logs information about the list of objects that have not been exported due to lack of security privileges. See "[Metadata Loader Log File](#)" on page 12-18 for details about viewing the list of objects that were not exported.

By default, READ privileges are provided on all the workspace objects to all registered users. See "[Export Advanced Options Dialog Box](#)" on page 12-10 if you want to export security information such as users, roles, role assignments, and object privileges.

You have two options for exporting metadata. You could either export the entire design or you could export selected objects in the workspace. If you want to export selected objects, then those objects must be within the same navigator. For example, you cannot export a table from the Project Navigator and a public transformation from the Globals Navigator at the same time; you must export them in two separate steps.

Before starting the export operation, take care of the following:

- Ensure the destination computer has sufficient disk storage. If there is insufficient disk space, the export fails. The destination computer must be able to contain the entire metadata file, because the export utility cannot save portions of the metadata file.
- Ensure that you are the only user accessing the objects to be exported. To ensure that you are exporting the most up-to-date metadata, it is advisable to ask all other users to log out of the workspace. See "[Multiple Session Concurrency and MDL](#)" on page 12-18.

Note: When you export a code template map that contains a user-defined code template, the code template is not automatically included in the export operation. You must first explicitly import any user-defined code templates that you want the code template mapping to contain.

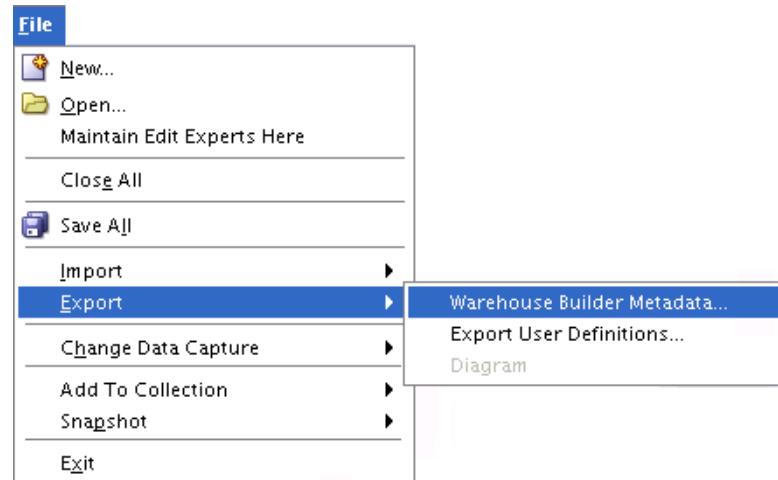
To export metadata from a workspace using the Design Center:

1. From the Design Center, select the object or objects you want to export.

You can select multiple objects by holding down the **Control** key and selecting the objects.

Note that you may export individual objects such as tables, or groups of objects. When you export projects nodes, or modules, you also export the objects they contain. When you export collections, you also export the objects they reference.

2. From the **File** menu, select **Export** and then **Warehouse Builder Metadata**.



3. [Optional] If you made changes to the repository metadata prior to running the export utility, a warning dialog box is displayed.

Click **Save** to save changes or **Revert** to revert to the previously saved version.



4. In the Metadata Export Dialog Box:

- Click **Browse** to specify the **File Name** (name and location) of the export file.
- Click **Browse** to specify the **Log File** (name and location).
- [Optional] Under **Annotations**, enter comments about the export file.
- [Optional] Check **Export all object dependencies** if you are also exporting all the dependencies of the exported objects.
- [Optional] Click **Advanced** to export additional metadata. See "[Export Advanced Options Dialog Box](#)" on page 12-10.

See "[Metadata Export Dialog Box](#)" on page 12-9.

5. Click **Export**.

6. When the export completes successfully (see "[Metadata Progress Dialog Box](#)" on page 12-16), in the Metadata Export Dialog box, click **OK**.

Importing Metadata Using the Design Center

You can use the Design Center to import metadata. The Metadata Import Utility also automatically upgrades metadata that was created using an earlier version of Warehouse Builder to the current version. See "[Upgrading Metadata from Earlier Releases](#)" on page 12-7 for more information about upgrading metadata.

To prepare for importing metadata, you must take care of the following:

- **Required Access Privileges:** To import metadata, the user performing the import must have the following privileges:
 - **EDIT** privilege on existing objects that are being replaced by the import.
 - **CREATE** privilege on existing folder objects under which new objects are created by the import.

By default, the **FULL_CONTROL** privilege is assigned on all workspace objects to registered users. The Metadata Import Utility skips objects for which the user importing metadata does not have the required privileges. The list of objects that have not been imported due to security privileges are logged in the Metadata Loader Log File. See "[Metadata Loader Log File](#)" on page 12-18

You can import security information such as users and roles. When importing user metadata, if a corresponding database user does not exist for a user, the import fails and an error message is written to the Metadata Loader Log File. See "[Import Advanced Options Dialog Box](#)" on page 12-14 for information about importing security information such as users and roles, and "[Metadata Loader Log File](#)" on page 12-18.

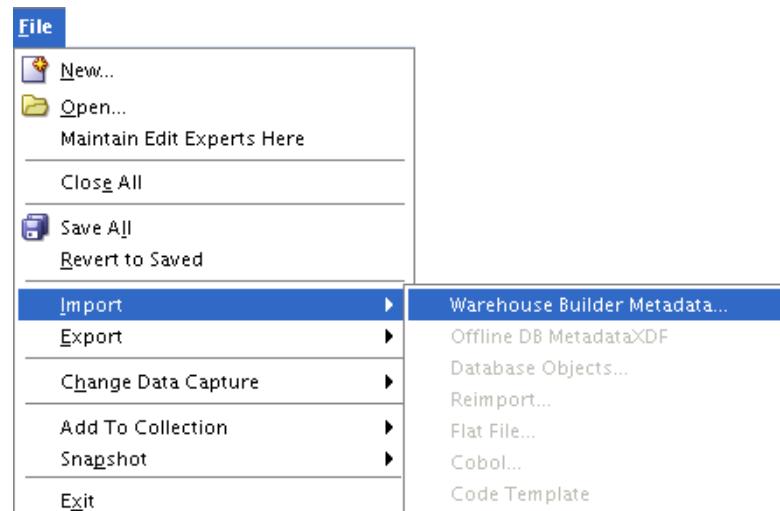
Because the Metadata Import Utility alters the repository, the metadata objects must be locked prior to importing. See "Multiple Session Concurrency and MDL" on page 12-18.

- **Backup of Current Workspace:** Consider making a backup of your existing workspace, either in the form of an export or a metadata snapshot, before attempting a large or complex import.
- **Multiple Language Support Base Language Compatibility:** The base language is the default language used in the repository and is set using the Repository Assistant during installation. You cannot alter this setting after installing the repository. See "Languages" on page 12-11 for more information about setting the base language in a repository.

Importing Objects

To import objects from an export file using the Design Center:

1. From the Design Center, select **File**, select **Import**, and then select **Warehouse Builder Metadata**.



2. [Optional] If you had made changes to the repository metadata prior to running the import utility, a warning dialog box is displayed.

Click **Save** to save changes or **Revert** to revert to the previously saved version.



3. In the Metadata Import dialog box:

- Click **Browse** to specify the **File Name** (name and location) of the import file.
- Click **Browse** to specify the **Log File** (name and location).

- Under **Object Selection**, choose either **Import all objects from file** or **Import selected objects from file**.
- Under **Import Option**, select one of **Create new metadata only**, **Update metadata replace existing objects and create new metadata**, **Merge metadata merge existing objects and create new metadata**, or **Replace existing objects only**.
- Under **Match by**, select one of **Universal Identifier** or **Names**.
- [Optional] Click **Show Summary** to view a summary of the contents of the export file. See "[File Summary Dialog Box](#)" on page 12-15.
- [Optional] Click **Advanced** to import metadata for additional languages, security information, or user-defined properties. See "[Import Advanced Options Dialog Box](#)" on page 12-14.

See "[Metadata Import Dialog Box](#)" on page 12-11.

4. Click **Import**.
5. The Metadata Import Progress dialog box is displayed; it indicates the progress of the import. Note that you can cancel an in-progress Metadata Import by clicking **Cancel**. See "[Metadata Progress Dialog Box](#)" on page 12-16.

If the MDL file that you selected for import was created using an earlier version of Warehouse Builder, clicking **Show Summary**, **Advanced**, or **Import** displays the Metadata Upgrade dialog box. This dialog box enables you to automatically upgrade the selected MDL file to the current version of Warehouse Builder. For more information about this dialog, see "[Metadata Upgrade Dialog Box](#)" on page 12-16.

Combining Import Modes and Matching Criteria

Each search method used as matching criteria may be combined with an import mode. Each combination offers different results in the import process. The mode that you select determines how the metadata import utility searches for metadata objects in the workspace, prior to importing.

For example, if the search is by the name of a repository object in the export file, the Metadata Import Utility searches the workspace for the object's name. If an object with the corresponding name is not found, the resulting actions are based on the import mode you select.

Table 12-1 describes what happens in the available import modes for repository objects that either match or do not match the object names.

Table 12-1 Import Mode Results

Import Mode	Result Using Matching Names	Result Not Using Matching Names
Create Mode	A message is written to the log file that the object exists, and the object is skipped.	A new object is created.
Replace Mode	The object is replaced.	A warning message is written to the log file that the object cannot be replaced because it does not exist in the workspace. The object is skipped.
Update Mode	The object is replaced.	A new object is created.
Merge Mode	The object is merged.	A new object is created.

The MDL reads and processes the imported metadata and writes status and diagnostic information in the log file.

Import Different Base Languages

When you import metadata in multiple languages, the language settings in the target repository may be different from the language settings in the export file. For example, the target repository may have the base language as English and additional language as French and German. But the export file may have the base language as French and additional language as English and German. This section describes how the MDL handles these conditions.

Different Base Languages in the Import File and Target Repository

When you import metadata, MDL compares the ISO identification of the base language in the import file with the ISO identification of the base language of the target repository. The ISO identification consists of the language ID followed by the locale, in the format *language_locale*. For example, *en_US* is American English and *fr_FR* is French.

If the base ISO identification languages are different, MDL displays a warning dialog box informing you that the base languages are different and warns you that Oracle recommends that you import metadata with the same character set and base language. You have the option to continue with the import. Click **Yes** to continue with the import. Click **No** to cancel the import.

Note that under certain circumstances, continuing to import metadata when the base languages are different may lead to corruption of the metadata being imported. Oracle recommends that you move metadata between repositories with the same character set and base languages.

If the base ISO identification languages are the same, but the locales are different, the Metadata Import Utility displays a warning dialog box to determine whether you want to continue with the import. For example, the export file contains English and the base language of the repository is American English. Click **Yes** to import metadata, or click **No** to cancel the import.

Importing Supported Languages

During the import, MDL checks if the additional languages in the import file exist in the target repository. If the import file contains additional languages that do not exist in the target repository, and you specify that these additional languages are to be imported, the Metadata Import utility writes a warning message in the MDL log file stating that the additional languages are not installed in the repository.

Import Validation Rules

When you import a set of definitions from previously exported metadata, the import utility updates existing definitions within a project. However, certain metadata definitions require attention to ensure that they are updated correctly. The following are examples of some of the errors you may see:

- **Mapping Definitions.** The Metadata Import Utility binds imported mapping operators to their physical objects if the associated objects exist in the workspace. However, if the associated physical objects do not exist in the workspace, the imported mapping operators are not bound. The Metadata Import Utility writes a warning message in the log file, which states that the mapping operators are not

bound. You must then synchronize the new mapping operators with the physical objects they represent.

- **Foreign Key Definitions.** It is possible that a source MDL file contains foreign key references to unique or primary keys that are not in the target workspace. If the referenced unique or primary keys for any foreign key appearing in the MDL file do not exist in the target workspace, the MDL Import Utility writes a warning message in the log file, which states that the workspace does not contain a referenced key for the foreign key.

Upgrading Metadata from Earlier Releases

While importing metadata, the Metadata Import Utility automatically upgrades metadata created using earlier releases to Oracle Warehouse Builder 11g Release 2 (11.2). You do not have to manually upgrade the metadata from a previous release of Warehouse Builder.

When you import an MDL file, the version used to create the file is automatically detected. If the MDL file was created using an earlier release of Oracle Warehouse Builder, the Metadata Upgrade dialog box is displayed. This dialog box enables you to upgrade the MDL file to the current version. See "["Metadata Upgrade Dialog Box"](#)" on page 12-16.

If you import an .mdl file that contains metadata for gateway modules (such as DB2 or Informix) from an older version of Warehouse Builder, the file may not import the metadata into the corresponding source module folders of a project. These imported files are stored under the Others node in the Project Navigator. You must manually copy the metadata for the gateway modules into the correct source module folders.

The production versions of Warehouse Builder from which metadata is automatically upgraded to Oracle Warehouse Builder 11g Release 2 (11.2) are as follows:

- Oracle Warehouse Builder 9.2
- Oracle Warehouse Builder 10g Release 1
- Oracle Warehouse Builder 10g Release 2
- Oracle Warehouse Builder 11g Release 1

If you must upgrade an MDL file from an older version of OWB, you must first upgrade it to one of OWB 9.2, 10.1, 10.2, or 11.1, and then upgrade this intermediate MDL file to OWB 11.2.

Important: Before importing the MDL file, you must stop the Control Center by running the script *OWB_HOME/owb/rtp/sql/stop_service.sql*.

Changes to Workspace Objects When Upgrading from OWB 9.2

When you upgrade from OWB 9.2, the upgrade utility makes the following changes to objects in the workspace:

- **Dimension and Cube Mapping Operators:** The mapping operators for dimensions and cubes are converted to table operators. These table operators use the physical tables created by the MDL Upgrade Utility for dimensions and cubes.
- **Dimensions:** An associated dimension table is created with the same name as the dimension. The table contains the columns, constraints, and attribute sets defined in the Dimension Editor Table Properties of the dimension in the previous release.

- **Mapping Display Sets for Dimension Hierarchies:** Any mapping sets originally created based on the named attribute set for a dimension hierarchy are removed. This is because display sets for dimension hierarchies are no longer automatically created and maintained.
- **Dimension Attributes:** For each level attribute upgraded, a dimension attribute with the same name is created, if it does not exist in the dimension.
- **Cubes:** An associated cube table is created with the same name as the cube. The cube table contains columns, constraints, and attribute sets defined in the Cube Editor Table Properties of the cube in the previous release.
- **Intelligence Objects and Reports:** In the previous release, intelligence objects and reports were available only using OMB Plus scripting. These objects are not upgraded.
- **Locations and Control Center Connections:** Locations and Control Center connections are moved out of the projects that own them so that they can be shared across the entire workspace. Thus the statistics in the import log file displays an additional project for these objects.
- **Control Centers and Locations:** After an upgrade, there is no association between the locations and the control centers that they reference. You must review the control center details using the Edit Control Center dialog box and select the locations associated with this control center.
- **Advanced Queues:** An associated queue table is created based on the property AQ queue table name. The queue table created by the MDL File Upgrade Utility contains a column whose data type is the object type for that advanced queue.
- **Advanced Queue Operator in a Mapping:** Mapping Advanced Queue operators are changed to contain only one attribute called PAYLOAD. For Mapping Advanced Queue operators that are used as a source, a new Expand operator is added after the Mapping Advanced Queue operator. For Mapping Advanced Queue operators that are used as a target, a new Construct operator is added before the Mapping Advanced Queue operator.
- **Mapping Operator Names:** The MDL Upgrade Utility ensures that the physical names and business names of all mapping operators are unique.

Changes to Workspace Objects When Upgrading from OWB 10.1, 10.2 or 11.1

When you upgrade from OWB 10.1, 10.2, or 11.1 to OWB 11.2, the upgrade utility makes the following changes to objects in the workspace:

- **Mapping Table Function Operators:** The mapping operators for table functions are upgraded to have only one input group and one output group. Old input groups that were configured as a REF cursor type are upgraded to a SYS_REF_CURSOR type, and a new construct operator is added to the map; its input group values are the same as in the REF cursor type input group, and the output group attribute is set to SYS_CURSOR type.
- **Mapping Key Lookup Operators:** The mapping operators for key lookups are upgraded to the new lookup operator.
- **Stream Capture Process Objects:** Stream capture process objects are not upgraded.
- **Business Presentation Modules:** Business presentation modules and their contents are not upgraded.
- **BI Beans Locations:** BI beans locations are not upgraded.

Checking for Warnings and Error Messages

After upgrading the metadata, check the log file for warnings and errors.

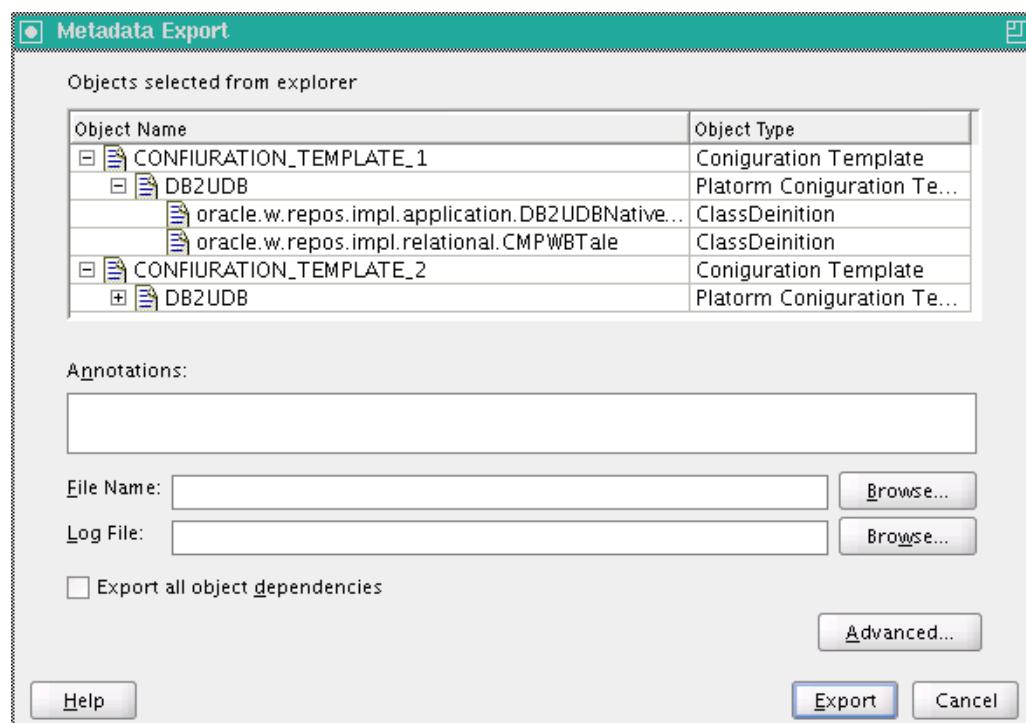
- If you receive warnings during the upgrade, the upgrade utility completes and logs the warnings. If you receive errors, the upgrade utility terminates and logs the errors.
- If warnings and errors are shown after an upgrade, search for the words Warning and Error in the log file to determine the problem.
- If an unexpected error occurs and the upgrade terminates, the log file contains the details. Check your log file or contact Oracle Support.

Design Center Metadata Management Dialog Boxes

The Design Center uses several dialogs to manage metadata export, import, and upgrade. These are:

Metadata Export Dialog Box

The Metadata Export dialog box displays the names and the types of the exported objects.



The Metadata Export dialog box also contains the following items:

- **Annotations:** Use this field to enter any comments about the file that contains the exported objects.
- **File Name:** Displays a default path and file name for the export file. You can retain this default or specify a directory and file name. Type the name of the export file to create or click **Browse** to locate a directory or file. The file name extension commonly used is .mdl.

