

An Oracle White Paper
December 2010
(supersedes previous versions)

Oracle Warehouse Builder 11.2: Upgrade and Migration Paths

Introduction	1
OWB 11gR2: Upgrades and Migrations from Previous Releases	2
OWB 11gR2 Releases and Recommended Patches	2
Understanding Migration vs. Repository Upgrade	3
Choosing Between Migration and Repository Upgrade	3
Migrating Objects as MDL from OWB 9.2, 10.x or 11.1	4
Migration Path for MDL Objects and Projects from Older Releases ...	5
Use the Latest Patch Releases for Intermediate Upgrades	5
Repository Upgrades from OWB 10.2/11.1 to OWB 11.2	5
Repository Upgrade to OWB 11.2 through MDL Files	7
In-Place Repository Upgrade to OWB 11.2	8
Repository Upgrade Paths for OWB 10.1 or 9.x to 11.2	9
OWB 10gR2 as an Intermediate Step in Repository Upgrade	9
OWB 9.x/10.1 Repository Upgrade Path Summary	9
Upgrading Repository from OWB 10.1.0.4 or Later to OWB 10.2.	10
Upgrading Repository from OWB 9 to OWB 10.1.0.4	10
Fixing Locations After Upgrade or Migration	10
Design Objects Modified During Migration or Upgrade	11
Changes to OWB 10.2 and 11.1 Objects	11
Changes to OWB 9.2 and 10.1 Objects	11
Running with an Upgraded Repository Without Regenerating	12

Introduction

This paper provides an overview of upgrade and migration paths to move existing designs and runtime metadata from older releases of Oracle Warehouse Builder (OWB) to OWB 11.2.

Previous discussions of OWB migration and upgrade have used the terms “upgrade” and “migrate” almost interchangeably, which has produced some confusion. This paper clarifies the distinction between the two, and explains the different methods by which customers with previous OWB projects can move their existing work to the 11gR2 version.

In some cases, intermediate upgrades or migrations may be required; this document identifies those cases and explains the required sequence.

Finally, the paper makes recommendations around reducing the risk associated with upgrading production systems without regenerating and redeploying code.

Links to specific content in documentation for different releases are also provided. The documentation contains the most detailed information about specific tasks, but the overall migration paths from different releases are best described here.

OWB 11gR2: Upgrades and Migrations from Previous Releases

All Oracle Warehouse Builder customers who are upgrading to the 11gR2 database will have to move to Warehouse Builder 11gR2, because Warehouse Builder 11gR2 is the only OWB release certified to work with Oracle Database 11gR2. Customers who are not immediately moving to the 11gR2 database should also consider upgrading to OWB 11gR2 as soon as possible.

- OWB 11gR2 offers updated, more flexible data integration capabilities: code template-based data integration, real-time DI and Web Services support.
- OWB 11gR2 is the only release of OWB that is certified to run on database 11gR2.
- Adopting OWB 11gR2, pre-installed with every 11gR2 Oracle database, is the best way to take advantage of the latest in OWB and Oracle database features.
- OWB 11gR2, released with the 11gR2 database, will be supported for as long as the 11gR2 database is supported.
- OWB 11gR2 offers a greatly improved user interface that exposes familiar and new features in a user interface built on the JDeveloper framework used in other Oracle developer tools. Customers on any supported database version can benefit from these features.

There are, however, different methods of bringing existing OWB solutions to 11gR2, appropriate for different situations. This paper presents:

- An overview of different migration and upgrade paths
- A guide to choosing a specific migration or upgrade scenario
- The general steps required to perform each available migration or upgrade process
- Changes to objects in projects from OWB 10.1 or earlier during the upgrade process, to map those objects on to the closest equivalents in OWB 11gR2

Note that there may be specific issues on upgrade and migration that arise after this paper is published. There are specific one-off patches that should be applied to OWB environments before an upgrade or migration. As of this writing, the recommended upgrade is to OWB 11.2.0.2. Contact Oracle Support for the most up-to-date information on recommended OWB patch levels for upgrades and migrations.