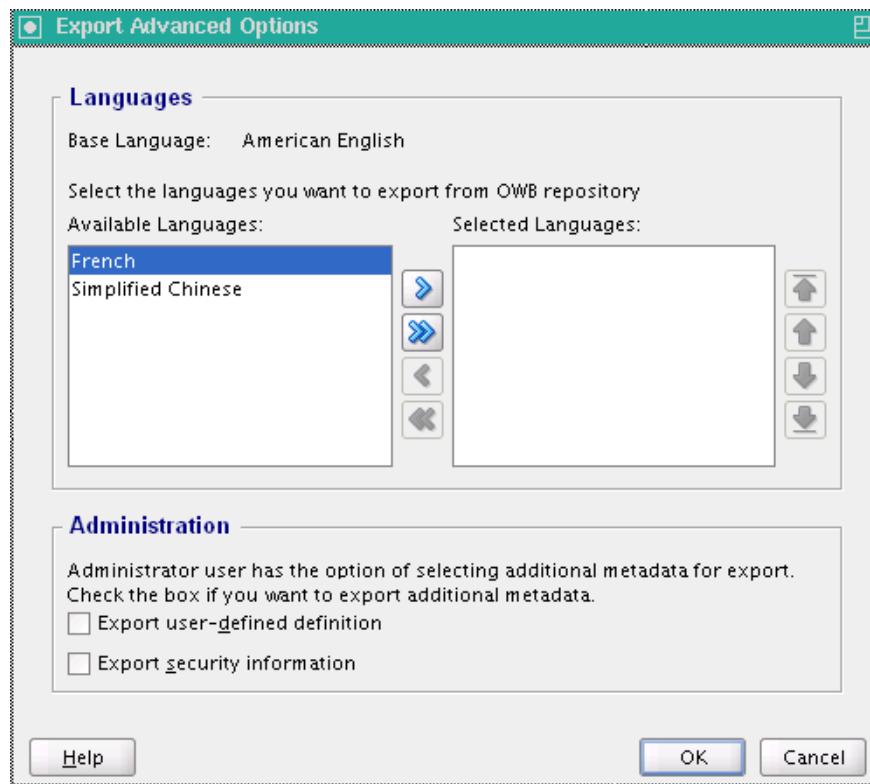
- **Log File:** Use this field to specify the file name and path for the log file that stores diagnostic and statistical information about the export. See "[Metadata Loader Log File](#)" on page 12-18
 - **Export all object dependencies:** Select this option to export all the dependencies of the objects being exported. For example, when you export a table, the location to which the table is deployed is also exported.
- Note that public objects such as locations, public transformations, public experts, public icon sets, or public data rules belong to a project called PUBLIC_PROJECT. You can export the PUBLIC_PROJECT and its objects if the selected exported objects have a dependency on the public objects and if you select the **Export all object dependencies** option.
- **Advanced:** Use the Advanced button to export additional metadata such as user-defined properties, security information, and additional languages. See "[Export Advanced Options Dialog Box](#)" on page 12-10.

Click **Export** to export the metadata for the selected objects. The Metadata Export Progress dialog box is displayed. See "[Metadata Export Dialog Box](#)" on page 12-9 for more information about the contents of this dialog box.

Export Advanced Options Dialog Box

Use the Export Advanced Options dialog box to export any of the following:

- Additional language metadata
- User-defined definitions
- Security information



This dialog box contains two sections: [Languages](#) and [Administration](#).

Languages

The **Base Language** field displays the base language of the repository. Warehouse Builder exports and imports data in the base language.

You can specify additional languages to export and import objects that contain translations for their business names and descriptions. The **Available Languages** list displays the list of languages that are installed in the repository. To export or import additional languages, select the language and click the arrows to move the language from the **Available Languages** list to the **Selected Languages** list. You may choose multiple languages at the same time by holding down the Control or Shift key while making your selection.

Note that the Available Languages list contains language entries only if you installed additional languages in the repository. For example, the repository has American English as base language, and additional languages French and Simplified Chinese. While exporting metadata from the repository, you may select French as the additional language. The Metadata Export Utility then exports the base language of the object, American English, and the additional language French for objects that contain a French translation. Note that additional languages are exported for an object only if they contain translations for the business names and descriptions.

Administration

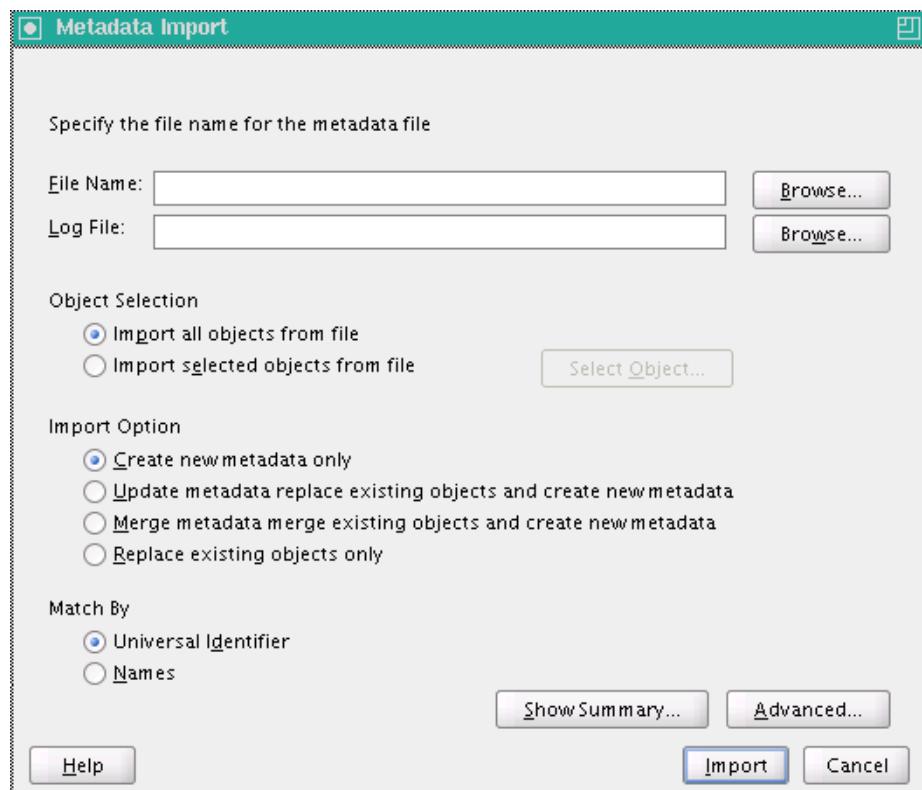
You may export additional metadata if you have administrator privileges. The options you may choose to export additional metadata are as follows:

- **Export user-defined definition:** Select this option to export the definitions of user-defined objects and user-defined properties.
- **Export security information:** Select this option to include security information such as object privileges or role assignments made to users. See [Chapter 13, "Managing Security"](#) for more information about security.

After you specify the options on the Export Advanced Options dialog box, click **OK** to close this dialog box and return to the Metadata Export dialog box.

Metadata Import Dialog Box

Use the Metadata Import dialog box to specify the information required to import metadata in an export file.



Specify the following information on this dialog box:

- **File Name:** Type the name of the MDL file or click **Browse** to locate the MDL file you want to import.
- **Log File:** Type the name of the log file, along with the path, that will store diagnostic and statistical information about the import. You can also click **Browse** to locate the log file. See "["Metadata Loader Log File"](#)" on page 12-18 for more information about log files.
- **Object Selection:** The Metadata Import Utility enables you to select the objects that you want to import from the MDL file. The Object Selection section contains the following options:
 - **Import all objects from file:** Select this option to import all objects contained in the export file.
 - **Import selected objects from file:** Select this option to import only some of the objects contained in the MDL file. Click **Select Objects** to select the objects that you want to import. The Import Object Selection dialog box is displayed.

The Import Object Selection dialog box is displayed contains two sections: **Available** and **Selected**. The Available section contains primary objects such as projects, modules, tables, views, and connections, which are specified in the MDL file. Expand the nodes in this section to view the objects they contain. When you select a node, all the objects that it contains are included in the import. For example, if you select a module node, all the objects contained in the module are imported. Use the shuttle buttons to move the selected objects from the Available section to the Selected section.

The MDL file that is imported may also contain administrative objects. To import these administrative objects, you must have administrative privileges. If these privileges are absent, the following occurs:

- The Available section of the Import Object Selection Page dialog box does not display any administrative objects that the MDL file may contain.
- The Import Utility displays an alert that the user does not have the required administrative privileges to perform the import, provided the MDL file contains only administrative objects.
- **Import Option:** Use the Import Option section to select the import mode. You can select one of the following options for the import mode:
 - **Create new metadata only:** This option, known as **create mode**, adds new objects to a workspace.
 - **Update metadata (replace existing objects and create new metadata):** This option, known as **update mode**, adds new objects to a workspace and also replaces existing objects with those in the MDL file that is imported.
 - **Merge metadata (merge existing objects and create new metadata):** When you select this option, known as **merge mode**, the MDL adds new objects and overwrites existing objects in the workspace only if they differ from those in the MDL file. The merge mode does not delete existing objects. Note that you cannot import metadata using the merge mode for mappings, pluggable mappings, and data auditors.
 - **Replace existing objects only:** This option, known as **replace mode**, replaces existing objects in your workspace but does not add new objects. In this mode, the Metadata Import Utility overwrites any existing metadata.

When you import metadata using the update or the replace modes, the import completely replaces the child objects of existing objects, so that the final object is exactly the same as the source object. Any existing children of a repository object that are not replaced or added are deleted. This occurs regardless of whether a child object occurs in a mapping or is a foreign, primary, or unique key column in a table or view.

For example, in the MDL export file, the CUST table contains three columns with the physical names: last_name, first_name, and middle_init. In the workspace, the same table exists, and contains four columns with the physical names: last_name, first_name, status, and license_ID. During a replace operation, the columns last_name and first_name are replaced, column middle_init is added, and column status and license_ID are deleted. The final result is that the CUST table in the workspace contains the same metadata from the CUST table in the export file.

Note that using the replace and update modes may cause loss of data constraints, metadata physical property settings, data loading properties, and mapping attribute connections. If you choose to use replace or update modes, ensure that you have the option of restoring the workspace from a backup.

- **Match By:** When you use the metadata import utility, it first searches the workspace for metadata objects that exist in the workspace and compares them to those in the import file. To compare metadata in the import file with the existing workspace metadata, it uses the matching criteria. How the comparison is made is determined by the import mode and by the chosen search method.

The Match By section provides the following options for matching criteria. By default, the import utility searches by UOIDs. Note that MDL import does not support merging existing mappings.

- **Universal Identifier** option searches the workspace using the Universal Object Identifiers (UUIDs) of the objects that are imported. The Metadata Import

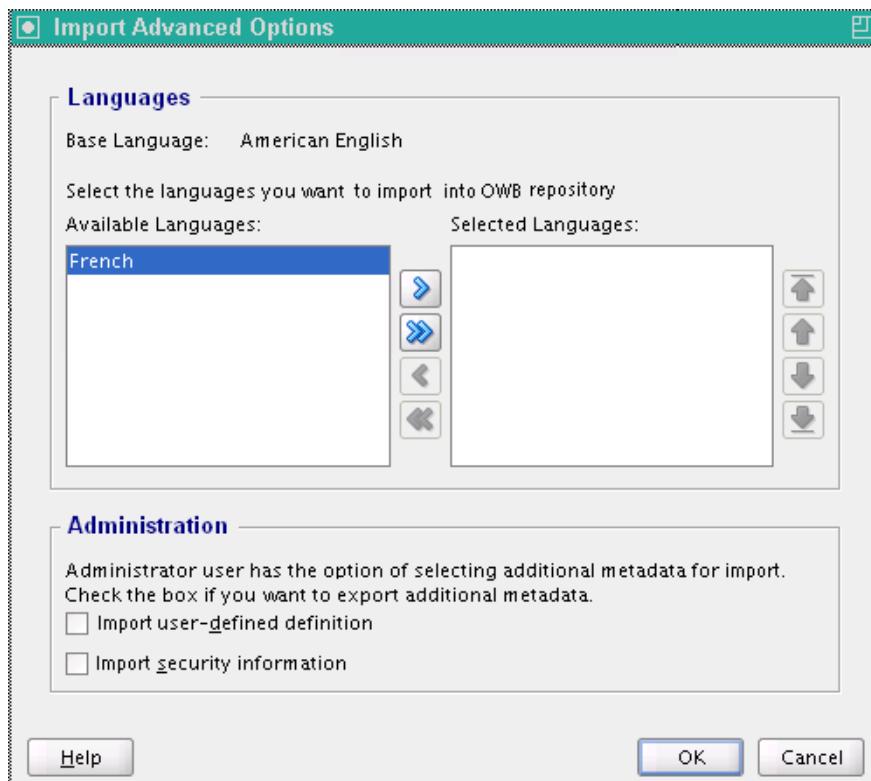
Utility uses the UOIDs to determine whether an object must be created, replaced, or merged during the import operation. Use this method if you want to maintain UOIDs across different workspaces even when object names in the target workspace have changed.

- **Names** option searches the workspace using the names of the objects that are imported. Physical names are exported to the export file; the physical name determines whether an object must be created, replaced, or merged during an import operation. Use this method if you want to create new UOIDs for those objects when object names in the target schema change.

Import Advanced Options Dialog Box

Use the Import Advanced Options dialog box to import any of the following:

- Additional language metadata
- User-defined definitions
- Security information



This dialog box contains two sections: [Languages](#) and [Administration](#).

Languages

The Base Language displays the base language of the repository. By default, data is imported in the base language.

You can specify additional languages to import. The Metadata Import Utility imports the translations of the object for business name and description. The Available Languages list displays the list of languages that are specified in the MDL file. For example, the MDL file contains the additional languages French and Simplified Chinese, but your repository contains only French as the additional language. Then

the Available Languages list displays only French. Select the language you want to import and click the arrow to move the language to the Selected Languages list. See ["Import Different Base Languages"](#) on page 12-6.

Administration

This option is available only if you have administrator privileges and the metadata exists in the MDL file used for import. The options you may choose to import additional metadata are as follows:

- **User-defined definitions:** To import the definitions for the user-defined objects and the user-defined properties, select the **Import User-defined definitions** option.
- **Security Grants:** Select **Import security information** to import security information such as object privileges and role assignments made to users.

If the MDL file contains any of these objects, then you can import this additional metadata.

When you import an MDL file into a new workspace, if you want to inherit the security information from the old workspace, you must import the security information before you import other objects. To do this you need to be connected to the workspace as a user with administrator privileges.

After you make your selections on the Import Advanced Options dialog box, click **OK** to save your selections and return to the Metadata Import dialog box.

Name Conflicts

Name conflicts may occur in one of the following cases:

- A different object with the same name exists in the target workspace.
- A different object with the same business name exists in the target workspace.
- A different object with the same UUID exists in the workspace.

When a name conflict occurs, the MDL reports an error and terminates the import.

File Summary Dialog Box

The File Summary dialog box contains a brief summary of the contents of the export file. The information on this page is divided into the following sections:

- **File** section contains the name of the data file, the creation timestamp, the name of the export user, the workspace connection information, the version of the Design Center used for the export, and annotations.
- **Administration** section contains information about the users and roles. It also lists the following details:
 - Base language of the export file
 - Additional languages in the export file
 - Whether security information were included in the export file
 - Whether user-defined definitions were included in the export file
- **Statistics** section contains details about the types of objects contained in the export file and the number of objects of each type.

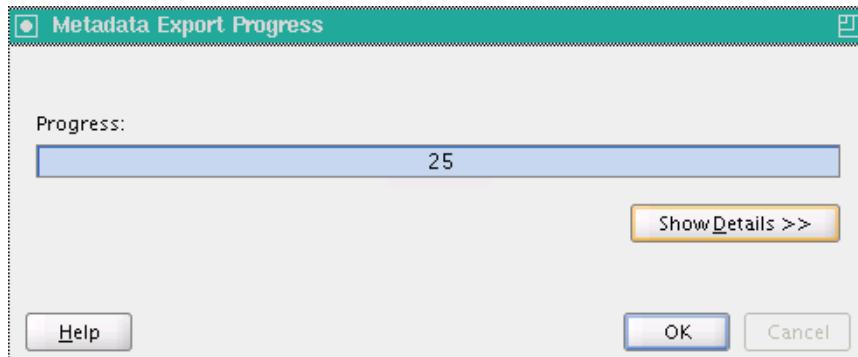
Metadata Upgrade Dialog Box

This dialog box is displayed automatically when it is determined that the MDL file being imported was created using a previous version. Use the **File Name** field to specify the name of the file that stores the upgraded MDL file. You can also click **Browse** to locate a directory or MDL file.

Click **Upgrade** to upgrade the MDL file to the current version. After the Upgrade completes, the Metadata Upgrade dialog box is closed. Click **Cancel** if you do not want to upgrade the MDL file.

Metadata Progress Dialog Box

The Metadata Progress dialog box is titled either Metadata Export Progress, Metadata Import Progress, or Metadata Upgrade Progress. It displays a progress bar that indicates the relative completeness (as a percentage) of the metadata export, import, or upgrade activity. If the operation is successful, a message indicating this displays above the progress bar; it may also display that an error occurred.



To view detailed information about the metadata export or import, click **Show Details**. The message log is displayed. The message log contains the following information:

- Start time of the export or import
- Names and types of objects exported or imported
- Warning or error messages
- End time of the export or import
- Location of the export or import log file
- Total export or import time in hh:mi:ss or milliseconds

You can hide the message log by clicking **Hide Details**.

To view details about the exported or imported objects, click **Show Statistics**. The Metadata Results dialog is displayed. See "[About Metadata Loader Results](#)" on page 12-19.

Note that you can cancel an in-progress Metadata Export or Import by clicking **Cancel**. Once the export or import completes, click **OK** in the Metadata Progress dialog.

Metadata Loader Utilities

The Metadata Loader consists of the following two utilities:

- Metadata Export Utility

Use the Metadata Export Utility to export metadata from a workspace.

- **Metadata Import Utility**

Use the Metadata Import Utility to import metadata into a workspace.

MDL uses its own format, and the Metadata Import Utility only reads files of MDL format (files created by the Metadata Export utility). The Metadata Loader file is a formatted ZIP file.

Metadata Export Utility

The Metadata Export Utility extracts metadata objects from a workspace and writes the information into a ZIP format file. This ZIP file has an .mdl extension and contains the following files:

- **Metadata Loader XML file:** This file contains the objects extracted from the workspace and formatted in XML. It has the same name as the name of the ZIP file, but with the extension .mdx.
- **Catalog:** The catalog file is called `mdlcatalog.xml` and it contains internal information about the Metadata Loader XML file.

The Metadata Export Utility enables you to specify a file name and a path for the exported MDL file. For example, you export the repository metadata into a file called `sales.mdl`. When you unzip this MDL ZIP file, you obtain two files. The file `sales.mdx` contains the repository objects. The file `mdlcatalog.xml` contains internal information about the MDL XML file.

You can export an entire project, collections, public objects, locations, or any subset of objects. If you export a subset of objects, the MDL exports definitions for each object that you have selected and the parent objects to which the subset belongs. This enables the MDL to maintain the tree relationships for those objects during metadata import.

For example, if you export a single dimension, the export file contains definitions for the dimension, the module to which the dimension belongs, the project to which the module belongs, and the configurations for the dimension.

If you are exporting a subset of objects, ensure that you both export and import all referenced object. You may export the objects referenced by a set of objects by selecting the **Export All Dependencies** option on the Metadata Export dialog box. For example, if you export a table `EMPLOYEES` and it contains a foreign key reference to the table `DEPARTMENTS`, you may choose to export both tables.

Metadata Import Utility

The Metadata Import Utility reads the metadata information from an exported MDL file and creates, replaces, or merges the metadata objects into a workspace. It imports information that belongs to exported metadata objects, such as table columns and their constraints, data loading configuration parameters, and named attribute sets. The Metadata Import Utility enables you to import repository objects even if the references for those objects cannot be satisfied.

You can use the Metadata Import Utility to import objects into a project or a collection. The Metadata Import Utility only reads files created by the metadata export utility.

If the MDL file being imported was created using an earlier product version, the Metadata Import Utility automatically upgrades it to the current version.

See Also: ["Upgrading Metadata from Earlier Releases"](#) on page 12-7
for more information about the automatic upgrade of MDL files

Multiple Session Concurrency and MDL

The repository allows multiple clients to access the same workspace concurrently. Warehouse Builder uses locks to allow only one client to change repository objects. While an object is locked, other clients can only view it in the state it existed after the last transaction instigated by any user is committed.

When replacing or merging objects, the MDL acquires locks on the primary objects that exist both in the repository and in the MDL file. Primary objects include, but are not limited to, projects, modules, tables, dimensions, cubes, mappings, views, and flat files. Secondary objects, such as columns and mapping attributes, are not locked. If locks cannot be obtained because other users are locking the primary objects, then the import fails. Therefore, you must be able to hold locks for primary objects that you are importing. To ensure a successful metadata import, you may need to be the sole client accessing the workspace.

The MDL saves changes made to the workspace after a successful metadata import (any import with no error messages, including imports with only information or warning messages). The MDL also executes a rollback after an unsuccessful import.

Metadata Loader Log File

Whenever you export or import repository metadata, the MDL writes diagnostic and statistical information to a log file. You can specify the location of the log file when you call the MDL.

The log file enables you to monitor and troubleshoot export and import activities in detail and contains the following information:

- Name of the data file
- Start time and end time of the export or import
- Time taken for the export or import in hours, minutes, and seconds (in hh:mi:ss format) or milliseconds
- Object types exported or imported
- Number of objects of each object type exported or imported

The import log file also displays the total number of objects that have been added, replaced, skipped, and deleted.

- Status messages

Status messages provide information about the import or export process. They are of the following types:

- **Informational:** Provides information about the import or export, such as missing metadata objects, whether or not objects were imported, and any reasons why objects were not imported or exported.
- **Warning:** Cautions you about the import or export of an object but does not indicate a failed or terminated export or import. A warning notifies you of the possibility of unexpected results that could occur as a result of the export or import.
- **Error:** Indicates that the MDL export or import was terminated and did not complete successfully. The error message provides a brief description of the reason for the failure.

About Metadata Loader Results

When you use the Metadata Loader Export or Import utilities, you can view the results of a successful export or import task. Use the Metadata Export Results dialog box or the Metadata Import Results dialog box to ensure that all of the objects were exported or imported. To view the results dialog box, click **Show Details** on the Metadata Export Progress dialog box or the Metadata Import Progress dialog box. This displays the Message Log. Click **Show Statistics** at the end of this log.

The results dialog box contains the following information:

- The name of the project exported or imported (if applicable).
- The number of objects of each type exported or imported.
- The number of objects of each object type skipped.

Details about the number of skipped objects is displayed only when you import metadata.

Managing Versions and History with the Metadata Loader

Using the Metadata Loader (MDL) utility, you can import and export metadata from any object in the Projects Navigator, Globals Navigator, and Connection Navigator. You can then move exported files into a third-party version control tool such as Oracle Repository, ClearCase, or SourceSafe. You can enter annotations for your MDL export file to keep track of the information contained in the file.

The Metadata Loader (MDL) enables you to populate a new repository and transfer, update, or restore a backup of existing repository metadata. You can copy or move metadata objects between repositories, even if those repositories reside on platforms with different operating systems.

Accessing the Metadata Loader

You can access the Metadata Loader using either the graphical user interface described in this section or using the OMB Plus scripting language described in Oracle Warehouse Builder API and Scripting Reference.

While the graphical interface guides you through the most commonly performed export and import tasks, the OMB Plus scripting language enables you to perform more specialized export and import tasks and enables you to manage a control file.

About Metadata Loader Control Files

When you use the OMB Plus commands related to the Metadata Loader, a control file provides you with greater control over how objects are imported or exported. For example, by default, the Metadata Loader exports objects into a binary zip format. To override the default of exporting to a zip file, use `OMBEXPORT MDL_FILE` with the `CONTROL_FILE` option with a control file that contains the option `ZIPFILEFORMAT=N`.

See *Oracle Warehouse Builder API and Scripting Reference* for more information about using a control file, the Metadata Loader, and information about each of the following commands:

- `OMBIMPORT`
- `OMBEXPORT`
- `OMUIMPORT`

- OMUIMPORT MDL_FILE
- OMBIMPORT MDL_FILE
- OMUEXPORT MDL_FILE
- OMBEXPORT MDL_FILE

Managing Security

This section discusses how to implement security options for Oracle Warehouse Builder.

This section includes the following topics:

- [About Metadata Security](#)
- [Evaluating Metadata Security Strategies](#)
- [Registering Database Users](#)
- [Editing User Profiles](#)
- [Support for a Multiple-user Environment](#)
- [Defining Security Roles](#)
- [Editing Role Profiles](#)
- [Applying Security Properties on Specific Metadata Objects](#)
- [Security Enforcement](#)
- [Managing Passwords in Warehouse Builder](#)

About Metadata Security

Warehouse Builder enables you to define security on the metadata stored in the design repository. Warehouse Builder metadata security operates in conjunction with Oracle Database security, with Oracle Database provides security for data, while Warehouse Builder provides security for the metadata.

In addition to being registered in the repository, all users must also be database users in the design repository database. Database users may access the data in the database by using SQL*Plus, but they cannot have access to Warehouse Builder and its metadata unless they are also registered in Warehouse Builder.

Metadata security is both optional and flexible. You may choose not to apply any metadata security controls, or define a metadata security policy. You have the option to define multiple users, and apply either full security control or none. You may also implement a custom security strategy based on the security service. After you define a custom security strategy, you may adapt it over time to be more or less restrictive.

The topics in this section describe how to implement metadata security using the Design Center. You may also implement security through OMB Plus. For more information, refer to the Oracle Warehouse Builder API and Scripting Reference.

About the Security Service

Only users with administrative privileges can access the security service under Globals Navigator to manage users and roles of the security policy in Warehouse Builder.

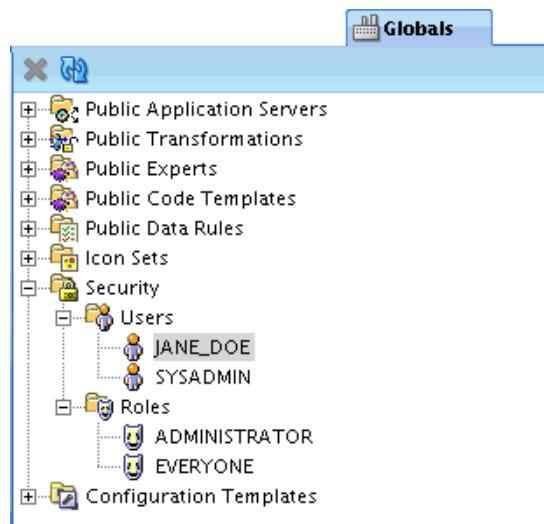
When you install Warehouse Builder and then use the Repository Assistant to create a design repository, Warehouse Builder makes the design repository owner the default administrator. The first time you start the Design Center after installation, you must log in as the design repository owner. You can then define additional administrators or other users as necessary.

When you log into the Warehouse Builder Design Center as the design repository owner, it displays the Globals Navigator.



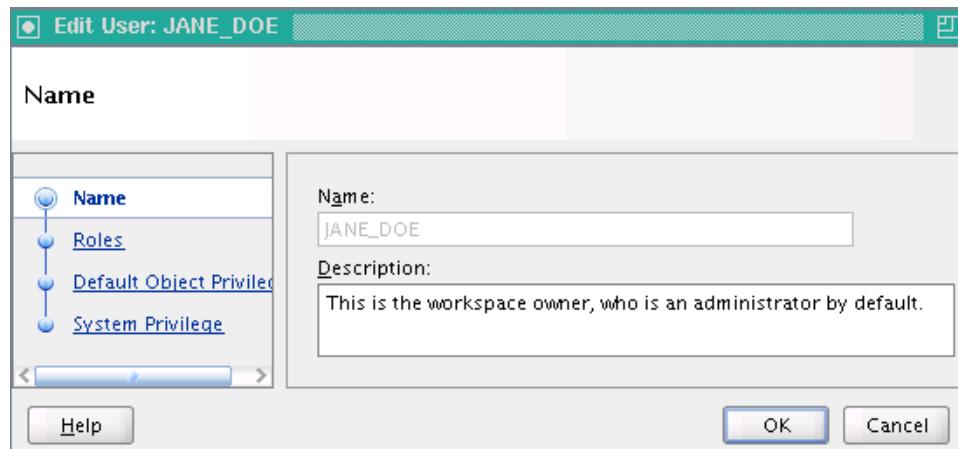
To view default security settings:

1. In Globals Navigator, expand Security.
2. Expand Users, and then expand Roles.



3. Notice that there are two predefined roles, ADMINISTRATOR and EVERYONE.
4. The one predefined user is the design repository owner; it is assigned the ADMINISTRATOR role by default.

To view or edit the details for a user, in the globals Navigator, under Security and then under Users, select that and double-click the user. The Edit User screen appears.



For a complete list of all the tasks administrators can perform, see "[Administrator Role](#)" on page 13-18.

Evaluating Metadata Security Strategies

Warehouse Builder enables you to design a metadata security strategy that fits your implementation requirements. As you define your metadata security strategy, recognize that more restrictive policies are more time consuming to implement and maintain.

Consider modeling your strategy based on one of the following security strategies:

- [Minimal Metadata Security Strategy \(Default\)](#)
- [Multiuser Security Strategy](#)
- [Full Metadata Security Strategy](#)

Minimal Metadata Security Strategy (Default)

Minimal metadata security is the default security policy when you create a new design repository. As your project requirements change over time, you may apply other metadata security strategies. For example, you may not need extra metadata security if you are implementing an internal pilot project, or if you anticipate only few trusted users.

In the case of a minimal metadata security strategy, all users may log into Warehouse Builder with the same user name and password, that of the design repository owner. Oracle Database security policies keep the data in the design repository secure, and the metadata is available to anyone who knows the design repository owner logon information. All users can create, edit, and delete all objects.

Multiuser Security Strategy

If your implementation has multiple users and you want to track who performs what operations, implement a multiuser security strategy. This strategy restricts to a single user the rights and access granted to the design repository owner. Although this strategy does not restrict user access to metadata objects, you can apply restrictions at a later date.

To implement security for multiple users:

Log into Warehouse Builder as an administrator and complete the instructions in the following sections:

1. [Registering Database Users](#) on page 13-4
2. [Editing User Profiles](#) on page 13-12

Full Metadata Security Strategy

This section describes a process for applying all the metadata security options available in Warehouse Builder. You can enable all or some of these options. For instance, you could take steps one through three but ignore the remaining steps.

Be sure to edit the security properties for all projects in the Project Navigator. By default, the EVERYONE role has FULL_CONTROL object privileges. To change this, select the project and then, from the View menu, select **Security**. Edit the privileges to the EVERYONE role to be more restrictive, and then press **Propagate Security Settings** icon on the upper left corner. This action applies the new restrictions to all children of this project. For newly created projects and other objects, use the default object privilege setting of OWB users to define access privileges.

To implement full metadata security for multiple users:

Log into Warehouse Builder as an administrator and complete the instructions in the following sections:

1. Set the parameter **Default Metadata Security Policy** to maximum.
In the Design Center select **Tools, Preferences**, expand **OWB**, and then select **Security Parameters**.
 2. [Registering Database Users](#) on page 13-4
 3. [Editing User Profiles](#) on page 13-12
- The **Default Metadata Security Policy** you set in step one of these instructions is not retroactive. It applies only to users you register after changing the setting. You must manually edit the profiles of preexisting users.
4. [Defining Security Roles](#) on page 13-17
 5. [Editing User Profiles](#) on page 13-12
 6. [Applying Security Properties on Specific Metadata Objects](#) on page 13-22

Registering Database Users

All Warehouse Builder users must also be Oracle Database users.

You can create new OWB users in one of two ways:

- Use Warehouse Builder to register existing database users, or to create new ones.
Note that you must have the database CREATE USER privilege to create a new user.
- Create new database users and then register them in Warehouse Builder.
Note that even though it is possible to create users in SQL Plus, Oracle recommends that you create users through the Warehouse Builder interface. This ensures that users are assigned all necessary roles and privileges.

For security reasons, you cannot register database administrator users, for example SYS. Also, the database default role settings must not be set to ALL. Note that OWB automatically sets the "database default role setting" for new users. You may change the database default role settings as described in "[Changing Database Default Roles](#)" on page 13-12.

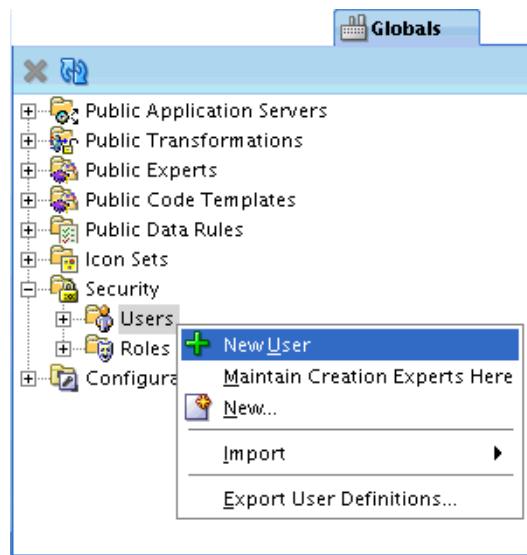
Note: The OWB user name can be alpha, alpha-numeric or all numeric characters.

Registering Existing Database Users

This section explains how to register existing database users.

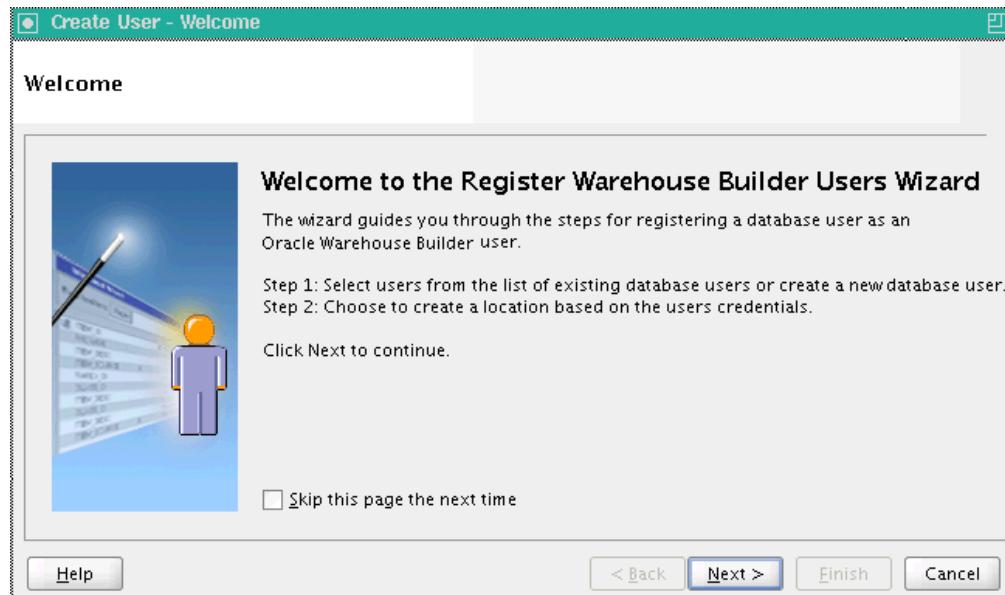
To register existing database users:

1. In the Globals Navigator, under Security, right-click Users and select **New User**.



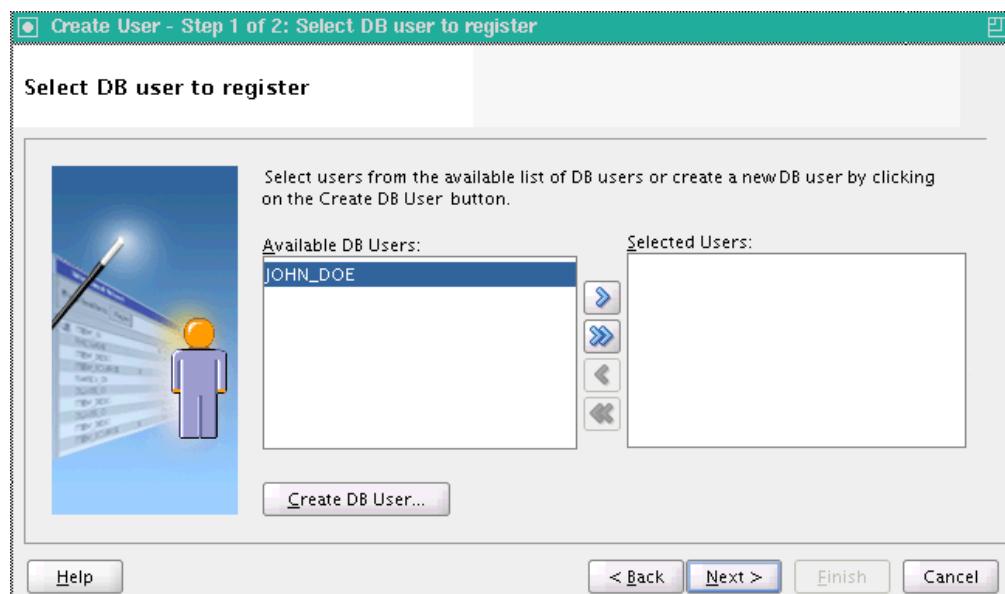
The Create User: Welcome screen appears.

2. On the Create User Welcome screen, click **Next**.



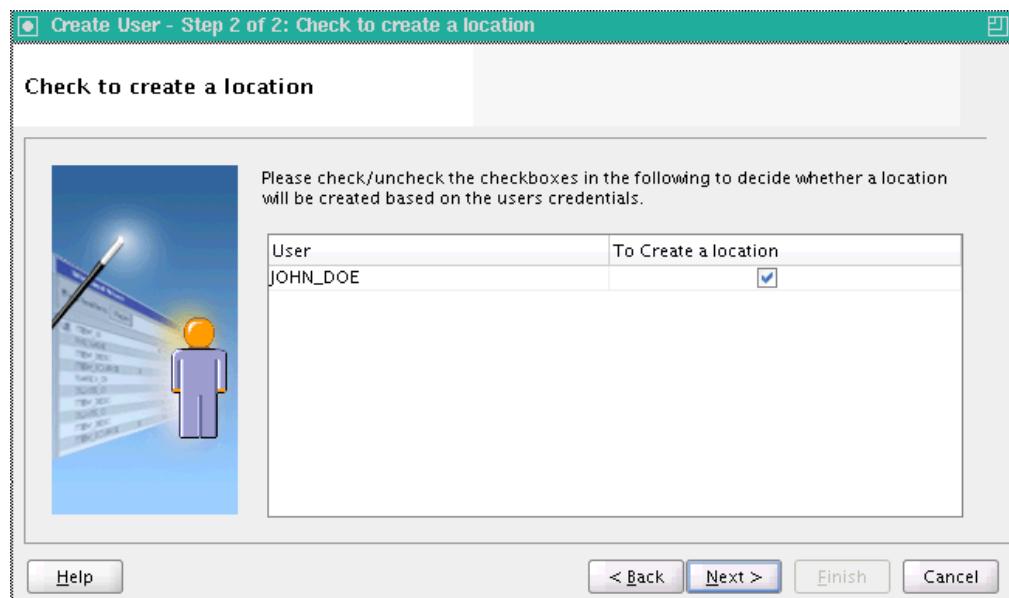
3. On the Select DB user to register screen, under **Available DB Users**, select the user or users you want to register, and click the appropriate transfer icon to add the user or users to the **Selected Users** list.