OWB 11gR2 Releases and Recommended Patches

Before upgrading or migrating repositories, it is important that you apply appropriate patches to resolve any outstanding issues.

At this writing (December 2010), the recommended patch level for all customers is OWB 11.2.0.2. On top of OWB 11.2.0.2, apply cumulative patch 10185523 to resolve remaining issues, including upgrade-related issues.

Customers using OWB 11.2.0.1 must apply cumulative patch 10270220 to avoid repository upgrade and migration-related issues.

It is recommended that you contact Oracle Support before beginning the repository upgrade or migration process to ensure that these are the latest patches.

Understanding Migration vs. Repository Upgrade

Previous discussions of OWB migration and upgrade, including the 11gR2 product documentation, have used the terms “upgrade” and “migration” almost interchangeably, which has produced some confusion. (In other contexts, “migration” can also refer to moving OWB objects from one instance to another, such as from development to test to production, or it can refer to moving to OWB from another ETL product altogether.)

For the purposes of this paper, the following definitions are used:

- Upgrading a repository is the process of taking an entire repository, including the control center audit data such as deployment and execution history, from a previous OWB release and moving it to OWB 11gR2. This process is performed from within Repository Assistant.
- Migration, in the context of moving between OWB versions, is the process of exporting existing OWB design-time objects from an earlier OWB version, and importing the resulting MDL file into OWB 11gR2. This process is performed from OWB Design Center.

Depending upon whether you plan upgrade or migration, different paths are supported from one release to the next. In some cases, intermediate upgrades or migrations may be required; this document identifies those cases and explains the required sequence.

Choosing Between Migration and Repository Upgrade

OWB 11gR2 can upgrade OWB 10gR2 and OWB 11gR1 repositories directly, using functionality in the OWB 11gR2 Repository Assistant. The advantages of upgrading an entire repository are:

- You can move up to OWB 11gR2 without regenerating code or redeploying your mappings, for maximum stability, then decide on objects to regenerate or redeploy based on whether you need specific 11gR2 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:

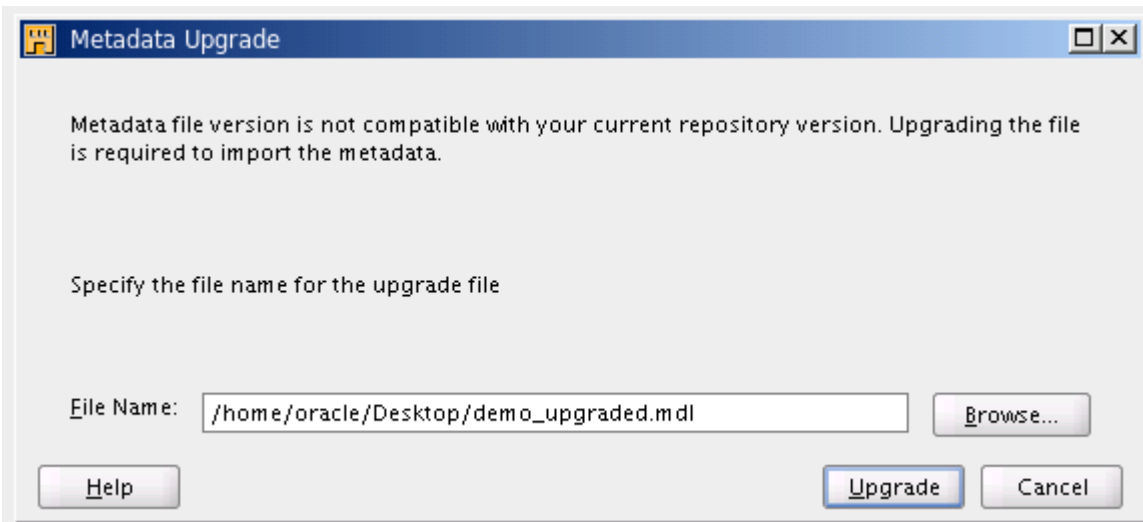
- Speed, if you have large volumes of runtime metadata that you do not need to preserve.
- Simplicity, in cases where the upgrade path for the complete repository is more complex than the migration path. In particular, OWB 11gR2 can directly import MDL for projects from OWB 9.2 and 10.1, but cannot upgrade 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.