Click **Next**.



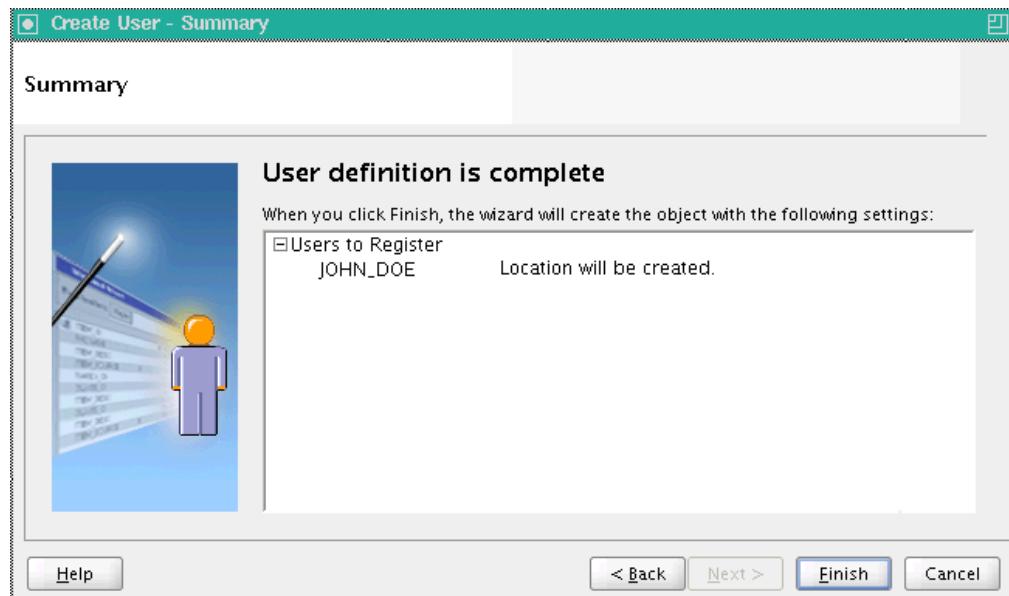
4. On the Check to create a location screen, check Create option next to the user you are registering.

Click **Next**.

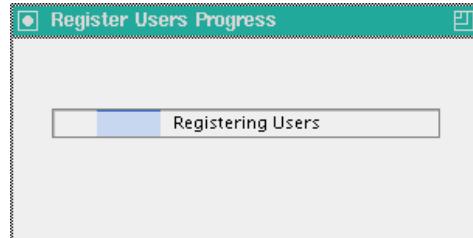


5. On the Summary screen, review the new user definition.

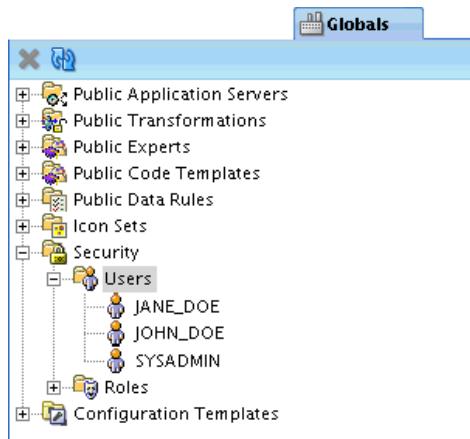
Click **Finish**.



6. The Register Users Progress appears. It disappears when registration is complete.



7. Note that your new user is now listed under Users.

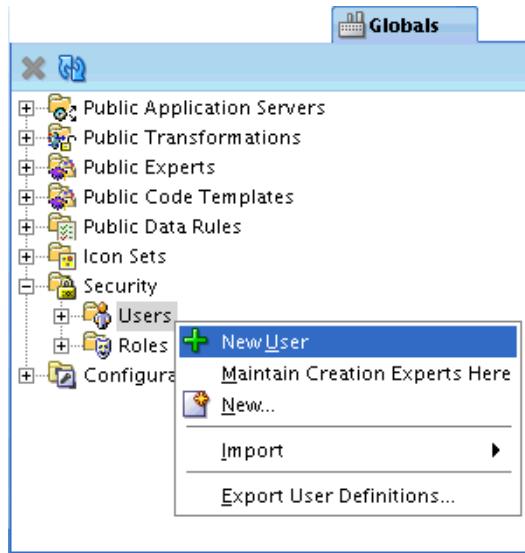


Creating New Oracle Database Users

This section explains how to create new database users. You must have the database system privilege CREATE USER.

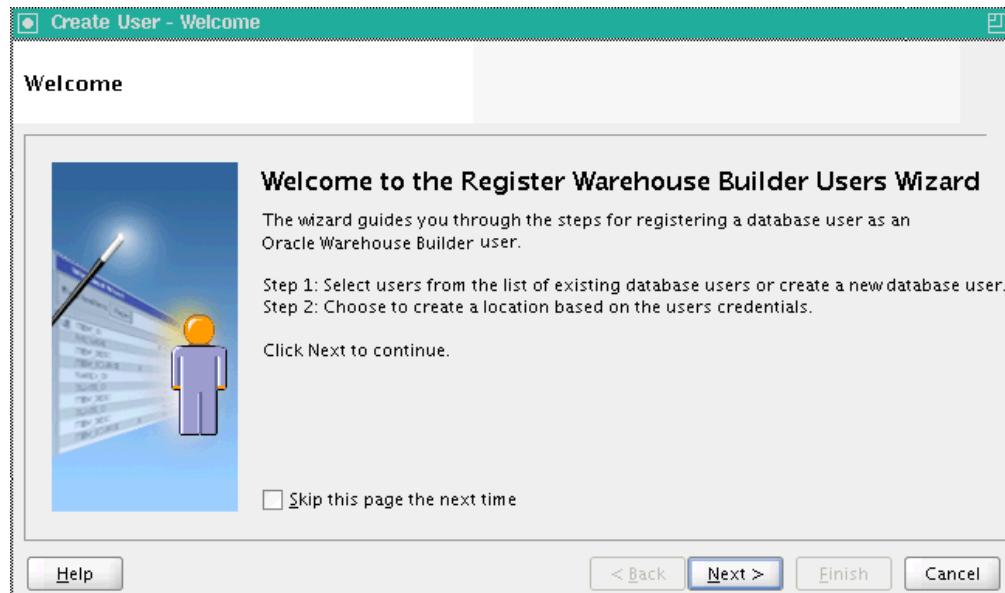
To create a new database user:

1. In the Globals Navigator, under Security, right-click Users and select New User.

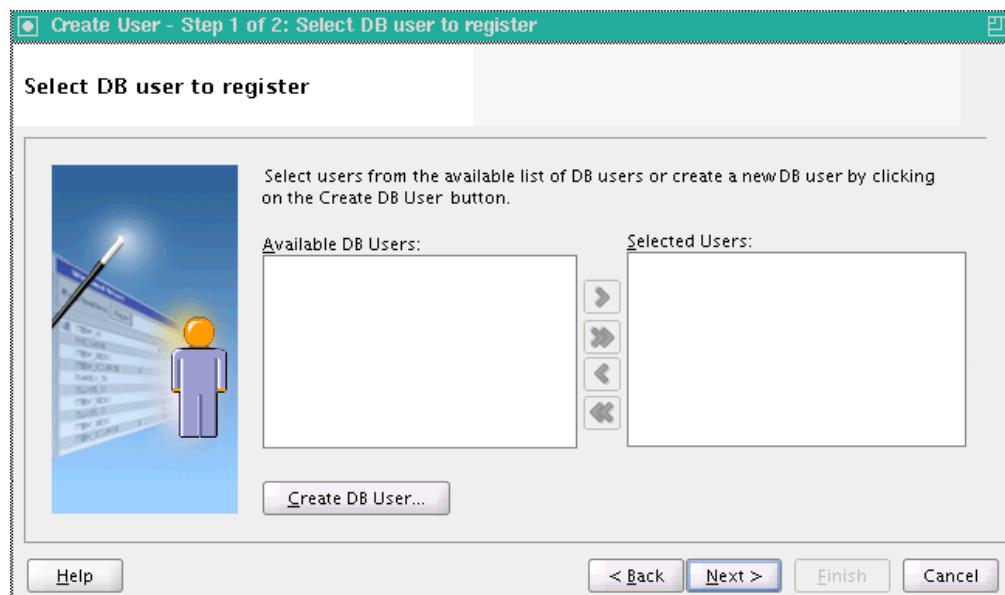


The Create User: Welcome screen appears.

2. On the Create User Welcome screen, click **Next**.



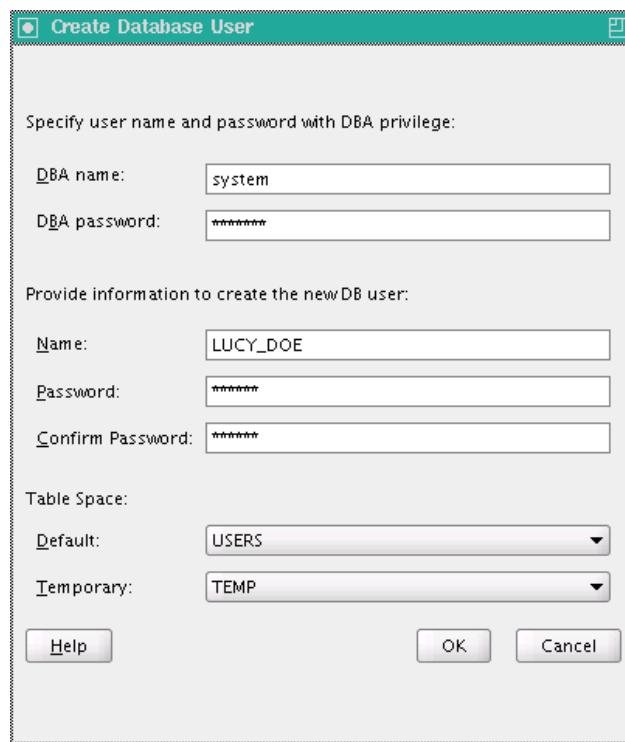
3. On the Select DB user to register screen, click **Create DB User**.



4. On the Create Database User screen, enter the **DBA password**, and the **Name** and **Password** (with confirmation) of the new user.

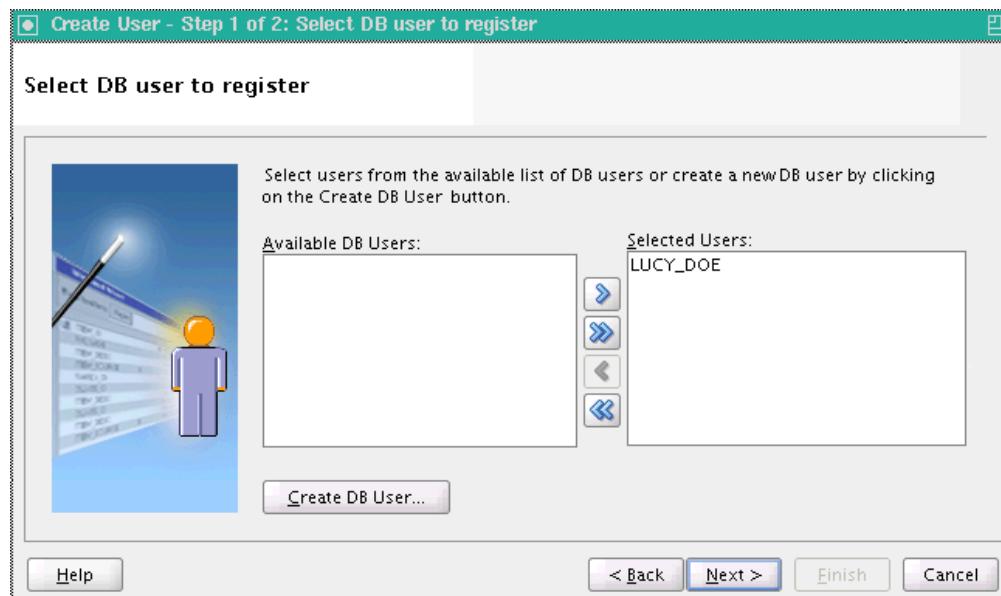
Click **OK**.

Note that you must specify a valid user name and password, and adhere to the security standard implemented on the Oracle Database. For more information about user names, passwords, and password complexity verification routines, refer to *Oracle Database Security Guide*.



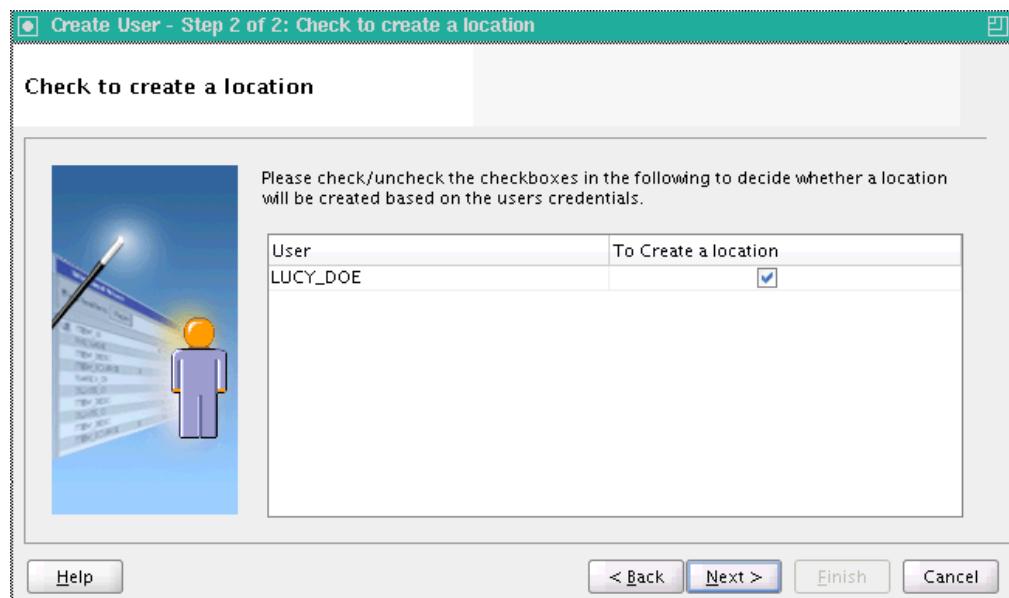
5. Note that on the Select DB user to register screen, the new user is automatically added to the Selected Users list.

Click **Next**.



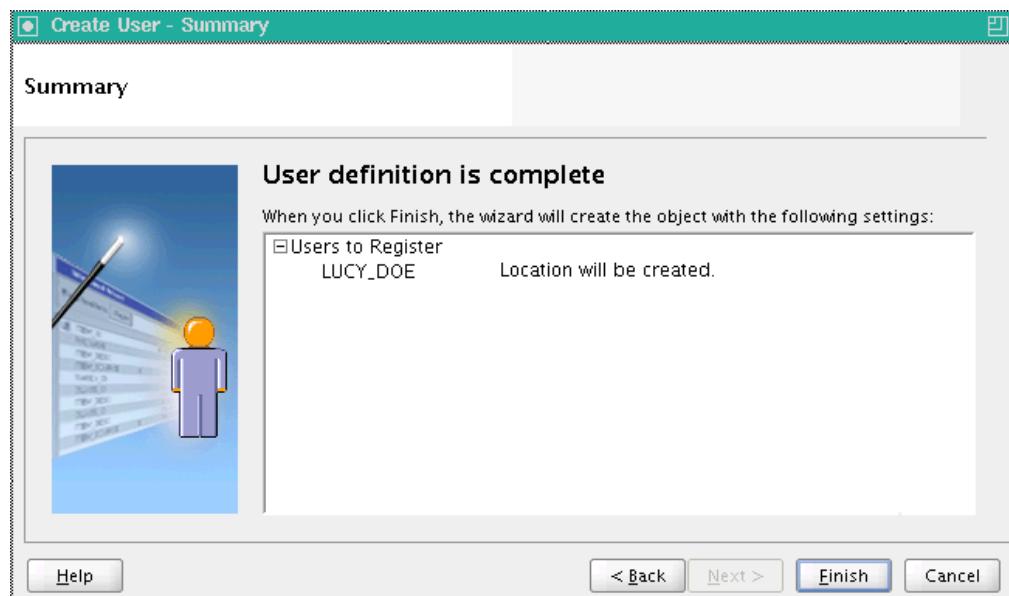
6. On the Check to create a location screen, check Create option next to the user you are registering.

Click **Next**.



7. On the Summary screen, review the new user definition.

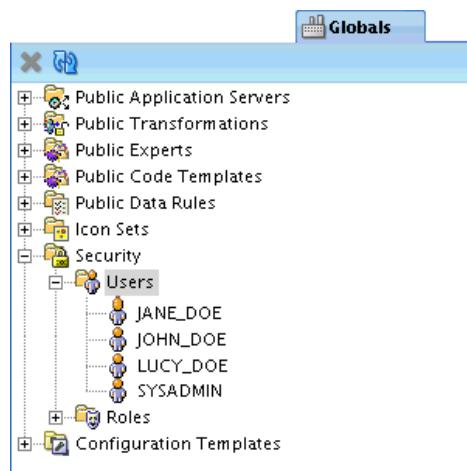
Click **Finish**.



8. The Register Users Progress appears. It disappears when registration is complete.



9. Note that your new user is now listed under Users.



Changing Database Default Roles

For security reasons, you cannot register database users that have ALL default roles in the database. However, it is possible to change this default setting by correcting the role assignment. There are two options: [Fix Now](#) and [Fix Later](#).

Fix Now

If you select the **Fix Now** option, type the user name and password with SYSDBA privileges. The user is registered, and the necessary commands are issued.

For example, when you register new users, the database role *OWB_repository_name* is assigned to each user. For security reasons, this role must not be the default role of any registered user. If you attempt to register a user under these conditions and then select **Fix Now**, the user is registered and the following command is issued:

```
alter user username default role all except OWB$CLIENT
```

Fix Later

If you select the **Fix Later** option, the user is not registered. You must manually change the default role setting in the database using SQL, and then register the user in OWB. To manually change the setting, connect to the database as a user with the ALTER USER system privilege and issue the required commands.

Note the following SQL script for changing the default roles of selected users. It changes the default role setting so that any role subsequently granted to the user cannot be the default role of that user. To change this, register the user and then issue a the following command:

```
alter user username default role all except OWB$CLIENT
```

Editing User Profiles

For each user, you can enter an optional description, assign the user to existing [Roles](#), specify the [Default Object Privilege](#) and the [System Privileges](#).

These are Oracle Database users, so you cannot rename a user in OWB; you must do that through Oracle Database.

Note that the granting or revoking of roles and privileges only takes effect in the next session OWB.

To edit a user profile:

1. In the Globals navigator, expand **Security**.
2. Expand **Users**.
3. Select the name of the user for editing. Right-click the user name, and select **Open**.
4. The Edit User: Username screen appears. It contains the following options for editing
 - **Name:** you cannot change the name itself, but the screen contains an editable **Description** text field.
 - **Roles:** you may assign various roles to the user by moving them from the list of Available Roles to the list of Granted Roles.
 - **Default Object Privilege:** you may assign default privileges to either Users or Roles by checking the appropriate boxes under **FULL_CONTROL**, **EDIT**, **COMPILE**, or **READ**.
 - **System Privilege:** you may assign system privileges to the user by checking appropriate boxes under Object System Privilege (**ACCESS_PUBLIC_VIEW_BROWSER**, **CREATE_PLATFORM**, **CREATE_PROJECT**, or **CREATE_SNAPSHOT**) and Control Center System Privilege (**CONTROL_CENTER_DEPLOYMENT**, **CONTROL_CENTER_EXECUTION**, or **CONTROL_CENTER_VIEW**).
5. When the edits are complete, click **OK**.

Roles

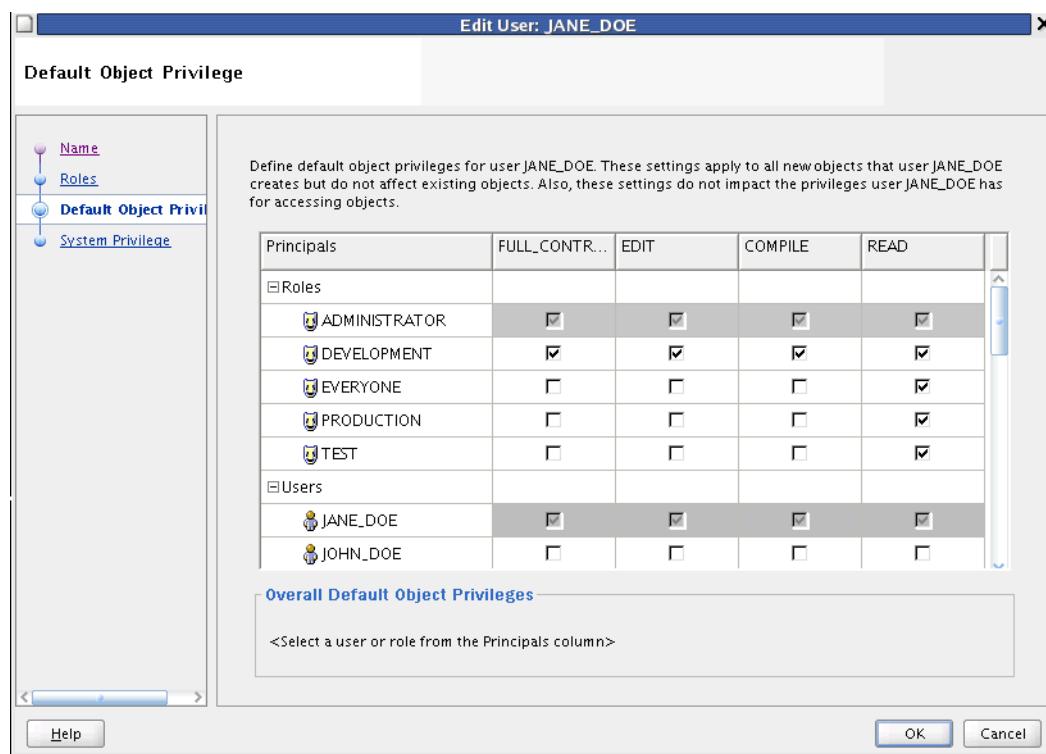
You can assign a user to one or more roles. If you assign multiple roles with conflicting privileges, then the user is granted the more permissive privilege, which is the union of all the privileges granted to the multiple roles. For example, if you assign to the same user a role that allows creating a snapshot and a role that restricts it, then the user is allowed to create snapshots.

If you want to assign a user to a role that does not display on the Available Roles List, close the editor, create the new role, and then edit the user account. To create a new role, right-click Roles under the Security node in the Globals Navigator and select **New Role**. For information on creating and editing roles, see [Defining Security Roles](#) on page 13-17 and [Editing Role Profiles](#) on page 13-19.

Default Object Privilege

Default object privileges define the access other users and roles have to objects that the selected user creates. These privileges do not impact the privileges the user has for accessing objects created by other users.

For example, for all objects that JANE_DOE creates, JANE_DOE, as well as ADMINISTRATOR and DEVELOPMENT roles, have full access. Note that EVERYONE, PRODUCTION, and TEST roles are restricted to read-only.



If you are familiar with UNIX operating system security, note that the default object privilege is similar to the UMASK command. When you edit the default object privilege, the change only effects objects the user creates subsequently; there is no effect on previously created objects. Therefore, if you set default object privileges early, you can expect little or no additional object-level security management.

To define the privileges other users have to objects the selected user creates, check the appropriate box for each role or user. You can grant the following privileges: **FULL CONTROL**, **EDIT**, **COMPILE**, and **READ**. All the privileges are additive. If you select **COMPILE**, then you apply both the compile and read privileges.

Note that access may be granted both to roles and to individual users. Note, however, that when you grant access to a role, the privilege is extended to all users in that role. For example, even though JOHN_DOE is not specifically granted access, he has read access through the EVERYONE role. Furthermore, if JOHN_DOE is a member of the DEVELOPMENT role, he has full control and access.

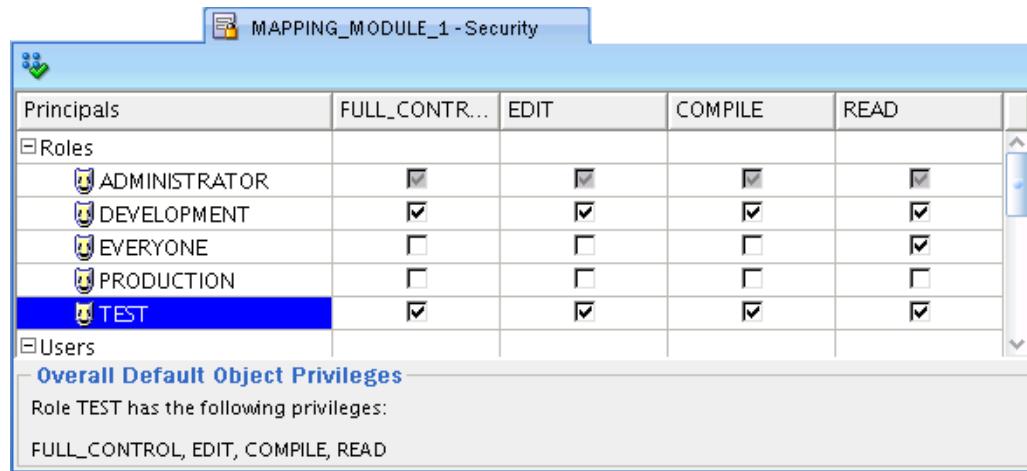
By default, when you create a new user, the EVERYONE role has full control on all objects. To enable metadata security, edit all user profiles and restrict the access the EVERYONE role has to objects each user creates.

Securing Metadata Objects Throughout their Life Cycle

Default object privileges work in conjunction with object security properties to provide security options throughout the life cycle of a given metadata object. Settings you specify on the Default Object Privilege tab persist until a qualified user overrides the restrictions, on an object-by-object basis.

Assume that JANE_DOE creates several mappings. When JANE_DOE designs and develops these objects, the security policy described earlier in this section may be desirable. However, assume that JANE_DOE completes the mappings and releases the objects to the testing team. In this case, the default object privilege is too restrictive. To extend access to the TEST role, JANE_DOE can select the mapping, then from the View

menu, select **Security**. She can then manually add all necessary privileges to the TEST role.



The screenshot shows a software interface titled "MAPPING_MODULE_1 - Security". It displays a table of principal assignments across four columns: FULL_CONTROL, EDIT, COMPILE, and READ. The rows are categorized into "Roles" and "Users". The "TEST" role is highlighted with a blue selection bar. The "Overall Default Object Privileges" section indicates that the TEST role has been granted the privileges: FULL_CONTROL, EDIT, COMPILE, and READ.

Principals	FULL CONTR...	EDIT	COMPILE	READ
Roles				
ADMINISTRATOR	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
DEVELOPMENT	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
EVERYONE	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
PRODUCTION	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
TEST	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Users				
Overall Default Object Privileges				
Role TEST has the following privileges:				
FULL_CONTROL, EDIT, COMPILE, READ				

For more details on overriding the default security on an object by object basis, see "["Applying Security Properties on Specific Metadata Objects"](#) on page 13-22.

Object Privileges

Object privileges apply to all metadata objects in the repository including projects, modules, and collections.

FULL CONTROL

Full control includes all the other privileges plus the ability to grant and revoke privileges on an object. Only users with full control over an object can override default security on an object-by-object basis as described in "["Applying Security Properties on Specific Metadata Objects"](#) on page 13-22.

EDIT

The edit privilege includes the compile, and read privileges. Additionally, edit allows users to delete, rename, and modify an object.

COMPILE

The compile privilege includes the read privilege and enables you to validate and generate an object.

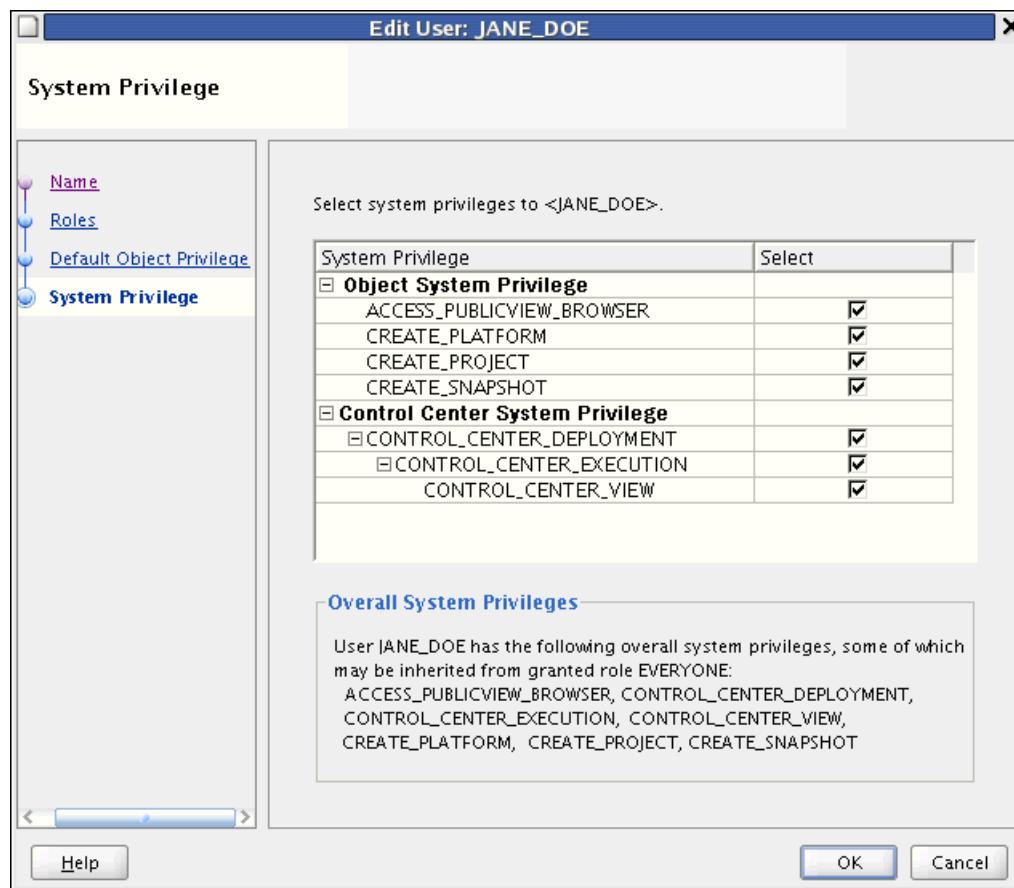
READ

The read privilege enables you to view an object.

System Privileges

System privileges define user access to workspace-wide services. Use the System Privilege tab to allow or restrict users and roles from performing administrative tasks.

You can control access to the following operations:



- **ACCESS_PUBLICVIEW_BROWSER:** Allows users to access the Repository Browser.
- **CREATE_PLATFORM:** Allows users to create new platforms in the workspace using OMB*Plus.
- **CREATE_PROJECT:** Allows users to create projects, which administrators create projects as a means of organizing metadata objects.
- **CREATE_SNAPSHOT:** Allows users to create snapshots which administrators use when backing up workspaces.
- **CONTROL_CENTER_DEPLOYMENT:** Allows users to deploy to the Control Center and then run those procedures.
- **CONTROL_CENTER_EXECUTION:** Allows users to run procedures from the Control Center.
- **CONTROL_CENTER_VIEW:** Allows users to view procedures from the Control Center.

Support for a Multiple-user Environment

Warehouse Builder enables multiple users to access the same Warehouse Builder repository at the same time by managing read/write privileges. Only one user is given write privileges to an object at any given time. All other users can have read-only access. If a user has write access to an object, Warehouse Builder maintains a lock on the object while the object editor is open. If no changes were made to the object, then the lock is released as soon as the object editor is closed. If changes were made, then

the lock is maintained until the user closes all editors associated with the object and either saves the changes or reverts to the last saved version. Other users cannot delete an object while it is in use.

ReadWrite Mode

Whenever you open an editor, property sheet, or dialog box, you access objects in read/write mode by default. Your changes are available to other users only after you save them to the repository.

Read-Only Mode

If you attempt to open an object locked by another user, or if you have only READ permissions for the object, then Warehouse Builder displays a message that prompts you either to cancel the request or access the object in read-only mode. If you choose to continue in read-only mode.

A user who is editing an object in READ/WRITE mode may save changes while a user with read-only privileges views the object. To synchronize the object with the repository, click **Refresh**.

Defining Security Roles

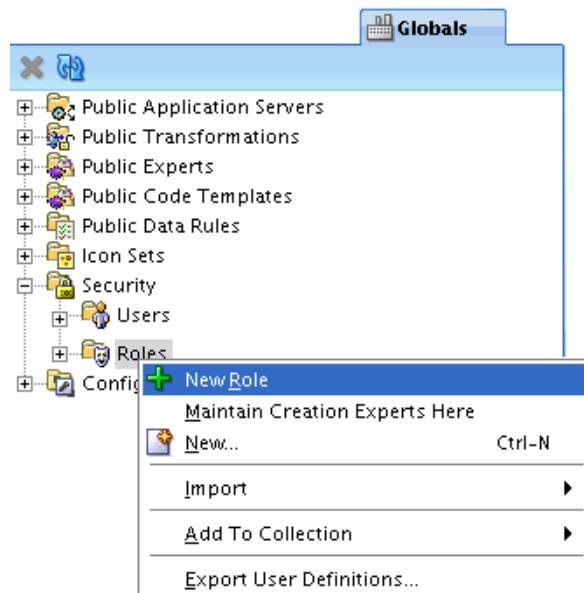
You can use roles to represent groups of users with similar responsibilities and privileges. Unlike users which are also database users, these roles are not database roles. These roles are purely design constructs for implementing security within the product.

Roles enable you to more efficiently manage privileges because it is more efficient to grant or restrict privileges to a single role rather than multiple users.

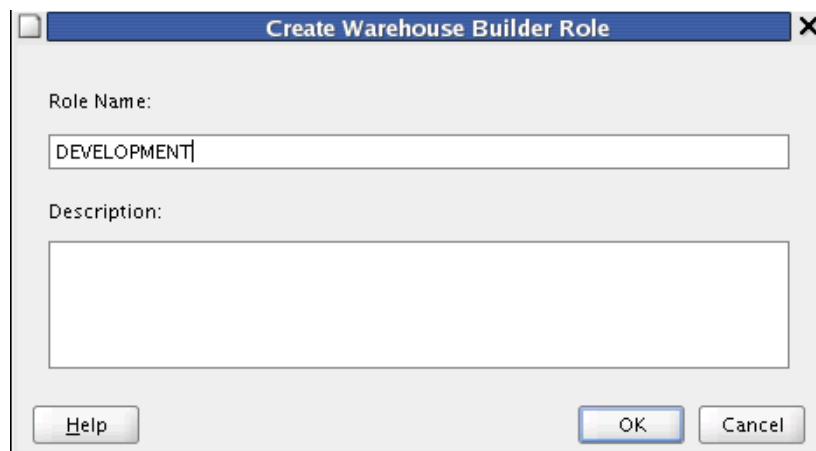
The [Everyone Role](#) and the [Administrator Role](#) are predefined roles. You edit the privileges but cannot delete or rename the predefined roles.

To create a new role:

1. In the Globals navigator, expand **Security**.
2. Under **Security**, select **Roles**.
3. Right-click **Roles**, and select **New Role**.



4. On the Create Warehouse Builder Role screen, enter the **Role Name**.



5. Click **OK**.

Everyone Role

Use this role to easily manage privileges for all users. When you register new users, Warehouse Builder assigns those users to the EVERYONE role by default.

Administrator Role

Administrators in Warehouse Builder can perform various security tasks, such as:

- [Registering Database Users](#) on page 13-4
- [Editing User Profiles](#) on page 13-12
- Changing User Passwords

You cannot change user passwords from within Warehouse Builder. Change passwords directly in the Oracle Database as described in *Oracle Database Security Guide*.

- [Defining Security Roles](#) on page 13-17
- [Editing Role Profiles](#) on page 13-19
- Deleting Users and Roles

You may delete users by right-clicking in the Globals navigator, and selecting **Delete**. You may delete all OWB users except the repository owner. Note that this does not delete or alter the user account in the Oracle Database.

You can delete all OWB users except for the design repository owner. Deleting a user from OWB does not delete or alter the user account on the Oracle Database.

You can delete all OWB roles except ADMINISTRATOR and EVERYONE roles. Deleting a role from OWB does not delete or alter roles in the Oracle Database.

- Renaming Roles

From the Globals Navigator, right-click a role and select **Rename**. You can rename all roles except the predefined administrator and everyone roles.

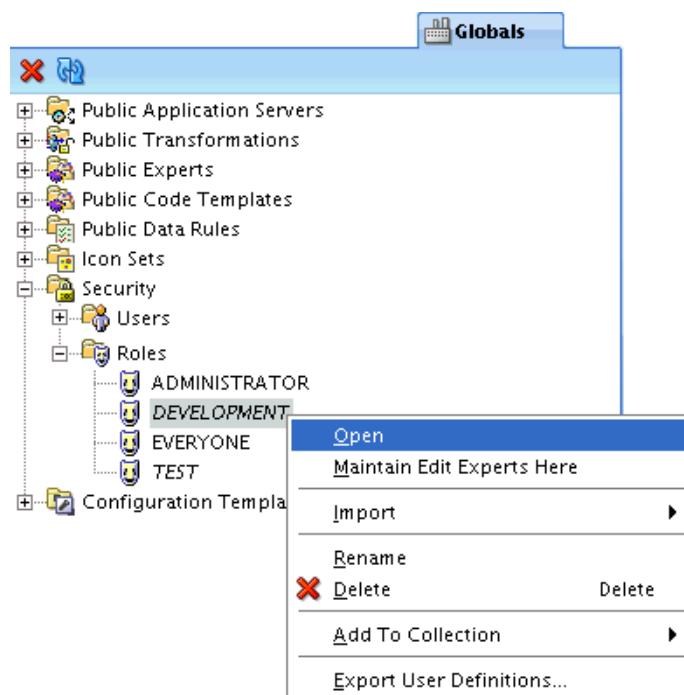
- [Applying Security Properties on Specific Metadata Objects](#) on page 13-22

Editing Role Profiles

For each role that you create, you can edit the name, enter an optional description, assign the role to existing [Users](#), and specify the system privilege. You cannot rename or edit the descriptions for the predefined roles EVERYONE and ADMINISTRATOR, nor can you delete them. Note that Warehouse Builder roles and database roles are separate constructs; therefore, deleting a Warehouse Builder role has no effect on the database. For more information on system privilege, see [System Privileges](#) on page 13-15.