Migrating Objects as MDL from OWB 9.2, 10.x or 11.1

OWB 11gR2 can directly import MDL for projects or other objects from OWB 9.2.0.x or higher. Importing metadata from MDL files from previous releases is performed in the OWB 11gR2 Design Center. The process is described at length in Oracle Warehouse Builder Installation and Administration Guide, chapter 12, “Using the Metadata Loader.”

The process is like importing any other MDL file; for example, select **File** → **Import** → **Warehouse Builder Metadata...** and select the file to import.

OWB detects the earlier MDL version and displays the Metadata Upgrade dialog box:



Use the File Name field to specify an output file name. OWB writes an upgraded version of your MDL file to that location, then imports that new MDL into your repository. You can save that upgraded MDL file for use in future imports.

Migration Path for MDL Objects and Projects from Older Releases

For releases older than 9.2.0.x, the supported migration path is:

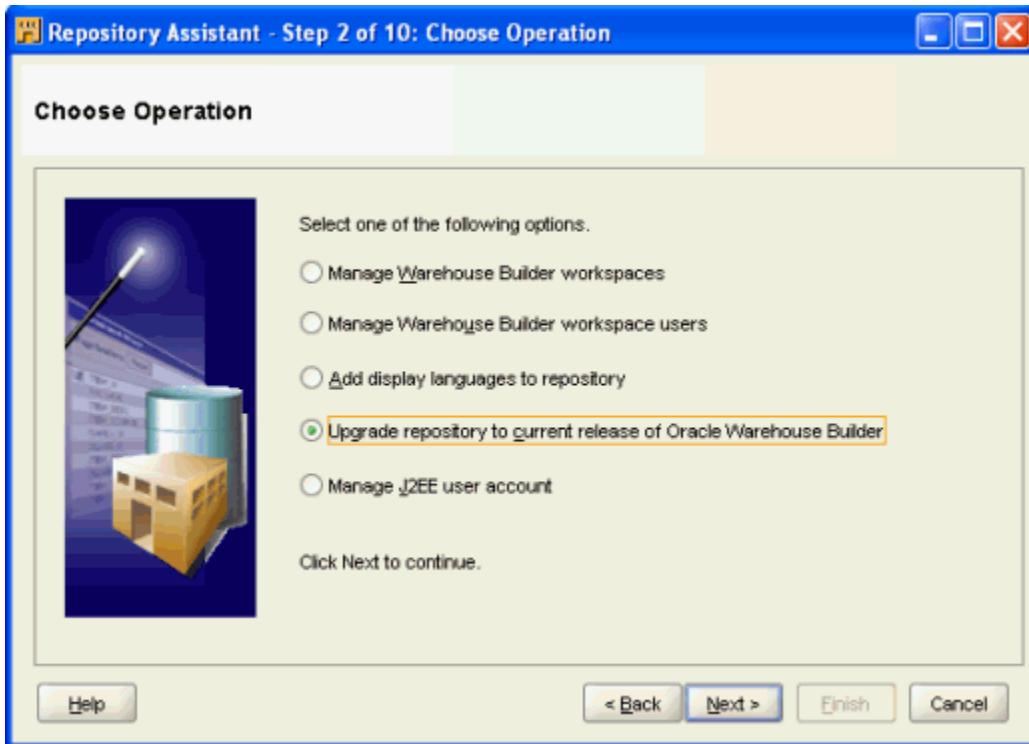
- Import the MDL to an intermediate 10.2.0.5 or 11.1.0.7 database
The MDL file upgrade process generates an upgraded MDL file; you will be prompted for a filename. The upgraded MDL file is imported into your intermediate database, which you can ignore.
- Import that upgraded MDL to 11.2.