To alter default security privileges for a role:

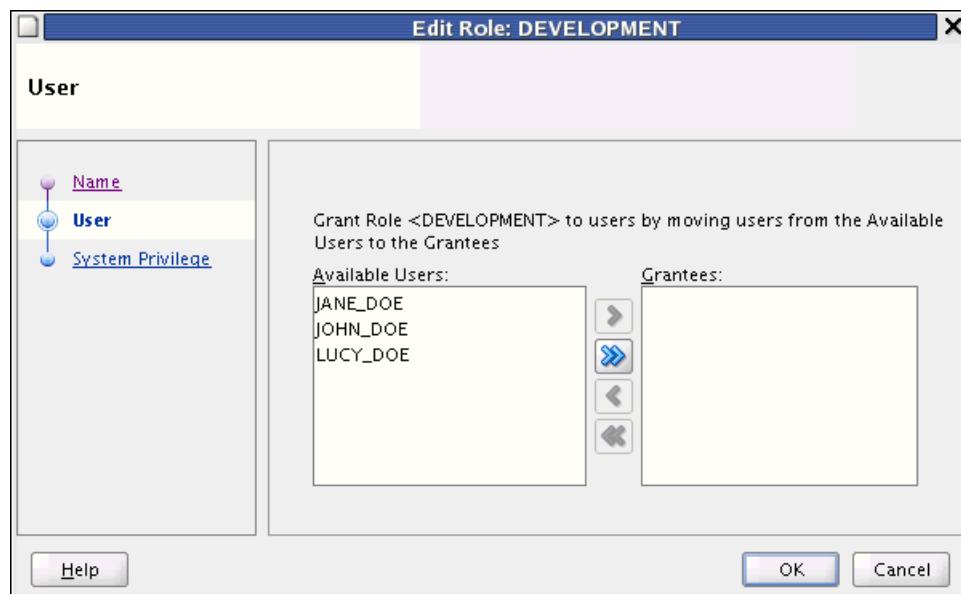
1. In the Globals navigator, expand **Security**.
2. Under **Security**, expand **Roles**.
3. Select the role to edit, and right-click. From the menu, select **Open**.



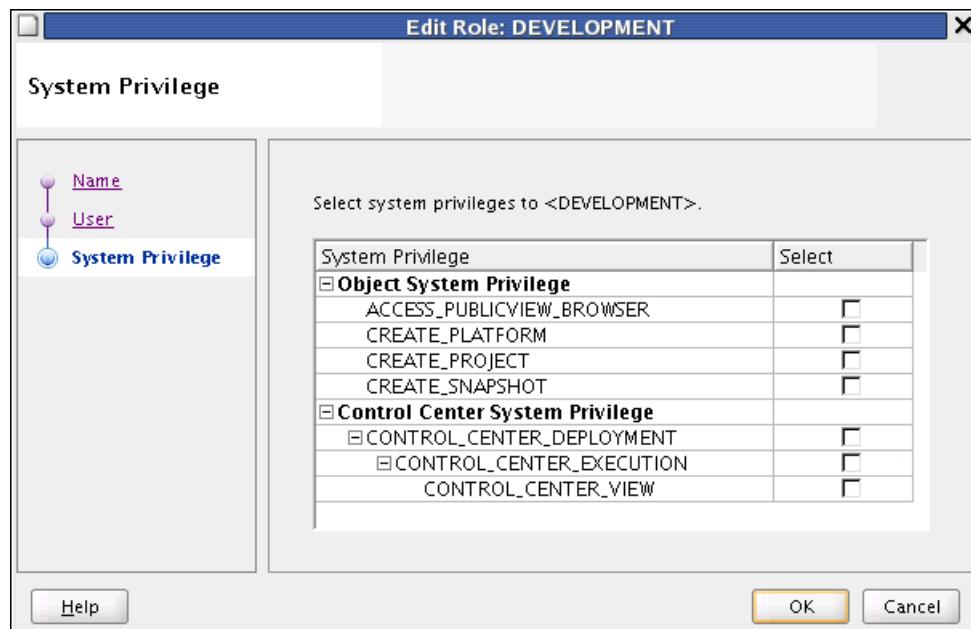
4. In the *Edit Role: RoleName* window, you may do any one of the following:
 - **Name:** you cannot change the name itself, but the screen contains an editable **Description** text field.



- **User:** you may assign various users to the role by moving them from the list of **Available Users** to the list of **Grantees**.



- **System Privilege:** you may assign system privileges to the role by checking appropriate boxes under Object System Privilege (**ACCESS_PUBLIC_VIEW_BROWSER**, **CREATE_PLATFORM**, **CREATE_PROJECT**, or **CREATE_SNAPSHOT**) and Control Center System Privilege (**CONTROL_CENTER_DEPLOYMENT**, **CONTROL_CENTER_EXECUTION**, or **CONTROL_CENTER_VIEW**).



5. When the edits are complete, click **OK**.

Users

You can assign multiple users to a role. If you want to assign a user that does not display on the Available Users list, then close the editor, create the user from the Security node in the Globals Navigator, and then edit the role. To create a new user, right-click **Users** from the Security node and select **New User**. For information on

creating and editing users, see [Registering Database Users](#) on page 13-4 and [Editing User Profiles](#) on page 13-12.

Applying Security Properties on Specific Metadata Objects

You can grant or restrict access to metadata objects on an object-by-object basis.

To change security properties of a specific metadata object:

1. Select the metadata object for changing.
2. From the View menu, select **Security**.
3. Edit the security privileges for the object, granting and revoking them either at Role level, or at User level.

Principals	FULL CONTR...	EDIT	COMPILE	READ
Roles				
ADMINISTRATOR	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
DEVELOPMENT	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
EVERYONE	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
PRODUCTION	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
TEST	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Users				
JANE_DOE	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
JOHN_DOE	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
LUCY_DOE	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

4. When all changes are made, from the File menu, select **Save All**.

Confirm changes.

Security Tab

Use the Security tab to define metadata security on an object-by-object basis. Only users that have full control privileges on an object can change the metadata access controls on the Security tab. Security properties are important in managing the life cycle of your projects, as described in ["Example: Using Security Properties to Freeze a Project Design"](#) on page 13-23.

While the [Default Object Privilege](#) defines metadata security for objects a specific user creates, the Security tab overrides that metadata security policy on an object-by-object basis. Assume that JANE_DOE is a developer that creates mappings and process flows. If you want all objects created by JANE_DOE made available to another developer, such as JOHN_DOE, then use the [Default Object Privilege](#). However, if you want to make only a few objects created by JANE_DOE available to JOHN_DOE or even every user who has a TEST role, locate each object in the Design Center and alter its security options.

To enforce a full metadata strategy, edit the security properties for all projects in the Project navigator. By default, the EVERYONE role has its object privileges set to full control. Change the EVERYONE role privilege to be more restrictive and select **Propagate Security Settings** icon to apply the changes to all children.

Propagating Security Properties to Child Objects

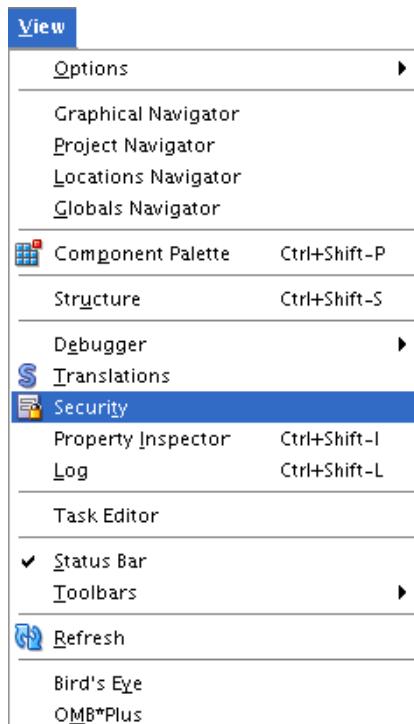
You can apply security properties to an object and all its children by selecting Propagate on the Security tab. This option is disabled when you select an object that cannot have child objects.

Example: Using Security Properties to Freeze a Project Design

When users complete the design of a project, you may want to freeze the contents of the project. Once you complete the following steps, only administrators can change the objects in the project.

To freeze a project design:

1. Log on as user with administrator privileges.
2. From the View menu, select **Security**.



3. On the Security tab, restrict the privileges for all users and roles other than the administrators, as appropriate.
4. Click Propagate Security Settings icon.

Security Enforcement

When any user attempts to perform an operation in Warehouse Builder, Warehouse Builder first verifies that the user has the required privileges to perform the operation. Table 13–1 lists the privileges required to run operations in Warehouse Builder.

Table 13–1 Privileges Required for the Execution of Operations

Warehouse Builder Operation	Security Check
Configure	User must have EDIT privilege on objects to be configured.

Table 13–1 (Cont.) Privileges Required for the Execution of Operations

Warehouse Builder Operation	Security Check
Copy	User must have READ privilege on the object to be copied.
Create object	User must have EDIT privilege on parent. For example, to create a mapping you must have EDIT privilege on the module.
Cut	User must have EDIT privilege on the object to be cut.
Delete	User must have EDIT privilege on the object to be deleted.
Deploy	User must have CONTROL_CENTER_DEPLOY system privilege.
Edit	User must have EDIT privilege on the object to be edited.
Export	User must have READ privilege on objects to be exported. Administrative users can export security information such as roles, users, and privileges when Export security information is enabled.
Generate	User must have COMPILE privilege on object to be generated.
Import	User must have EDIT privilege on objects to be imported. Administrative users can import security information such as roles, users, and privileges when Import security information is enabled.
Move	User must have privileges listed for the Cut and Paste operations.
Paste	User must have EDIT privilege on the parent to receive the copied object.
Rename	User must have EDIT privilege on the object to be renamed.
Snapshot: compare snapshots	To compare with another snapshot or other repository object, user must have READ privilege on that snapshot and the snapshot or other repository object.
Snapshot: restore snapshot	To restore an object based on a snapshot, a user must have READ privilege on that object. To restore a folder, a user must have EDIT privilege on the folder and all of its children.
Snapshot: take snapshot	User must have the CREATE_SNAPSHOT system privilege to create snapshots.
Source import	User must have EDIT privilege on objects to be replaced by imported objects.
Synchronize inbound	User must have READ privilege on the object in the repository and EDIT privilege on the object in the editor.
Synchronize outbound	User must have EDIT privilege on the object in the repository.
Validate	User must have COMPILE privilege on object to be validated.

Managing Passwords in Warehouse Builder

You can manage passwords within Warehouse Builder in the following ways:

- [Credential Memory on Logon Panel](#)
- [Changing Passwords that Access Control Centers](#)
- [Encrypting Passwords to Warehouse Builder Locations](#)

Credential Memory on Logon Panel

The logon dialog that appears when the Warehouse Builder Design Center is launched retains a list of previously used credentials. This is a convenience for Design Center users who frequently work with the same workspaces. The feature enables OWB to remember log in information.

Changing Passwords that Access Warehouse Builder

In keeping with standard security practices, you may want to periodically change the passwords used to access Warehouse Builder repositories.

Changing Passwords that Access Design Repositories

Manage the password to design repositories as you would any other Oracle Database.

Changing Passwords that Access Control Centers

To change the password for a repository that hosts a Control Center and is therefore a deployment environment, you must first stop the Control Center service, run a script to change the password, and restart the Control Center service.

To change the password for a repository that hosts a Control Center:

1. Log on to the Control Center as the repository owner.
2. Stop the Control Center by running the script
OWB_HOME/owb/rtp/sql/stop_service.sql.

The script returns values of Unavailable or Available to indicate the status of Control Center.

3. Change the password by running the script
OWB_HOME/owb/rtp/sql/set_repository_password.sql.
When prompted, specify the new password.
4. Restart the Control Center by running the script
OWB_ORACLE_HOME/owb/rtp/sql/start_service.sql.

Encrypting Passwords to Warehouse Builder Locations

Warehouse Builder users create a location for each database, file server, or application that want to extract or load metadata and data. Locations include the user name and password used to access these various sources and targets. Warehouse Builder can store these passwords in the repository in an encrypted manner. The switch that turns on and off the password storage is Persist Location Password in Metadata, which is located in the Design Center under **Tools, Preferences, Security Parameters**.

The default encryption algorithm utilized is DES56C that is valid for Oracle Database 9i and subsequent versions. If the repository Database is version 10g or later, then you can set the encryption algorithm to 3DES168 or any other more powerful encryption by changing *OWB_HOME/owb/bin/admin/jdbcdriver.properties* file and specifying the following encryption parameters:

```
encryption_client; default = REQUIRED
encryption_types_client; default = ( DES56C )
crypto_checksum_client; default = REQUESTED
crypto_checksum_types_client; default = ( MD5 )
```

For the protocol to work, set the server to the default ACCEPTED mode. For more information, see *Oracle Database JDBC Developer's Guide*.

A

Troubleshooting the Installation and Setup

This section describes some troubleshooting tasks you may need to perform during installation and setup of Oracle Warehouse Builder.

This section includes the following topics:

- "General Steps for Troubleshooting Warehouse Builder" on page A-1
- "Inspecting Log Files in Warehouse Builder" on page A-2
- "Error Messages Related to Installation" on page A-3
- "Troubleshooting Other Installation Problems" on page A-12
- "Checking the Java Virtual Machine (JVM)" on page A-13
- "Using OWB Client Logging to Control Logging and Trace Problems" on page A-13
- "Checking the System Path of Oracle Warehouse Builder" on page A-14
- "Configuring OWB for Oracle RAC" on page A-14

General Steps for Troubleshooting Warehouse Builder

Take the following steps to troubleshoot errors in Warehouse Builder:

1. Review this section for a possible solution to the problem.

If Warehouse Builder displays an error message during the installation process, then refer to "Error Messages Related to Installation" on page A-3. If you did not note the error number, you can review the "Log Files for Installation Errors" on page A-2.

In the absence of an error message, refer to "Troubleshooting Other Installation Problems" on page A-12.

2. Check for additional information about the problem by "Inspecting Log Files in Warehouse Builder" on page A-2.
3. If the problem remains unresolved, search for a possible solution at My Oracle Support at <https://support.oracle.com/>.
4. From the Repository Assistant, use the Check for Updates Wizard to see if there is a patch or information that addresses the problem. See "About the Check for Updates Wizard for Oracle Warehouse Builder" on page 6.
5. Review the Oracle Warehouse Builder Release Notes for installation notes or known issues.

6. If you are unable to resolve the problem in the previous steps, contact Oracle Support.

Oracle Support may ask you to complete the steps in "[Using OWB Client Logging to Control Logging and Trace Problems](#)" on page A-13.

Inspecting Log Files in Warehouse Builder

This section outlines all the different types of error messages that are logged by Warehouse Builder and how to access them.

Warehouse Builder logs the following types of errors:

- [Log Files for Installation Errors](#) on page A-2
- [Log Files for Metadata Import and Export Errors](#) on page A-2
- [Log File for Validation Errors](#) on page A-2
- [Log File for Generation Errors](#) on page A-3
- [Log Files for Deployment and Execution Errors](#) on page A-3
- [Log File for Name and Address Server Errors](#) on page A-3

Log Files for Installation Errors

When you run Oracle Universal Installer to install Warehouse Builder, the installation error logs are automatically stored in:

`C:\ProgramFiles\Oracle\Inventory\logs\installActions<timestamp>.log`

When you run the Warehouse Builder Repository Assistant, the workspace installation error logs are stored in:

`OWB_HOME\UnifiedRepos\log_timestamp.log`

See "[Error Messages Related to Installation](#)" on page A-3 for suggested actions for commonly encountered errors during installation.

Log Files for Metadata Import and Export Errors

Metadata Import: When you import a project or specific objects into your workspace using the Metadata Import Utility, Warehouse Builder records details of the import process in a log file. You can specify the name and location of this log file from the Metadata Import dialog box.

Metadata Export: When you export a Warehouse Builder project or specific objects using the Metadata Export Utility, Warehouse Builder records the details of the export in a log file. You can specify the name and location of this log file from the Metadata Export dialog box.

Log File for Validation Errors

In Warehouse Builder, you can validate all objects by selecting the objects from the console tree and then selecting **Validate** from the Object menu. After the validation is complete, the validation messages are displayed in the Validation Results window.

You can also validate mappings from the Mapping Editor by selecting **Mapping**, then **Validate**. The validation messages and errors are displayed in the Validation Results window.

On the Validation tab of the Validation Results window, double-click an object name in the Object column to display the editor for that object. You can fix errors in the editor.

Double-click a message in the Message column to display the detailed error message in a message editor window. To save the message to your local system, select **Code** in the menu bar, then select **Save as File**.

Warehouse Builder saves the last validation messages for each previously validated objects. You can access these messages at any time by selecting the object from the console tree in the Project Navigator, select **View** from the menu bar, and then click **Validation Messages**. The messages are displayed in the Validation Results window.

Log File for Generation Errors

After you generate scripts for Warehouse Builder objects, the Generation Results window displays the generation results and errors. Double-click an error under the Messages column on the Validation tab to display a message editor that enables you to save the errors to your local system.

After you generate scripts for Warehouse Builder objects, the Generation Results window displays the generation results and errors. Double-click an error under the Messages column on the Validation tab to display a message editor that enables you to save the errors to your local system.

Log Files for Deployment and Execution Errors

You can store execution or deployment error and warning message logs on your local system by specifying a location for them. In the Project Navigator, select the project. Then from the Tools menu, select **Preferences**. In the Preferences dialog box, click the Logging option in the object tree to the left. In the list box on the right, you can set the log file path, file name and maximum file size. You can also select the types of logs you want to store.

You can view this log of deployment and error messages from the Warehouse Builder console by selecting **View** from the menu bar, and then **Messages Log**. This Message Log dialog box is read-only.

Errors related to the Control Center Service are stored at the following path:

OWB_HOME\log\Repository_Name\log.xx on Oracle Database server.

Errors related to transforming or loading data are stored in the Control Center audit tables. You can access these error reports using the Repository Browser. The Browser provides detailed information about past deployments and executions. Click the Execution tab in the Execution reports to view error messages and audit details.

Log File for Name and Address Server Errors

If you are using the Name and Address cleansing service provided by Warehouse Builder, you can encounter related errors.

Name and address server start up and execution errors can be located at:

OWB_HOME\owb\bin\admin\NASver.log

If your Name and Address server is enabled in:

OWB_HOME\owb\bin\admin\NameAddr.properties:TraceLevel=1,

then it produces the log file *NASvrTrace.log*.

Error Messages Related to Installation

This section includes the following topics:

- No fonts were found in '<drive>:\Program Files\ Qarbon\viewlet Builder3jre\lib\fonts'
- OWBSYS is not granted access to OWB_HOME/owb/bin/admin/rrepos.properties: Please run UnifiedRepos/reset_owbcc_home.sql, specifying the path of Oracle home from which the Control Center Service is being run.
- SYS user does not have SYSDBA privileges.
- RTC-5301: The Control Center Service is not currently available.
- API5022: Cannot Connect to the Specified Repository
- Run-time Assistant fails with LoadJava Error.
- Error when specifying a SYSDBA user.
- Regional Name and Address Data Libraries Are Not Available.
- Lineage and impact analysis reports: Extensive tablespace requirements for materialized views.
- Java out of memory error occurs during a batch operation.
- ORA-01925: Maximum of 30 enabled roles exceeded
- INS0009: Unable to connect to the database. Verify the connect information.
- INS0022: A spawned program error.
- ORA-12154: TNS: Could not resolve service name.
- ORA-12514: TNS: listener could not resolve SERVICE_NAME given in connect descriptor.
- PL/SQL: ORA-04052: Error occurred when looking up remote object
- IMP-00003: ORACLE error 30371 encountered
- Unable to connect to SQL*Plus in <Oracle Database version>
- ORA-04020 deadlock detected while trying to lock object or ORA-04021 timeout occurred while waiting to lock object
- ORA-04088: error during execution of trigger 'DVSYS.DV_BEFORE_DDL_TRG'
- DPF-0029: Source <Table_Name> must have less than 165 attributes

Causes and Actions

No fonts were found in '<drive>:\Program Files\ Qarbon\viewlet Builder3jre\lib\fonts'

Cause: After installing Warehouse Builder client components, you installed another software program that relies on Jinitiator and overwrote Java objects necessary for Oracle products. This may prevent you from launching Warehouse Builder or any other Oracle product that depends on Java objects.