Use the Latest Patch Releases for Intermediate Upgrades

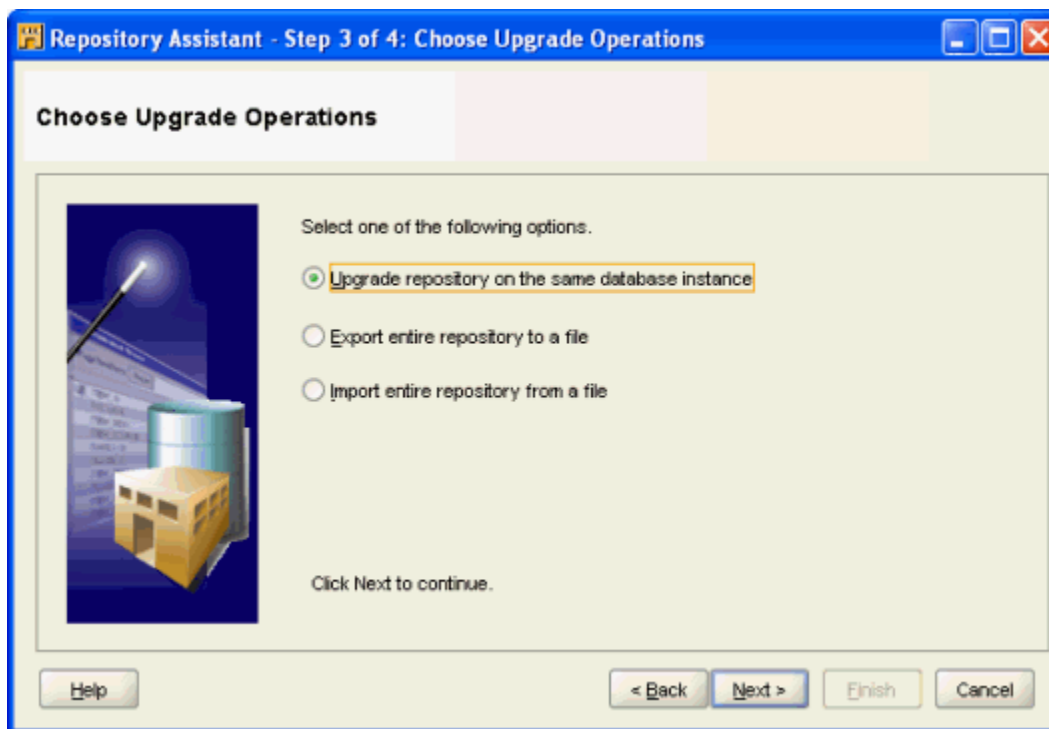
Note: The latest 10.1, 10.2 or 11.1 patch release is strongly recommended for cases where an intermediate release is used in the upgrade path. In some cases, earlier patch releases may not import or upgrade metadata from all earlier releases of the product.

Repository Upgrades from OWB 10.2/11.1 to OWB 11.2

Upgrading repositories is performed from the OWB 11.2 Repository Assistant. From the “Choose Operation” dialog box in Repository Assistant, you can access upgrade operations by selecting “Upgrade repository to current release of Oracle Warehouse Builder”.



The upgrade operations are presented in the next screen:



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

- Exporting a complete 10.2 or 11.1 repository to an MDL file, which can then be imported on the same or a different database; (sometimes called migration through MDL files);
- Upgrading the repository on the same database instance (sometimes called an in-place repository upgrade).

The discussion here provides a basis for choosing between them. Full details of using the OWB 11.2 Repository Assistant and pre-upgrade and post-upgrade steps required are described in *Oracle Warehouse Builder Installation and Administration Guide 11g Release 2*, Chapter 6, “Migrating Oracle Warehouse Builder from Earlier Releases.”

Repository Upgrade to OWB 11.2 through MDL Files

Migration through MDL files is generally used when moving your entire repository to a new database. (Exporting an entire repository as MDL can also be a way to back it up.) Note that the MDL files created in this process can only be read by Repository Assistant and cannot be imported through Design Center.

NOTE: There is an issue with importing repository export MDL files into an OWB 11.2 repository. Before importing an MDL export of a repository from a previous release, you must

create at least one workspace in the destination 11.2 repository. Create a workspace with a name that does not conflict with workspace names from your 10.2 or 11.1 source repository.