Action: Re-install Jinitiator.

OWBSYS is not granted access to OWB_HOME/owb/bin/admin/rrepos.properties: Please run UnifiedRepos/reset_owbcc_home.sql, specifying the path of Oracle home from which the Control Center Service is being run.

Cause: When running the script `reset_owbcc_home.sql` and prompted for the `OWB_HOME`, you typed an invalid path for `OWB_HOME`.

Action: Run the script again and enter the correct path.

On all platforms, including both Windows and Unix, the path you enter must use forward slashes, and is case-sensitive. The case of the path entered here must match exactly the case of the path for the Warehouse Builder home as known by the operating system.

On Unix, the correct path to enter is the path for the *OWB_HOME* directory. On Windows, to determine the correct path for the *OWB_HOME* directory, and examine the path displayed as part of the default Windows command prompt.

SYS user does not have SYSDBA privileges.

Cause: In a standard database installation, the SYS user has SYSDBA credentials and REMOTE_LOGIN_PASSWORDFILE=EXCLUSIVE. You can verify the credentials by issuing the following connect statement:

```
SQL> CONNECT sys@tns_name_of_db AS SYSDBA;
```

Enter password: sys_password

If your database is configured with REMOTE_LOGIN_PASSWORDFILE=NONE, then the statement fails.

Action: If the statement fails, then you have the following options:

- Reconfigure your database with
REMOTE_LOGIN_PASSWORDFILE=EXCLUSIVE and create a password file if none exists.
- If the preceding is not an option, reconfigure your database with
O7_DICTIONARY_ACCESSIBILITY=TRUE.

RTC-5301: The Control Center Service is not currently available.

Cause: A Control Center Service must be running to enable the Control Center to manage deployments and executions. The service connects to the Control Center using JDBC and can be run from any Warehouse Builder home. Normally the service is runs on the server host.

Action: You can start a service on the server host by using the script `start_service.sql`.

If it is not possible to run the service on the server host, then start the Control Center Service on the local computer using the script `local_service_login.sh` or `local_service_login.bat` as appropriate. Use this script as follows:

```
local_service_login.sh [-startup | -closedown] OWB_HOME
```

In this mode, the Control Center Service runs on the local computer and is available only when that computer is available and can connect to the Control Center.

Use the script `show_service.sql` to determine the status of the service.

Control center service log file reports "DBMS_OBFUSCATION" or "No key is found."

Cause: The encryption of the passwords is out of sync with the client.

Action: Reset the repository and restart the control center service. To reset the repository, run `owb/rtp/sql/reset_repository.sql`.

API5022: Cannot Connect to the Specified Repository

This error occurs when you try to connect to the a repository after having performed a database export or import from the Warehouse Builder repository schema.

Cause: The package NAMESPACESERVICEIMPL may be invalid. This occurs after a database export or import from the Warehouse Builder repository schema if the repository owner has no SELECT privilege on SYS.V_\$SESSION. You can diagnose the cause as follows:

1. In SQL*Plus, connect to the Warehouse Builder repository schema.
2. Enter the following command at the SQL prompt:

```
ALTER PACKAGE NAMESPACESERVICEIMPL compile body;
```

3. If **Warning: Package body altered with compilation errors** appears, enter the following command at the SQL prompt:

```
show errors;
```

4. The following errors mean that the Warehouse Builder repository owner has no SELECT privilege on SYS.V_\$SESSION.

```
PL/SQL: SQL statement ignored  
PLS-00201: Identifier 'SYS.V_$SESSION' must be declared
```

Action: Complete the following steps:

1. In SQL*Plus, connect as the SYS user.
2. At the SQL prompt, enter the following command:

```
grant SELECT on V_$SESSION to Warehouse_Builder_Repository_Owner;
```

3. Connect to the *Repository_Owner*.

4. Enter the following command at the SQL prompt:

```
alter package NAMESPACESERVICEIMPL compile;
```

Run-time Assistant fails with LoadJava Error.

Cause: This can occur if the Oracle Database does not have the JServer option installed.

Action: Make sure that the Oracle Database has JServer option installed.

Error when specifying a SYSDBA user.

Oracle Warehouse Builder Assistants require you to provide SYSDBA credentials when installing the Oracle Warehouse Builder Design Repository or run-time components.

Cause: In a standard database installation, the SYS user has SYSDBA credentials. You can verify this from SQL*Plus by issuing the following connect statement:

```
connect sys/sys_password@TNS_NAME_OF_DB as sysdba;
```

In a standard database installation, the preceding connect statement works because REMOTE_LOGIN_PASSWORDFILE=EXCLUSIVE and the default password file is created by the installation process.

If your database is configured with

REMOTE_LOGIN_PASSWORDFILE=NONE, then the following statement fails:

```
connect sys/sys_password@TNS_NAME_OF_DB as sysdba;
```

In this case, you have two options.

Action: Reconfigure your database with

REMOTE_LOGIN_PASSWORDFILE=EXCLUSIVE and create a password file if none exists.

Action: If the preceding is not an option, then reconfigure your database with O7_DICTIONARY_ACCESSIBILITY=TRUE. With this setting, the statement connect sys/*sys_password*@*TNS_NAME_OF_DB* enables the Warehouse Builder Assistants to connect to SYS user.

Regional Name and Address Data Libraries Are Not Available.

Cause: The Name and Address regional data libraries may not have been installed in the correct location.

Action: Ensure that you have successfully extracted regional data to the NAS_DATA directory.

1. From the *OWB_HOME*, start the Name and Address Server:

For Windows, run *owb\bin\win32\NAStart.bat*.

For Linux, Run *owb/bin/unix/NASTART.sh*.

2. Open the log file: *owb\bin\admin\NASvr.log*.

The log contains a list of installed countries.

If there is no such list, then verify that you have extracted the regional library data to the correct location. If you have extracted the data to the wrong location, then you can either reinstall the data, or modify the *owb\bin\admin\NameAddr.properties* file to indicate the correct file path. If you modify the *NameAddr.properties* file, then stop and restart the Name and Address Server as follows:

For Windows, start the server by running *owb\bin\win32\NAStart.bat*. Stop the server by running *owb\bin\win32\NAStop.bat*.

For UNIX, start the server by running *owb/bin/unix/NASTART.sh*. Stop the server by running *owb/bin/unix/NASTOP.sh*.

3. Once you have verified the installation, you can stop the Name and Address Server if you want, because it is automatically started at the execution of any mapping that employs the Name and Address operator.

Lineage and impact analysis reports: Extensive tablespace requirements for materialized views.

The first time you refresh a materialized view, it is populated from the Oracle Warehouse Builder repository. The materialized view can occupy up to twice the amount of space allocated to the entire Warehouse Builder repository.

Cause: Insufficient space has been allocated to the Warehouse Builder repository schema.

Action: If the Warehouse Builder repository schema is created in a tablespace that is dedicated to its use, these issues are easier to monitor. Ensure that sufficient free space exists on the physical drive for tablespace expansion. Within Oracle Enterprise Manager, ensure that the tablespace is set to Autoextend On.

Note: To grant permission to an OWB repository user to use Enterprise Manager for performing tasks, enter the following command in SQL*Plus: GRANT SELECT any dictionary to "&OWB repository user";

Java out of memory error occurs during a batch operation.

Operations requiring large amounts of memory can result in a Java out of memory error, if the system resources (such as virtual memory) are constrained.

Cause: There is not enough virtual memory. The Warehouse Builder client runs with a maximum heap size, as defined by the `-mx` parameter in the `owbclient.bat` file. The `-Dlimit` parameter in the `owbclient.bat` file specifies the memory threshold (80% of Dlimit) at which the Warehouse Builder memory manager begins to assist Java garbage collection. If you change the `-mx` parameter value, set the `-Dlimit` parameter to the same value, or at least to 90% of the value. Note that setting the `-Dlimit` to a low value can have a negative impact on the performance of Warehouse Builder.

Action: Increase the `-Dlimit` parameter in Warehouse Builder as follows:

1. Exit Warehouse Builder.
2. Open this file in a text editor:
For Windows, open the `$OWBHOME\bin\win32\ombplus.bat`.
For UNIX, open the `$OWBHOME\bin\win32\owbclient.sh`.
3. Change the `-Dlimit` parameter to 334.
4. Save and close the file.
5. Restart Warehouse Builder.

ORA-01925: Maximum of 30 enabled roles exceeded

This error occurs when installing a repository or a target schema.

Cause: The maximum number of enabled roles in the database has been exceeded. When you create a repository or a target schema, new roles are created in the database assigned to the schema in question. When the number of roles exceeds the value of the `MAX_ENABLED_ROLES` parameter, this error occurs.

Action: Increase the value of the `MAX_ENABLED_ROLES` parameter in the `init.ora` file. When you uninstall a repository or a target schema, delete the associated roles as well.

INS0009: Unable to connect to the database. Verify the connect information.

This error occurs when you try to connect to a database.

Cause: See the cause for [ORA-12514: TNS: listener could not resolve SERVICE_NAME given in connect descriptor](#).

Action: Follow the instructions for [ORA-12514: TNS: listener could not resolve SERVICE_NAME given in connect descriptor](#).

INS0022: A spawned program error.

Cause: This error message can result from a server issue when installing Warehouse Builder run-time components on an HP-UX operating system.

Action: To identify the server issue, complete the following steps:

1. From SQL*Plus, connect to a SYS user.

```
Create user test_lj identified by test_lj;
Grant connect, resource to test_lj;
```
2. Create `OWB_HOME/owb/bin/unix/test.sh` with the following contents:

```
../unix/loadjava -thin -verbose -order -resolve -user
'test_lj/test_lj@(DESCRIPTION=(ADDRESS=(PROTOCOL=TCP) (HOST=hpdgpa3) (PORT=15
```

```
22)) (CONNECT_DATA=(SERVICE_NAME=dgpadw)) )'
.../lib/int/rtpserver.jar
```

3. Change directory to *OWB_HOME/owb/bin/unix/*.
4. Run `test.sh`.

ORA-12154: TNS: Could not resolve service name.

This error occurs when you try to connect to a database.

Cause: You defined a Warehouse Builder location and specified connection information using SQL*Net. However, the required TNS name is not accessible.

Action: To set up a TNS name for use during deployment in general and for execution of mappings and process flows, the TNS name needs to be accessible from the *OWB_HOME* being used to run the control center service. To ensure access, run the Net Configuration Assistant from the *OWB_HOME* and then restart the control center service.

To set up a TNS name for use by database links, the TNS name needs to be accessible from the database server home. To ensure access, run the Net Configuration Assistant from the database server home.

Cause: See the cause for [ORA-12514: TNS: listener could not resolve SERVICE_NAME given in connect descriptor](#).

Action: Follow the instructions for [ORA-12514: TNS: listener could not resolve SERVICE_NAME given in connect descriptor](#).

ORA-12514: TNS: listener could not resolve SERVICE_NAME given in connect descriptor.

This error occurs when you try to connect to a database.

Cause: If you used Oracle Net Easy Configuration or Oracle Net Assistant to create the Net Service Name entry, and you used the default option (Service Name) on the newly created Net Service Name, then the parameter `SERVICE_NAME` is added to the `TNSNAMES.ORA` as a subclause to the `CONNECT_DATA` section in the Net Service Name entry. This replaces the `(SID=SIDname)` subclause in previous releases of the database, for example, Oracle Database8i (8.1.x).

Action: Implement the `TNSNAMES.ORA` file as follows:

1. Use the `GLOBAL_DBNAME` parameter in the `LISTENER.ORA` for each SID that you want to identify as a separate service. Use the value of this parameter as the value of the `SERVICE_NAME` parameter. You need to activate any changes that you make to `LISTENER.ORA` for this purpose by stopping and restarting the listener process.
2. Use the values of the parameters that exist in the `INIT.ORA`, namely `SERVICE_NAMES` and `DB_DOMAIN`, to determine the value of the `SERVICE_NAME` that you must use in `TNSNAMES.ORA`. The valid construction of this value is `SERVICE_NAMES.DB_DOMAIN` with the period separating the two `INIT.ORA` values. If your `SERVICE_NAMES` is `BIKES` and your `DB_DOMAIN` is `COM`, then your `SERVICE_NAME` is `BIKES.COM`.
3. If there is no `DB_DOMAIN` parameter set in your `INIT.ORA`, or if there is no `GLOBAL_DBNAME` in the `LISTENER.ORA`, then you can use the `SERVICE_NAMES` from the `INIT.ORA` in your `TNSNAMES.ORA` for the `SERVICE_NAME` parameter.

For example, if INIT.ORA contains SERVICE_NAMES = "TEST817" and db_domain is not set, then the TNSNAMES.ORA entry is: CONNECT_DATA =(SERVICE_NAME = "TEST817").

4. If you have multiple values specified in the SERVICE_NAMES parameter in the init.ora, then you can use one of them. If SERVICE_NAMES is not set, then you can use DB_NAME.DB_DOMAIN parameters from the INIT.ORA file.
5. If SERVICE_NAMES and DB_DOMAIN is not set in the INIT.ORA and there is no GLOBAL_DBNAME in the LISTENER.ORA, then your SERVICE_NAME in TNSNAMES.ORA file is DB_NAME.

PL/SQL: ORA-04052: Error occurred when looking up remote object

This error occurs when you have upgraded to <Oracle Database version> and are trying to redeploy mappings without first redeploying connectors.

Cause: While upgrading Oracle Database, you moved your database to a new computer. Your old and new database instances do not have the same domain name. You can verify the cause by logging in to SQL*Plus as a SYS user and entering the following command: SELECT * FROM GLOBAL_NAME; If the Global Name of the old database does not match that of the new database, then a domain mismatch is causing this error.

Action: Either add the domain name to the Global Name in your new database by issuing a command similar to the following statement: ALTER DATABASE RENAME GLOBAL_NAME TO xxx10G.US.ORACLE.COM; or redeploy your connectors.

Refer to *Oracle Warehouse Builder User's Guide* for information on deploying connectors.

IMP-00003: ORACLE error 30371 encountered

ORA-30371: column cannot define a level in more than one dimension

This error occurred when you were importing your target schema during migration.

Cause: The Warehouse Builder target schema is created with the select_catalog_role privilege. If you have the same dimension object defined in multiple Warehouse Builder target schemas, then Oracle Export creates duplicates in the export file, and this error occurs when you import.

Action: Connect as a SYS user to the existing version of the Oracle Database from which you exported the target schemas. Enter the following statement in

SQL*Plus: revoke select_catalog_role from OLD_Target_Schema;

Export the target schema into an Oracle .DMP file again, and then import the file into Oracle Database.

Unable to connect to SQL*Plus in <Oracle Database version>

Cause: Your Oracle home or Path is not set correctly, or your Net Service Names are not configured.

Action: Ensure your Oracle home and Path are set correctly, and your Net Service Names are configured in Oracle Database.

- Ensure that *ORACLE_HOME* and *PATH* are set correctly. Your Oracle home directory must point to the *OWB_HOME*. Set your *PATH* variable to include the *OWB_HOME\bin* directory before any other Oracle products.
- Ensure that the *TNSNames.ora* file is configured correctly:

For Windows, from Oracle Database program group, start **Net Configuration Assistant** and select **Local Net Service Name Configuration** to configure TNSNames.ora.

For UNIX, set ORACLE_HOME and PATH to the *OWB_HOME* for Warehouse Builder 11g Release 2 (11.2), then run *OWB_HOME/bin/netca* to start **Net Configuration Assistant**. Select **Local Net Service Name Configuration** to configure TNSNames.ora.

ORA-04020 deadlock detected while trying to lock object or ORA-04021 timeout occurred while waiting to lock object

When creating run-time objects, the Run-Time Assistant halts and produces these errors in the error log when trying to lock sys.dbms_aq.

Cause: User sessions may be pinning Advanced Queue objects.

Action: First, log into SQL*Plus as a SYS user and run a query to identify which user sessions are pinning the Advanced Queue packages, using the following query as an example:

```
column s.sid format a5;
column s.serial# format a8;
column s.username format a10;
column objectname format a10;
select distinct
  s.sid,
  s.serial#,
  s.username,
  x.kglnaobj as objectname
from
  dba_kgllock l,
  v$session s,
  x$kgllk x
where
  l.kgllktype = 'Pin' and
  s.saddr = l.kgllkuse and
  s.saddr = x.kgllkuse and
  x.kglinaobj in ('DBMS_AQ', 'DBMS_AQADM');
```

The following is an example of the output you may receive:

SID	SERIAL#	USERNAME	OBJECTNAME
---	-----	-----	-----
9	29623	RTU_4942	DBMS_AQ

Noting the SID and Serial Number, issue the following command to kill the user sessions:

```
ALTER SYSTEM KILL SESSION 'SIDNoted, SerialNumberNoted';
```

For example, enter the following command to kill the session listed in the sample output for this error:

```
ALTER SYSTEM KILL SESSION '9,29623';
```

ORA-04088: error during execution of trigger 'DVSYS.DV_BEFORE_DDL_TRG'

Cause: When you attempt to create a Warehouse Builder repository on an Oracle Database that includes the Database Vault option, you may encounter an error such as ORA-04088.

Action: Disable the triggers DV_BEFORE_DDL_TRG and DV_AFTER_DDL_TRG.

Import wizard throws error ORA-00997 when importing a table.

Cause: When you import table definitions from an Oracle Database, you may encounter an error such as "Repository Error Message: ORA-00997: illegal use of LONG datatype...". This occurs when the CURSOR_SHARING parameter is set to FORCE or SIMILAR.

Action: Set the database parameter CURSOR_SHARING to EXACT.

DPF-0029: Source <Table_Name> must have less than 165 attributes

Cause: Creating data profiling on a table having more than 165 columns.

Action: Select a subset of columns from the table for profiling, by defining an attribute set. This is a data profiling restriction.

Troubleshooting Other Installation Problems

This section includes causes and actions for the following installation problems:

- [Warehouse Builder Clients that Previously Launched Now Momentarily Display the Splash Screen and Fail to Start](#)
- [Newly Installed Warehouse Builder Clients Fail to Start and Previously Launched Oracle Products Fail to Start](#)
- [A Warehouse Builder Client Freezes or Hangs](#)
- [Defining a User and an Owner in the Basic Installation in release 10.2](#)

Causes and Actions

Warehouse Builder Clients that Previously Launched Now Momentarily Display the Splash Screen and Fail to Start

Cause: If you attempt to start a Warehouse Builder client such as the Design Center and the splash screen displays momentarily but the client fails to start, you may have overwritten required java objects during the subsequent installation of another software product.

If the client is installed on Windows and you launched the client from the Start menu, you may not see any error messages.

Action: Manually start the client by typing at the DOS prompt `run OWB_HOME\owb\owbclient.bat`. You are likely to encounter an error message such as `No fonts were found in '<drive>:\Program Files\ Qarbon\viewlet Builder3jre\lib\fonts'` on page A-4.

Newly Installed Warehouse Builder Clients Fail to Start and Previously Launched Oracle Products Fail to Start

Cause: After installing Warehouse Builder software, an error in the path variable can prevent you from launching Warehouse Builder clients and other Oracle products that previously launched without problems.

Action: Verify the that the path for `OWB_HOME\bin` is listed correctly in the Environmental Variables.

A Warehouse Builder Client Freezes or Hangs

Cause: Client software may freeze or hang due to various causes.

Action: If a Warehouse Builder client appears to freeze or hang, perform a stack trace as follows:

1. At the DOS command prompt, enter:

```
cd OWB_HOME\owb\bin\win32\
```

2. Run `owbclient.bat`.
3. When the program hangs, press **Ctrl+Break** keys.

This produces the thread-dump. Contact Oracle Support and provide them with this information to help identify the problem.

Defining a User and an Owner in the Basic Installation in release 10.2

Cause: If you are upgrading from release 10.2 to 11.2, then you may expect a default repository owner to be assigned as OWBRT_SYS. This is incorrect. This owner is not created.

Action: No action needed.

Checking the Java Virtual Machine (JVM)

To check, verify, or reinstall the Java Virtual Machine (JVM) server in the database, refer to My Oracle Support:

1. In your Web browser, go to the following URL:
<https://support.oracle.com>.
2. Log into My Oracle Support, or register as a new user.
3. Enter the following terms into the Search field, separating each term by semicolons):
INITJVM.SQL; INSTALL; JAVAVM; JVM; VERIFY; SERVER; INSTALL;
CLEANUP
4. Press **Enter**.

This search returns the support notes for the JVM. The number of available documents frequently changes because My Oracle Support creates, merges, and deletes various support notes. This string of search words returns the most current and pertinent documents.

Using OWB Client Logging to Control Logging and Trace Problems

If Warehouse Builder is producing errors or exhibiting other unexpected results, additional error logging can help you and Oracle Support identify the cause. You can create a temporary file for logging by using `owbclient.logging.properties` in the OWB user home directory to get trace information from the client. This should only be used for specific troubleshooting as this will generate excessive debugging information if left on the system. The following example writes a logging file `owbclient.log` to `/tmp`. Using Java logging properties, enter:

```
console.messageFormat={1,time}: {2}
file.messageFormat=[{3} ({4})] {1,time}: <{0}> {2}
handlers=java.util.logging.FileHandler
java.util.logging.FileHandler.pattern=/tmp/owbclient.log
level = DEBUG_ALL
java.util.logging.FileHandler.level=ALL
java.util.logging.ConsoleHandler.level=DEBUG_ALL
```

Stop and restart the client for this to take effect. Remember to remove this after you have your log, and then stop and restart the client again.

Checking the System Path of Oracle Warehouse Builder

You must ensure that Oracle Universal Installer added the bin directory of the new Oracle Warehouse Builder 11.2 installation to the system path *ahead* of other Oracle product bin directories. This is easily checked using the command prompt.

To verify the system path for Oracle Warehouse Builder on Windows:

1. In the Command Prompt window, enter path.

```
C:\>path
```

2. The system returns the value of the path variable; verify that C:\OWB112 precedes other Oracle products.

```
PATH=C:\OWB112\bin;C:\oracle\product\11.2.0\db_1\bin; ... \
```

To verify the system path for Oracle Warehouse Builder on Linux:

1. Echo the PATH variable in your shell.

Configuring OWB for Oracle RAC

When installing Oracle Warehouse Builder on an Oracle RAC cluster, the Oracle RAC service names must be unique and match the node that they're running on. This ensures that the Control Center Service uses a database service on the correct node, which in turn ensures access to the correct file system.

For example, the wb_rt_service_nodes should be similar to those in [Table A-1](#).

Table A-1 Naming Oracle RAC Service Nodes

Node	Instance	Host	Port	Service Name
1	1	host_1	port_number_1	RACSRVC_1
2	2	host_2	port_number_2	RACSRVC_2
3	3	host_3	port_number_3	RACSRVC_3

Index

A

access
 multiple-user access, about, 13-16
 read-only mode, 13-17
 read/write mode, 13-17
accessing
 Metadata Loader, 12-19
adding
 workspace user, 3-12, 4-11
adding role, 7-7
Administrator role, 13-18
application name, 7-13
architecture, 1-1
assigning
 roles, 13-13
 roles to users, 13-21
 security roles, 13-17
 system privileges, 13-15
audit connection pool, 7-18
audit data source
 configuring, 7-3
audit table data source
 configuring, 7-21
audit table logging, 7-18

C

cat_owb.sql, 3-2, 4-2
changing
 Control Center Agent settings with
 JRTAdministratorMBean, 7-25
 metadata, 11-2
 passwords, 13-25
 session timeout, 7-1
Check for Updates Wizard, 1-6
clean_owbsys.sql, 3-2, 4-2, 5-10
cleaning repository
 clean_owbsys.sql, 3-3, 4-3
 Linux, 4-3
 Windows, 3-3
Cluster Topology, 7-4
Code Template, 7-2
code templates
 installing, 7-26
components, 1-1

configuration architecture, 1-1
configuring
 audit data source, 7-3
 audit table data source, 7-21
 Control Center Agent security roles, 7-4
 Control Center Agent server-side logging, 7-3
 Control Center Agent web service
 (jrt-web.war), 7-16
 Control Center service
 Linux, 4-4
 reset_owbcc_home.sql, 3-4, 4-4
 Windows, 3-4
 on Linux, 4-1
 on Windows, 3-1
OWB installed as a standalone
 Linux, 4-2
 Windows, 3-2
OWB installed with Oracle Database 11g Release 2
 Linux, 4-1
 Windows, 3-1
owbclient.bat file for memory, A-8
OWBSYS audit connection pool, 7-18
repository browser environment, 7-1
connecting
 database, errors connecting to, A-8
 Design Repository, error, A-5
connection
 testing, 7-20
connection pool type, 7-19
connection pools, 7-18
context root, 7-13
Control Center
 enabled roles, A-8
Control Center Agent, 1-2
 configuring server-side logging, 7-3
 deploying to OC4J, 7-12
Control Center Agent audit table logging
 enabling, 7-18
Control Center Agent files, 7-24
Control Center Agent runtime log file, 7-24
Control Center Agent security roles, 7-4
 configuring, 7-4
Control Center Agent settings, 7-25
Control Center Agent settings with
 JRTAdministratorMBean
 changing, 7-25

Control Center Agent topology, 1-5
Control Center Agent web service
 configuring, 7-16
Control Center Manager, 1-2
Control Center Service, 1-2
Control Center service
 configuring
 Linux, 4-4
 Windows, 3-4
 reset_owbcc_home.sql, 3-4, 4-4
stopping
 Linux, 4-3
 Windows, 3-3
creating
 database user, 3-12, 4-11
 database users, 13-8
 JMS queue, 7-9
 repository workspace
 Linux, 3-6
 Windows, 4-5
 workspace, 3-7, 4-7
creating OWBSYS schema
 cat_owb.sql, 3-4, 4-4
 Linux, 4-3
 Windows, 3-3
ct_permissions.properties, 7-2

D

data libraries
 Name and Address, A-7
data source type, 7-22
Data Sources, 7-22
database roles, changing, 13-12
database user
 creating, 3-12, 4-11
database users
 creating in Warehouse Builder, 13-8
 registering as Warehouse Builder users, 13-4
data-sources.xml, 7-3
default object privileges, 13-13
defining
 security roles, 13-15
deleting
 repository, 5-10
 Repository users, 5-2
 schema objects, 5-9
 workspace, 5-6
 workspace owner, 5-6
 workspace user, 5-2
deploying
 Control Center Agent to OC4J, 7-12
 deployment errors, A-2
deployment plan, 7-13
Design Center, 1-2
Design Control Center
 enabled roles, A-8
Design Repository
 connecting error, A-5
destination name, 7-10

destination type, 7-10
-Dlimit parameter
 owbclient.bat file, memory threshold, A-8
dropping
 workspace, 5-7

E

E-Business Suite, 7-26
editing user profiles, 13-12
enabling
 access to EBS, 7-27
 access to metadata, 7-27
 Control Center Agent audit table logging, 7-18
encrypting passwords, 13-25
enforcing security, 13-23
Enterprise Manager, 5-9
error logs
 interpreting error logs, A-2
ETL mappings, 1-5
execution
 errors, A-2
exporting
 metadata, for user grants, 12-11
 metadata, for user-defined definitions, 12-11
 metadata, in additional languages, 12-11
 metadata, results, 12-19
 metadata, using the Design Center, 12-1
snapshots, 11-17

F

FND_APPLICATION table, 7-27
FND_APPLICATION_VL table, 7-27
FND_COLUMNS table, 7-27
FND_FOREIGN_KEY_COLUMNS table, 7-27
FND_FOREIGN_KEYS table, 7-27
FND_PRIMARY_KEY_COLUMNS table, 7-27
FND_PRIMARY_KEYS table, 7-27
FND_SEQUENCES table, 7-27
FND_TABLES table, 7-27
FND_VIEWS table, 7-27

G

generation
 errors, A-2
GLOBAL_DBNAME parameter
 Net Service Name, A-9

H

host name, 5-2
 port number, 3-6, 4-6

I

iIntegrating
 E-Business Suite, 7-26
import language options, 12-6
importing

metadata, combining import modes and matching criteria, 12-5
metadata, for security grants, 12-15
metadata, for user-defined definitions, 12-15
metadata, in additional languages, 12-14
metadata, language options, 12-6
metadata, results, 12-19
metadata, using the Design Center, 12-3
metadata, validation rules, 12-6
snapshots, 11-17
init.ora file
 MAX_ENABLED_ROLES parameter, A-8
install option, 7-28
installing Warehouse Builder
 on RAC environment, 2-7
installation
 errors, A-2
 troubleshooting, A-1
installation architecture, 1-1
installation configuration
 choosing, 1-1
 Control Center Agent topology, 1-5
 multiple database topology, 1-3
 separate client and database server topology, 1-3
 separate Client, Control Center Service, and Database Server topology, 1-4
 single computer topology, 1-2
installing
 code templates, 7-26
 earlier database releases, 2-2
 JDBC drivers for OC4J, 7-23
 JServer option, Oracle Database, A-6
 on Oracle Database 11g Release 2, 2-1
 on server, 2-1
 optional components, 7-1
 Oracle Universal Installer, 2-3
 Oracle Workflow, 7-28
 Repository Browser, 7-1
 standalone software, 2-2
 third-party name and address data, 7-27
 Warehouse Builder, 2-9
installing Warehouse Builder
 on RAC environment, 2-7
Instance Level Security, 7-5
instance level security, 7-8
integrating
 third-party name and address data libraries, 7-27

J

j2ee-logging.xml, 7-3
Java Virtual Machine (JVM)
 verifying and reinstalling JVM, A-13
jazn.com, 7-6, 7-8, 7-9
JDBC drivers for OC4J
 installing, 7-23
JDBC Resources, 7-22
JDBC resources, 7-18
JDBC URL, 7-19
JMS destinations, 7-9

JMS queue
 creating, 7-9
JNDI location, 7-10, 7-23
JobID file, 7-24
JRT
 owb_abortQueue, 7-9
 owbQueue, 7-9
JRTAdministrator, 7-25
JRTAdministratorMBean, 7-25
jrt-internal-log-handler, 7-3
JRTWebService, 7-16, 7-18
jrt-web.war, 7-16
jrtws.war, 7-18
JServer option
 Repository Assistant, for the, A-6
jython_lib.zip, 7-2
jython_owlolib.jar, 7-2

K

km_security_needed, 7-2

L

language
 select, 3-11, 4-10
LD_LIBRARY_PATH, 7-28
LDAP parameters, 7-29
Level Security, 7-6
licence agreement, 2-2
listener.ora file
 GLOBAL_DBNAME parameter, A-9
LoadJava error
 Repository Assistant, A-6
log files
 Metadata Loader logs, 12-18
logging
 audit table, 7-18
logs
 interpreting error logs, A-2

M

mailer parameters, 7-29
managing
 workspace, 3-7, 4-7
 workspace user, 5-3
materialized views, insufficient table space, A-7
MAX_ENABLED_ROLES parameter, A-8
MaxPermSize, 7-2
MDL
 see Metadata Loader
memory
 errors during batch operation, A-8
metadata, 7-1
 changing, 11-2
full security strategy, 13-4
import and export errors, A-2
minimal security strategy, 13-3
multiuser security strategy, 13-3
security, 13-1 to 13-26

upgrading, 12-7
Metadata Export Utility, 12-17
metadata import
 required access privileges, 12-3
Metadata Import Utility, 12-17
Metadata Loader
 about, 12-19
 accessing, 12-19
 log files, 12-18
 multiple user access, 12-18
 results, 12-19
 utilities, 12-16
metadata objects
 applying security properties, 13-22
metadata repository, 3-3, 4-3
metadata snapshots, 11-2 to ??
modes
 read-only mode, 13-17
 read/write mode, 13-17
modifying
 OC4J startup script, 7-2
multiple database topology, 1-3
multiple user access with MDL, 12-18
multiple user ecurity strategy, 13-3
multiple-user access
 about, 13-16
 read-only mode, 13-17
 read/write mode, 13-17
-mx parameter
 owbclient.bat file, virtual memory, A-8
My Oracle Support, 2-5

N

Name and Address
 regional libraries unavailable, A-7
Name and Address server
 errors, A-2
names and addresses
 installing third-party data, 7-27
Net Service Name
 SERVICE_NAME parameter, A-9

O

object privileges, default, 13-13
objects
 restoring from snapshots, 11-15
OC4J startup script
 modifying, 7-2
oc4j_app_administrators, 7-8
oc4jadmin, 7-8
oc4j-administrators, 7-7
ODL job log files, 7-24
opmn.xml, 7-2
optional components
 installing, 7-1
Oracle Database
 JServer option, A-6
Oracle service name, 3-6, 4-6, 5-2

Oracle Workflow
 installing, 7-28
ORACLE_HOME, 7-28
ORACLE_PORT, 7-28
ORACLE_SID, 7-28
OWB Browser, 1-2
OWB Client, 1-2
OWB software
 uninstalling, 5-10
OWB_INTERNAL_USERS, 7-9
OWB_J2EE_ADMINISTRATOR, 7-8
OWB_J2EE_EXECUTOR, 7-7, 7-8
OWB_J2EE_OPERATOR, 7-7, 7-8
OWB_USER, 7-29
owbclient.bat file
 -Dlimit memory threshold parameter, A-8
 -mx virtual memory parameter, A-8
OWBSYS
 creating, 3-3
 definition, 2-1
 unlocking, Linux, 4-1
 unlocking, Windows, 3-1
OWBSYS account, 2-1
OWBSYS audit connection pool
 enabling, 7-18
OWBSYS schema
 creating
 Linux, 4-3
 Windows, 3-3
OWBSYS_AUDIT
 definition, 2-1
 unlocking, Linux, 4-1
 unlocking, Windows, 3-1
owbsys_audit, 7-19
OWBSYS_AUDIT account, 2-1

P

passwords
 changing, 13-25
 encrypting, 13-25
 in Warehouse Builder, 13-24
PATH environment variable, 7-11
performance
 -Dlimit parameter, A-8
performing a stack trace, A-12
PermGen, 7-2
persistence, 7-10
port number, 5-2
pre-installation checklist, 2-9
privileges
 default, 13-13
 object, 13-13
 SYS user, checking privileges, A-5
 system, 13-15
Process Flow Editor, 7-29
profiles, role, 13-19
properties
 security, 13-22
Python.path, 7-2

R

RAC environments, installing Warehouse Builder
on, 2-7
read-only mode, 13-17
read/write mode, 13-17
Realms, 7-6
remote installation
 OWB home, 3-10, 4-9
 setting OWB home, 3-5, 4-5
remote_owb_install.sql, 3-2, 4-2
Repository, 1-2, 4-3
repository
 cleaning
 Linux, 4-3
 Windows, 3-3
 deleting, 5-10
Repository Assistant, 3-3, 3-6, 4-3, 4-5
 LoadJava error, A-6
Repository Assistant, Check for Updates Wizard
 in, 1-6
Repository Browser
 installing, 7-1
repository browser environment
 configuring, 7-1
repository objects, 3-3, 4-3
Repository users, deleting, 5-2
required access privileges
 metadata import, 12-3
requirements
 JServer option, Oracle Database, A-6
reset_owbcc_home.sql, 3-2, 4-2
role
 adding, 7-7
Roles, 7-6
roles
 Administrator, 13-18
 assigning, 13-13
 assigning to users, 13-21
 available, 7-7
 changing default, 13-12
 Everyone, 13-18
 profiles, 13-19
 security, 13-15, 13-17
 selected, 7-7

S

schema objects
 deleting, 5-9
schema objects, deleting, 5-9
secure website, 7-17
secure-web-site.xml, 7-17, 7-18
security
 enforcement, 13-23
 metadata, 13-1 to 13-26
 multiple-user access, 13-16
 read-only mode, 13-17
 read/write mode, 13-17
 roles, 13-15
security properties, 13-22

Security Providers, 7-5
security roles, 13-17
 Control Center Agent, 7-4
security strategy
 full, 13-4
 minimal, 13-3
 multiuser, 13-3
select
 languages, 3-11, 4-10
separate client and database server topology, 1-3
separate Client, Control Center Service, and Database
 Server topology, 1-4
service names
 errors resolving service name, A-9
session timeout
 changing, 7-1
session-config, 7-1
setting
 PATH environment variable, 7-11
setting OWB home
 remote installation
 Linux, 4-5
 Windows, 3-5
 remote_owb_install.sql, 3-6, 4-5
silent install response values, 2-11
single computer topology, 1-2
snapshots
 access privileges, 11-10
 adding components, 11-7
 comparing, 11-11
 converting type, 11-14
 deleting, 11-20
 exporting, importing, 11-17
 metadata, 11-2 to ??
 restoring, 11-15
 uses, 11-2
standalone software
 download, 2-2
SYS password, 7-28
SYS user
 verifying privileges, A-5
SYSDBA privileges
 checking for SYS user, A-5
system privileges, 13-15

T

tablespace, 7-29
 select, 3-10, 4-10
tablespace, and materialized views, A-7
tablespaces
 materialized views, allocation for, A-7
 setting for Control Center, A-7
Target Schema
 enabled roles, A-8
testing
 connection, 7-20
 web service, 7-16
third-party name and address data
 installing, 7-27

third-party name and address data libraries
 integrating with, 7-27
TNS connect descriptor, 7-29
tnsnames.ora file
 SERVICE_NAME parameter, A-9
troubleshooting
 installations, A-1

U

uninstalling
 OWB software, 5-10
uninstalling OWB, 5-1
unlocking OWBSYS, 3-5, 4-4
unlocking OWBSYS_AUDIT, 3-5, 4-4
unregistering
 workspace user, 5-4
user profiles, editing, 13-12
users
 assigning roles to, 13-21
 creating in Warehouse Builder, 13-8
 registering as Warehouse Builder users, 13-4
USERS tablespace, 3-3, 4-3
users, multiple
 security strategy, 13-3
using
 web service with a secure website, 7-17
utilities
 Metadata Loader, 12-16

V

validation
 errors, A-2
validation rules for metadata import, 12-6
virtual memory
 errors, A-8

W

Warehouse Builder
 frozen, A-12
 hanging, A-12
 installing, 2-9
 passwords, 13-24
web service
 testing, 7-16
web service with secure website, 7-17
web.xml, 7-1
wfinstall.bat, 7-28
wfinstall.csh, 7-28
wf.log, 7-29
workflow account, 7-28
Workflow Client, 7-29
Workflow Configuration Assistant, 7-29
workflow password, 7-28
Workflow Proxy User, 7-29
workspace, 3-3, 4-3
 creating, 3-7, 4-7
 Linux, 4-5
 Windows, 3-6

deleting, 5-6
dropping, 5-7
managing, 3-7, 4-7
workspace owner
 deleting, 5-6
workspace user
 adding, 3-12, 4-11
 deleting, 5-2
 managing, 5-3
 unregistering, 5-4