The general steps are:

- Ensure that the destination database has an OWB 11.2 OWBSYS schema ready for importing a repository
- Create a dummy workspace in the 11.2 repository (assigning it a unique name such as “DUMMY_WORKSPACE_001”)
- Create any users in the destination environment that are referenced in the projects in the old repository
- Run Repository Assistant against the 10.2 or 11.1 source database
- When prompted, select “Export entire repository to a file” and name an output MDL file
- Run Repository Assistant against the 11.2 destination database
- When prompted, select “Import entire repository from a file” and select the MDL to import
- After the repository upgrade process is complete, upgrade the destination locations to work with OWB 11.2, as described in “Fixing Locations After Upgrade or Migration” below.

For a 10.2 repository, a single workspace is created in the destination 11.2 repository, and named based on the name of the original repository schema. For an 11.1 repository, all workspaces are recreated in the destination repository.

Full details of using the OWB 11.2 Repository Assistant and pre-upgrade and post-upgrade steps required are described in *Oracle Warehouse Builder Installation and Administration Guide 11g Release 2*, Chapter 6, “Migrating Oracle Warehouse Builder from Earlier Releases.”

In-Place Repository Upgrade to OWB 11.2

In-place repository upgrade can be used to move OWB 10.2 or 11.1 repositories to 11.2. The process behind the scenes is the same as for the migration through MDL files, but the destination database is the same as the source, and fewer manual steps are required.

The general steps are:

- Ensure that the destination database has an OWB 11.2 OWBSYS schema ready for import (run `cat_owb.sql` if necessary to create it)
- Run Repository Assistant against the database to be upgraded
- When prompted, select “Upgrade Repository on the same database instance”
- After the repository upgrade process is complete, upgrade the destination locations to work with OWB 11.2, as described in “Fixing Locations After Upgrade or Migration” below.

For OWB 10gR2, Repository Assistant exports the specified repository to an MDL file, then imports that MDL into a new 11gR2 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 11gR2 repository in OWBSYS, and re-imports the repository from MDL.

Note: Because the OWB 11.1 and 11.2 repositories are always stored in the OWBSYS schema, the upgrade of the OWB 11.1 repository within the same database overwrites the OWBSYS schema with an OWB 11.2 repository. As a result, always back up your database and your 11.1 repository before performing an upgrade to OWB 11.2.

Repository Upgrade Paths for OWB 10.1 or 9.x to 11.2

OWB 11gR2 can directly upgrade repositories from OWB 10.2.0.x or OWB 11.1.0.x. Repositories from older releases cannot be upgraded directly.

OWB 10gR2 as an Intermediate Step in Repository Upgrade

OWB 10gR2 combined the separate design-time and run-time repositories from previous 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 run-time repository contents from OWB 10.1.0.4 into the new combined repository model.

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

- Prepare an OWB 10.2 environment
- Upgrade the separate design-time and run-time repositories to the combined OWB 10.2 repository in that environment
- Upgrade the resulting 10.2 repository to OWB 11.2, as described in this document.

OWB 9.x/10.1 Repository Upgrade Path Summary

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

Current Release	Repository Upgrade Path
10.1.0.x	To 10.2.0.x, then to 11.2
9.2.0.x or Older	To 10.1.0.4, then to 10.2.0.x, 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 have a 9.2 repository, you must set up an OWB 10.1.0.4 environment and an OWB 10.2.0.5 environment for use during the upgrade steps.

If you are moving to OWB 11.2 from OWB 10.1 or earlier, then it is usually simpler is to migrate objects on a project-by-project basis into a clean OWB 11gR2 workspace. Upgrading an entire repository from releases before OWB 10.2 is only recommended if you must preserve your runtime repository metadata.

Upgrading Repository from OWB 10.1.0.4 or Later to OWB 10.2

The process of upgrading a repository from 10.1.0.4 or 10.1.0.5 to 10.2 is described in Chapter 3, “Upgrading to Oracle Warehouse Builder 10g Release 2”, in the *Oracle Warehouse Builder Installation and Configuration Guide* for OWB 10gR2, here:

http://download.oracle.com/docs/cd/B31080_01/doc/install.102/b28224/upgrade_03.htm

It is also presented in an OWB 10gR2 Oracle By Example, here:

<http://www.oracle.com/technology/obe/admin/owbmigrate.htm>

Upgrading Repository from OWB 9 to OWB 10.1.0.4

The process of upgrading 9.2.0.x or older to OWB 10.1.0.4 is described in chapter 3, “Upgrading to 10g Release 1 and Migrating Data”, in the *Oracle Warehouse Builder Installation and Configuration Guide* for 10gR1, here:

http://download.oracle.com/docs/cd/E10926_01/doc/owb.101/b12150/upgrade03.htm#i1002327

The OWB 10gR1 documentation is available on OTN, here:

http://download.oracle.com/docs/cd/E10926_01/welcome.html

Fixing Locations After Upgrade or Migration

Locations created during migration or upgrade from previous releases require an additional step before they are properly registered. You have two choices to make the locations work:

- Upgrade the locations to 11gR2 to avoid redeploying mappings and preserve past deployment history.
- Unregister and re-register locations. The effect of this, however, is that all mappings must be redeployed. All deployment history is lost.

In general, Oracle recommends upgrading the locations rather than re-registering them.

The user interface for upgrading locations is in the Repository Browser. This process is described in the *Oracle Warehouse Builder Installation and Administration Guide* for 11gR2, chapter 6, in the section entitled “Upgrading Locations.”

Design Objects Modified During Migration or Upgrade

When objects from previous projects are moved into OWB 11gR2, some design objects are modified to fit the current feature set during the upgrade process.

Changes to OWB 10.2 and 11.1 Objects

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_REFCURSOR 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_REFCURSOR 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 9.2 and 10.1 Objects

When you move a design from OWB 9.2 or OWB 10.1 to OWB 11gR2, the following objects are altered 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 previous 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 previous 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 previous 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.

Running with an Upgraded Repository Without Regenerating

Some customers may want to upgrade a production environment to OWB 11gR2 without regenerating and redeploying all of their objects to production immediately. The conflict this may create is a concern that, because of bug fixes or other changes in OWB 11gR2 code generation, regenerating the upgraded design may not exactly match the code running in production.

The following strategy can allow confidence in continuing to run the existing deployed code after an upgrade.

Assume a development environment DEV and a production environment PROD. Perform the following tasks in your DEV environment:

1. Before the upgrade, run your ETL process in DEV and capture the results, to establish a baseline for future comparisons.
2. Upgrade the DEV repository to OWB 11gR2.
3. Before regenerating, run your ETL process in DEV again. Compare the results to the original baseline from before the upgrade. The output should match. This confirms that the upgrade process itself did not change your results.
4. Regenerate and redeploy your design in DEV.
5. Run your ETL process again using the new generated code. Compare the results to the original baseline. The output should match. This confirms that the code generated by 11gR2 produces the same result as the original code.

If the results match in all cases, then you can upgrade OWB on your production system and have confidence that you do not have to redeploy all your objects because of the upgrade. Of course, future changes to your design will require deployment of the updated objects to production.



Implement Right-Time Data Loading with Oracle
Data Integration and OWB 11gR2

October 2010

Author: David Allan

Oracle Corporation
World Headquarters
500 Oracle Parkway
Redwood Shores, CA 94065
U.S.A.

Worldwide Inquiries:
Phone: +1.650.506.7000
Fax: +1.650.506.7200
oracle.com



Oracle is committed to developing practices and products that help protect the environment

Copyright © 2010, Oracle and/or its affiliates. All rights reserved. This document is provided for information purposes only and the contents hereof are subject to change without notice. This document is not warranted to be error-free, nor subject to any other warranties or conditions, whether expressed orally or implied in law, including implied warranties and conditions of merchantability or fitness for a particular purpose. We specifically disclaim any liability with respect to this document and no contractual obligations are formed either directly or indirectly by this document. This document may not be reproduced or transmitted in any form or by any means, electronic or mechanical, for any purpose, without our prior written permission.

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